กรอบแนวคิดสำหรับการสร้างกรณีทดสอบด้วยออนโทโลยีความต้องการซอฟต์แวร์
A Framework of Test Case Generation with Software Requirements Ontology

สุไรยา เจริญเระ
Suraiya Charoenreh

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
มหาวิทยาลัยสงขลานครินทร์
A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Information Technology
Prince of Songkla University
2565

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

กรอบแนวคิดสำหรับการสร้างกรณีทดสอบด้วยออนโทโลยีความต้องการซอฟต์แวร์
A Framework of Test Case Generation with Software Requirements Ontology

สุไรยา เจริญเระ
Suraiya Charoenreh

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
มหาวิทยาลัยสงขลานครินทร์
A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Information Technology
Prince of Songkla University
2565

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

ชื่อวิทยานิพนธ์ กรอบแนวคิดสำหรับการสร้างกรณีทดสอบด้วยออนโทโลยีความต้องการซอฟต์แวร์
ผู้เขียน นางสาวสุไรยา เจริญเระ
สาขาวิชา เทคโนโลยีสารสนเทศ

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

.....
(ดร.อดิศักดิ์ อินทนา)

คณะกรรมการสอบ

..... ประธานกรรมการ
(ดร.กาญจนา เหล่าเส็น)

.....กรรมการ
(ดร.อดิศักดิ์ อินทนา)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สิรยา สิทธิสาร)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

.....
(ศาสตราจารย์.ดร.ดำรงศักดิ์ ฟ้ารุ่งสาง)
คณบดีบัณฑิตวิทยาลัย

ขอรับรองว่า ผลงานวิจัยนี้มาจากการศึกษาวิจัยของนักศึกษาเอง และได้แสดงความขอบคุณบุคคลที่มีส่วนช่วยเหลือแล้ว

ลงชื่อ.....
(ดร.อดิศักดิ์ อินทนา)
อาจารย์ที่ปรึกษาวิทยานิพนธ์

ลงชื่อ.....
(นางสาวสุไรยา เจริญเระ)
นักศึกษา

ข้าพเจ้าขอรับรองว่า ผลงานวิจัยนี้ไม่เคยเป็นส่วนหนึ่งในการอนุมัติปริญญาในระดับใดมาก่อน และไม่ได้ถูกใช้ในการยื่นขออนุมัติปริญญาในขณะนี้

ลงชื่อ.....
(นางสาวสุไรยา เจริญเระ)
นักศึกษา

ชื่อวิทยานิพนธ์ กรอบแนวคิดสำหรับการสร้างกรณีทดสอบด้วยออนโทโลยีความต้องการซอฟต์แวร์
ผู้เขียน นางสาวสุไรยา เจริญเระ
สาขาวิชา เทคโนโลยีสารสนเทศ
ปีการศึกษา 2565

บทคัดย่อ

ในยุคปัจจุบันเป็นยุคที่เทคโนโลยีได้เข้ามามีบทบาทในชีวิตประจำวันของมนุษย์โดยสิ้นเชิง ทำให้นักพัฒนาเกิดแนวคิดและออกแบบเทคโนโลยีใหม่ ๆ ขึ้นมามากมายเพื่อตอบสนองความต้องการของผู้ใช้งาน แต่มีหลายครั้งที่ระบบที่ถูกพัฒนาขึ้นมาใหม่นั้นเกิดข้อผิดพลาดจากกระบวนการที่ไม่มีประสิทธิภาพในการพัฒนาซอฟต์แวร์ ทำให้นักพัฒนาต้องกลับมาทวนสอบระบบใหม่อีกครั้งเพื่อหาข้อผิดพลาด ดังนั้นขั้นตอนในการทดสอบซอฟต์แวร์จึงมีความสำคัญในการทำให้ผู้ใช้มั่นใจได้ว่าระบบที่พัฒนาขึ้นมานั้นมีคุณภาพและไม่เกิดความผิดพลาดในระหว่างการใช้งาน

ในการทดสอบการยอมรับจากผู้ใช้และการทดสอบระบบ นักทดสอบจะใช้เอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ในการสร้างกรณีทดสอบ ซึ่งเอกสารจะถูกเขียนขึ้นด้วยภาษาธรรมชาติทำให้ข้อกำหนดความต้องการที่ระบุในเอกสารมีโครงสร้างที่ไม่ชัดเจน ส่งผลให้การสร้างกรณีทดสอบไม่ได้มาตรฐานและเกิดข้อผิดพลาดในระหว่างการทดสอบซอฟต์แวร์

ดังนั้นในงานวิจัยนี้ผู้วิจัยจึงได้นำเสนอกรอบแนวคิดสำหรับการสร้างกรณีทดสอบด้วยออนโทโลยีความต้องการซอฟต์แวร์ที่ถูกใช้แทนความหมายของข้อกำหนดความต้องการและการประยุกต์ใช้เครื่องมือภาษาธรรมชาติควบคุม เพื่อให้ความต้องการมีโครงสร้างที่ชัดเจนก่อนนำไปสร้างกรณีทดสอบ รวมถึงผู้วิจัยได้พิจารณาเทคนิคการทดสอบแบบกล่องดำที่มีประสิทธิภาพ คือ เทคนิคการรวมกันของชั้นสมมูลและต้นไม้การจำแนกมาใช้ เพื่อให้การสร้างกรณีทดสอบมีความครอบคลุมและลดความซ้ำซ้อน ส่งผลให้กรณีทดสอบที่ได้มีความถูกต้องแม่นยำ และทำให้การทดสอบมีประสิทธิภาพตรงตามความต้องการของผู้ใช้งานมากยิ่งขึ้น

คำสำคัญ: การทดสอบซอฟต์แวร์ ข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ ออนโทโลยี กรณีทดสอบ การแบ่งชั้นสมมูลและต้นไม้การจำแนก

Thesis Title A Framework of Test Case Generation with Software Requirements Ontology
Author Ms. Suraiya Charoenreh
Major Program Information Technology
Academic Year 2022

ABSTRACT

Recently modern society technology is involved in the daily life of all humans. This allows developers to create new concepts and technology designs in order to respond to users' requirements. Therefore, this causes the system development to become more complex. Consequently, it leads to a high possibility of system errors. Therefore, software testing is an important aspect to guarantee users that the developed system is error-free.

In user acceptance and system testing, test cases are normally generated from the software requirements specification (SRS). As the SRS is written in a natural language which has an uncertain structure, it can possibly be ambiguous. As a result, this may cause the wrong interpretation of the requirements and finally it can allow the occurrence of incorrect test case generation.

This research proposes a framework of test case generation with software requirements ontology to enhance the reliability of existing verification and validation (V&V) techniques. This framework uses ontology and Control Natural Languages (CNL) to represent the semantics of functional requirements in SRS documents in order to increase the structure of natural language. Furthermore, the effective black-box testing technique, Combination of Equivalence and Classification Tree Method (CCTM), is included in this framework. This testing technique enables test case coverage and reduction compared with other testing techniques. This results in the generated test cases to be more accurate and efficient

Keywords: Software Testing, Software Requirements Specification, Ontology, Test Case, Equivalence and Classification Tree Method

กิตติกรรมประกาศ

งานวิจัยฉบับนี้สำเร็จลุล่วงได้ด้วยความอนุเคราะห์และความกรุณาจาก ผู้มีพระคุณหลายท่าน โดยเฉพาะอย่างยิ่งอาจารย์ที่ปรึกษาวิทยานิพนธ์ ดร.อดิศักดิ์ อินทนา ที่กรุณา ให้ความรู้ ชี้แนะแนวทาง ให้กำลังใจ และสละเวลาอันมีค่าให้คำปรึกษาทั้งในเวลาราชการและนอกเวลาราชการ ซึ่งเป็นการสร้างพลังให้ผู้วิจัยมีความอดทนและพยายามทำการศึกษาอย่างเต็มความสามารถ ผู้วิจัยรู้สึกซาบซึ้งในความเมตตากรุณา และความเสียสละของอาจารย์ในการประสิทธิ์ประสาทวิชา จึงขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ ดร.กาญจนา เหล่าเส็น ประธานกรรมการสอบวิทยานิพนธ์ และผู้ช่วยศาสตราจารย์ ดร.สิรยา สิทธิสาร กรรมการสอบวิทยานิพนธ์ รวมถึงคณาจารย์วิทยาลัยการคอมพิวเตอร์ทุกท่านที่คอยแนะนำ ช่วยเหลือ ให้กำลังใจ และถ่ายทอดวิชาแก่ข้าพเจ้า เพื่อเป็นแนวทางที่มีประโยชน์ในการปรับปรุงวิทยานิพนธ์ และนำไปประยุกต์ใช้กับงานอื่น ๆ ต่อไปในอนาคต

ขอขอบคุณ คุณวรนุช ญาณศักดิ์ เจ้าหน้าที่งานวิจัยและพัฒนานักศึกษา และคุณฐิติมา วศินพัฒนวิศิษฎ์ เจ้าหน้าที่งานบัณฑิตศึกษา ที่คอยชี้แนะแนวทางให้คำแนะนำและช่วยเหลือในด้านการติดต่อประสานงานต่าง ๆ เกี่ยวกับการทำวิทยานิพนธ์ รวมถึงบุคลากรทุกท่านที่ให้ความช่วยเหลือด้วยดีตลอดมา ทั้งนี้ขอขอบคุณวิทยาลัยการคอมพิวเตอร์ และบัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ ที่มอบโอกาสในการศึกษาและทุนสนับสนุนงานวิจัยในครั้งนี้

ขอขอบคุณบิดา มารดา และครอบครัว ที่คอยให้การสนับสนุน ส่งเสริม และอยู่เคียงข้าง มอบความห่วงใยเสมอมา ตลอดจนกัลยาณมิตรทุกท่านที่ให้ความช่วยเหลือเกื้อกูลในทุกด้าน รวมทั้งให้กำลังใจที่มีคุณค่าอย่างยิ่ง ผู้วิจัยรู้สึกซาบซึ้งในความกรุณาและความปราถนาดีจึงขอขอบพระคุณทุกท่าน มา ณ โอกาสนี้

สุไรยา เจริญเระ

สารบัญ

	หน้า
บทคัดย่อ (ภาษาไทย)	(5)
บทคัดย่อ (ภาษาอังกฤษ)	(6)
กิตติกรรมประกาศ	(7)
สารบัญ	(8)
รายการตาราง	(10)
รายการภาพประกอบ	(12)
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของงานวิจัย	1
1.2 คำถามของงานวิจัย	5
1.3 วัตถุประสงค์ของงานวิจัย	5
1.4 ขอบเขตของงานวิจัย	6
1.5 ประโยชน์ที่คาดว่าจะได้รับ	6
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	7
2.1 ทฤษฎีที่เกี่ยวข้อง	7
2.2 วรรณกรรมและงานวิจัยที่เกี่ยวข้อง	29
2.3 เปรียบเทียบงานวิจัยที่เกี่ยวข้อง	33
บทที่ 3 ขั้นตอนการดำเนินงานและกรอบแนวคิดของงานวิจัย	36
3.1 ขั้นตอนการดำเนินงานวิจัย	36
3.2 กรอบแนวคิดของงานวิจัย	40
บทที่ 4 กรณีศึกษาของต้นแบบ	54
4.1 กรณีศึกษาที่ 1 ระบบจัดการห้องสมุด (Library Management System)	54
4.2 กรณีศึกษาที่ 2 ระบบวิเคราะห์คำนวณค่าไต (GFR module)	65
บทที่ 5 กระบวนการพัฒนาต้นแบบ	76
5.1 เครื่องมือที่ใช้ในการพัฒนาต้นแบบ	76
5.2 สถาปัตยกรรมของต้นแบบ	77
5.3 หน้าจอการทำงานและส่วนต่อประสานกับผู้ใช้ (User Interface)	78
บทที่ 6 กระบวนการทดสอบและประเมินผลต้นแบบ	86
6.1 สภาพแวดล้อมที่ใช้ในการทดสอบและประเมินผลต้นแบบ	86
6.2 การทดสอบต้นแบบ	87
6.3 การประเมินผลต้นแบบ	93
6.4 สรุปการประเมินผลต้นแบบ	100

สารบัญ (ต่อ)

	หน้า
บทที่ 7 อภิปรายและสรุปผลการวิจัย	101
7.1 อภิปรายผลการวิจัย	101
7.2 ข้อจำกัดและข้อเสนอแนะของการวิจัย	102
7.3 สรุปผลการวิจัย	103
เอกสารอ้างอิง	104
ภาคผนวก ก	108
ภาคผนวก ข	115
ภาคผนวก ค	128
ประวัติผู้เขียน	150

รายการตาราง

หน้า

ตารางที่ 2.1 ตัวอย่างและคำอธิบายความหมายในโครงสร้าง ROO 14
ตารางที่ 2.2 โครงสร้างเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) 16
ตารางที่ 2.3 อธิบายสัญลักษณ์การจำแนกโดยใช้ต้นไม้การจำแนก 24
ตารางที่ 2.4 ตัวอย่างกรณีทดสอบแบบวีคนอร์มัล 26
ตารางที่ 2.5 ตัวอย่างกรณีทดสอบแบบสตรองนอร์มัล 27
ตารางที่ 2.6 ตัวอย่างกรณีทดสอบแบบวีคโรบัส 27
ตารางที่ 2.7 ตัวอย่างกรณีทดสอบแบบสตรองโรบัส 28
ตารางที่ 2.8 แนวทางการสร้างกรณีทดสอบจากการแบ่งชั้นสมมูล 29
ตารางที่ 2.9 ตารางเปรียบเทียบงานวิจัยที่เกี่ยวข้อง 33
ตารางที่ 3.1 ตัวอย่างความต้องการระบบจัดการห้องสมุด 41
ตารางที่ 3.2 โครงสร้าง ROO ของความต้องการระบบจัดการห้องสมุด 43
ตารางที่ 3.3 ตัวอย่างกฎการแปลง OWL เป็น XML 45
ตารางที่ 3.4 ตัวอย่างตัวแปรระบบจัดการห้องสมุด 51
ตารางที่ 4.1 ความต้องการระบบจัดการห้องสมุด 55
ตารางที่ 4.2 ความสัมพันธ์ระหว่างคลาสของระบบจัดการห้องสมุด 56
ตารางที่ 4.3 คุณสมบัติข้อมูลของระบบจัดการห้องสมุด 57
ตารางที่ 4.4 ค่าข้อมูลของระบบจัดการห้องสมุด 57
ตารางที่ 4.5 ตัวแปรระบบจัดการห้องสมุด 59
ตารางที่ 4.6 เงื่อนไขการยืมทรัพยากรของสมาชิกแต่ละประเภท 60
ตารางที่ 4.7 การแบ่งชั้นสมมูลระบบจัดการห้องสมุด 62
ตารางที่ 4.8 ตัวอย่างกรณีทดสอบระบบจัดการห้องสมุด 65
ตารางที่ 4.9 ความต้องการระบบวิเคราะห์คำนวณค่าไต 66
ตารางที่ 4.10 ความสัมพันธ์ระหว่างคลาสของระบบวิเคราะห์คำนวณค่าไต 67
ตารางที่ 4.11 คุณสมบัติข้อมูลของระบบวิเคราะห์คำนวณค่าไต 68
ตารางที่ 4.12 ค่าข้อมูลของระบบวิเคราะห์คำนวณค่าไต 68
ตารางที่ 4.13 ตัวแปรระบบวิเคราะห์คำนวณค่าไต 70
ตารางที่ 4.14 เงื่อนไขการแปลผลระยะความล้มเหลวของไต 71
ตารางที่ 4.15 การแบ่งชั้นสมมูลระบบวิเคราะห์คำนวณค่าไต 73
ตารางที่ 4.16 สูตรการคำนวณค่า GFR 74
ตารางที่ 4.17 ตัวอย่างกรณีทดสอบระบบวิเคราะห์คำนวณค่าไต 75
ตารางที่ 6.1 บทภาพเหตุการณ์การทดสอบฟังก์ชันของต้นแบบ 87

รายการตาราง (ต่อ)

หน้า

ตารางที่ 6.2 การทดสอบฟังก์ชันการแปลงออนโทโลยีจาก OWL เป็น XML 88
ตารางที่ 6.3 ตัวอย่างตัวแปรระบบจัดการห้องสมุด 90
ตารางที่ 6.4 การประเมินผลต้นแบบโดยกรณีศึกษาทั้ง 2 กรณีศึกษา 94
ตารางที่ 6.5 ค่าข้อมูลที่มีการเปลี่ยนแปลง 95
ตารางที่ 6.6 คำถามการประเมินต้นแบบ 96
ตารางที่ 6.7 รายละเอียดจำนวนผู้เชี่ยวชาญในการประเมินต้นแบบ 97
ตารางที่ 6.8 ผลการประเมินต้นแบบโดยผู้เชี่ยวชาญ 98
ตารางที่ 6.9 คำแนะนำและข้อเสนอแนะของผู้เชี่ยวชาญ 99

รายการภาพประกอบ

หน้า

รูปที่ 2.1 แสดงการแทนแบบโครงข่ายความหมาย 8
รูปที่ 2.2 แสดงการแทนแบบกรอบหรือเฟรม 8
รูปที่ 2.3 แสดงการแทนแบบสคริปต์ 9
รูปที่ 2.4 แสดงการแทนแบบกราฟความคิด 10
รูปที่ 2.5 แสดงการขยายกราฟความคิด 10
รูปที่ 2.6 แสดงการบ่งชี้กราฟความคิด 10
รูปที่ 2.7 แสดงขั้นตอนการออกแบบและพัฒนาออนโทโลยีของ Noy and Mcguinness 12
รูปที่ 2.8 ตัวอย่างหน้าจอการใช้งานเครื่องมือ ROO 15
รูปที่ 2.9 หน้าจอแสดงคำเตือนของเครื่องมือ ROO 15
รูปที่ 2.10 สัญลักษณ์แทนยูสเคส 17
รูปที่ 2.11 สัญลักษณ์แทนผู้กระทำ 18
รูปที่ 2.12 สัญลักษณ์การเชื่อมความสัมพันธ์ 18
รูปที่ 2.13 ความสัมพันธ์แบบส่วนขยาย 18
รูปที่ 2.14 ความสัมพันธ์แบบการเรียกใช้งาน 19
รูปที่ 2.15 ความสัมพันธ์แบบองค์ประกอบหรือการจำแนกประเภท 19
รูปที่ 2.16 สัญลักษณ์แทนชื่อระบบ 20
รูปที่ 2.17 แผนภาพยูสเคสระบบลงทะเบียนเรียน 20
รูปที่ 2.18 ตัวอย่างการประกาศ Element 21
รูปที่ 2.19 ตัวอย่างการประกาศ Attribute 22
รูปที่ 2.20 ตัวอย่างโครงสร้าง XML Schema 22
รูปที่ 2.21 กระบวนการทำงานของการทดสอบแบบกล่องดำ 23
รูปที่ 2.22 แผนภาพการจำแนกแบบต้นไม้การจำแนก 23
รูปที่ 2.23 ตัวอย่างต้นไม้การจำแนก 25
รูปที่ 2.24 ตัวอย่างการแบ่งชั้นสมมูล 25
รูปที่ 3.1 ขั้นตอนการดำเนินงานวิจัย 37
รูปที่ 3.2 กรอบแนวคิดของงานวิจัย 40
รูปที่ 3.3 แผนภาพกิจกรรมกระบวนการวิศวกรรมออนโทโลยี 42
รูปที่ 3.4 แผนภาพกิจกรรมของกระบวนการสร้างกรณีทดสอบ 44
รูปที่ 3.5 ตัวอย่างการแปลงคลาส 47
รูปที่ 3.6 ตัวอย่างการแปลงคลาสย่อย 47
รูปที่ 3.7 ตัวอย่างการแปลงความสัมพันธ์ของคลาส 47

รายการภาพประกอบ (ต่อ)

หน้า

รูปที่ 3.8 ตัวอย่างการแปลงคุณสมบัติของคลาส 48
รูปที่ 3.9 ตัวอย่างการแปลงองค์ประกอบของคลาสความสัมพันธ์ 48
รูปที่ 3.10 ตัวอย่างการแปลงตัวบ่งปริมาณสูงสุดของคุณสมบัติ 48
รูปที่ 3.11 แผนภาพวิเคราะห์คำสำหรับการแปลงออนโทโลยีจาก OWL เป็น XML 49
รูปที่ 3.12 การวิเคราะห์คำสำหรับกฎการแปลง 49
รูปที่ 3.13 ไวยกรณ์ XML ของกฎการแปลงคลาส 50
รูปที่ 3.14 ตัวอย่างรายละเอียดไฟล์ยูสเคสระบบจัดการห้องสมุด 50
รูปที่ 3.15 แผนภาพต้นไม้การจำแนก 52
รูปที่ 3.16 แผนภาพการแบ่งชั้นสมมูล 53
รูปที่ 4.1 ออนโทโลยีระบบจัดการห้องสมุด 56
รูปที่ 4.2 โครงสร้าง XML ระบบจัดการห้องสมุด 58
รูปที่ 4.3 แผนภาพยูสเคสระบบจัดการห้องสมุด 59
รูปที่ 4.4 แผนภาพต้นไม้การตัดสินใจระบบจัดการห้องสมุด 60
รูปที่ 4.5 ต้นไม้การจำแนกระบบจัดการห้องสมุด 61
รูปที่ 4.6 สูตรการคำนวณค่าปรับระบบจัดการห้องสมุด 64
รูปที่ 4.7 ออนโทโลยีระบบวิเคราะห์คำนวณค่าไต 67
รูปที่ 4.8 โครงสร้าง XML ระบบวิเคราะห์คำนวณค่าไต 69
รูปที่ 4.9 แผนภาพยูสเคสระบบวิเคราะห์คำนวณค่าไต 70
รูปที่ 4.10 แผนภาพต้นไม้การตัดสินใจระบบวิเคราะห์คำนวณค่าไต 71
รูปที่ 4.11 ต้นไม้การจำแนกระบบวิเคราะห์คำนวณค่าไต 72
รูปที่ 5.1 แผนภาพส่วนประกอบของต้นแบบ 77
รูปที่ 5.2 หน้าจอเริ่มต้นการทำงานของต้นแบบ 78
รูปที่ 5.3 หน้าจอการสร้างโปรเจคใหม่ 79
รูปที่ 5.4 หน้าจอการนำเข้าไฟล์ของต้นแบบ 79
รูปที่ 5.5 แสดงโครงสร้างไฟล์หลังจากนำเข้า 80
รูปที่ 5.6 หน้าจอเลือกฟังก์ชันระบบสำหรับสร้างกรณีทดสอบ 80
รูปที่ 5.7 แสดงหน้าจอเลือกฟังก์ชันเมื่อมีการสร้างกรณีทดสอบ 81
รูปที่ 5.8 หน้าจอการจัดการตัวแปรของต้นแบบ 81
รูปที่ 5.9 ป็อปอัพการเพิ่มตัวแปร 82
รูปที่ 5.10 หน้าจอการตั้งค่าระยะเวลา 82
รูปที่ 5.11 หน้าจอการออกแบบต้นไม้การตัดสินใจ 83

รายการภาพประกอบ (ต่อ)

หน้า

รูปที่ 5.12 หน้าจอการสร้างต้นไม้การจำแนก 83
รูปที่ 5.13 หน้าจอการแบ่งชั้นสมมูล 84
รูปที่ 5.14 ตัวอย่างหน้าจอกรณีทดสอบที่ได้จากต้นแบบ 84
รูปที่ 5.15 หน้าจอสำหรับจัดเก็บโปรเจคที่เคยสร้างกรณีทดสอบ 85
รูปที่ 6.1 การปรับกฎการแปลงของกฎที่ 11 89
รูปที่ 6.2 การจัดการตัวแปรของต้นแบบ 89
รูปที่ 6.3 การออกแบบต้นไม้การตัดสินใจของต้นแบบ 90
รูปที่ 6.4 ต้นไม้การจำแนกของต้นแบบ 92
รูปที่ 6.5 การแบ่งชั้นสมมูลของต้นแบบ 92
รูปที่ 6.6 กรณีทดสอบของต้นแบบ 93
รูปที่ 6.7 กราฟเส้นสถิติแสดงผลการประเมินคุณภาพของเครื่องมือสำหรับประเมินต้นแบบ 97
รูปที่ 6.8 กราฟแท่งสถิติแสดงการกระจายระดับความพึงพอใจของผู้เชี่ยวชาญ 98

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของงานวิจัย

ปัจจุบันเทคโนโลยีได้เข้ามามีบทบาทสำคัญในการใช้ชีวิตของมนุษย์ทุกคน การออกแบบนวัตกรรมใหม่ ๆ หรือซอฟต์แวร์ต่าง ๆ เริ่มได้รับความสนใจและมีจำนวนเพิ่มมากขึ้นเรื่อย ๆ เพื่อตอบสนองความต้องการของมนุษย์ทุกคน ในกระบวนการผลิตซอฟต์แวร์จำเป็นต้องคำนึงถึงความถูกต้องแม่นยำและคุณภาพของผลิตภัณฑ์ที่พัฒนาขึ้น เพื่อให้ได้ผลิตภัณฑ์ที่ตรงตามความต้องการและผู้ใช้งานได้รับความพึงพอใจสูงสุด ในด้านอุตสาหกรรมซอฟต์แวร์การพัฒนาผลิตภัณฑ์ให้ได้มาตรฐานและมีคุณภาพประกอบไปด้วยขั้นตอนและกระบวนการต่าง ๆ มากมายซึ่งกระบวนการหนึ่งที่เป็นส่วนสำคัญหลักในการรับประกันคุณภาพของซอฟต์แวร์ คือ ขั้นตอนของการทดสอบซอฟต์แวร์ (software testing) อย่างไรก็ตาม จากรายงานประจำปีค.ศ. 2013 ของมหาวิทยาลัยเคมบริดจ์ (Cambridge University) ระบุว่าค่าใช้จ่ายของข้อผิดพลาดซอฟต์แวร์อันเป็นผลมาจากขั้นตอนด้านคุณภาพที่ไม่ดีในอุตสาหกรรมซอฟต์แวร์มีมูลค่าประมาณ 312 พันล้านเหรียญสหรัฐ (Brady, 2013) หรือประมาณ 10,489.44 พันล้านบาทต่อปี และมีแนวโน้มว่าค่าใช้จ่ายยังคงเพิ่มขึ้นเรื่อย ๆ อย่างต่อเนื่องซึ่งเป็นตัวเลขที่ค่อนข้างสูงสำหรับค่าใช้จ่ายที่เกิดขึ้น โดยข้อผิดพลาดเกิดมาจากสาเหตุหลายประการ แต่เป็นที่น่าแปลกใจว่าสาเหตุหลักของข้อบกพร่องที่เกิดขึ้นไม่ได้เกิดจากข้อผิดพลาดในการเขียนโปรแกรม (coding) หรือการออกแบบ (design) แต่สาเหตุอันดับหนึ่งของข้อผิดพลาดเกิดจากขั้นตอนของการระบุข้อกำหนดความต้องการ (specification) (Patton, 2005) ซึ่งเป็นขั้นตอนเริ่มต้นในการพัฒนาซอฟต์แวร์เพื่อรวบรวมความต้องการของผู้ใช้งาน จากข้อผิดพลาดดังกล่าวข้างต้น วิธีการหนึ่งที่จะช่วยลดระยะเวลาและค่าใช้จ่ายในการจัดการข้อผิดพลาดเหล่านี้ คือ การทวนสอบ (verification) ตั้งแต่กระบวนการเริ่มต้นซึ่งเป็นขั้นตอนแรกของการรวบรวมข้อกำหนดความต้องการที่เป็นสาเหตุหลักของข้อผิดพลาดที่เกิดขึ้นในกระบวนการทดสอบ

กระบวนการเริ่มต้นของการทวนสอบและการทดสอบซอฟต์แวร์ (software verification and validation) คือ การวางแผนและการรวบรวมความต้องการสำหรับการทดสอบระบบ (system testing) และการทดสอบการยอมรับจากผู้ใช้ (user acceptance testing) ในการรวบรวมความต้องการเพื่อระบุข้อกำหนดของซอฟต์แวร์เป็นการทำข้อตกลงร่วมกันระหว่างผู้ใช้กับนักพัฒนา เพื่อให้ได้ข้อกำหนดที่มีความเข้าใจตรงกันของทั้งสองฝ่ายก่อนลงมือทำการพัฒนา ในการระบุความต้องการจะเก็บรวบรวมและเขียนไว้ในเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (Software Requirement Specification: SRS) ซึ่งจะถูกอธิบายด้วยรูปแบบของภาษาอย่างไม่เป็นทางการ เช่น ภาษาธรรมชาติ (Thongglin, et al., 2013) ด้วยเหตุนี้จึงทำให้ข้อกำหนด

ความต้องการมีโครงสร้างที่ไม่ชัดเจนทำให้เอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ที่เขียนขึ้นอาจจะไม่มีคุณภาพและอาจนำไปสู่การตีความคลาดเคลื่อนจากความต้องการที่แท้จริง ตัวอย่างเช่น “Each student enrolls in six courses per term.” (Berry, et al., 2003) เป็นประโยคที่อาจทำให้การตีความหมายผิดพลาด โดยคำว่า “each” เป็นคำที่ใช้ในการบ่งบอกถึงปริมาณ ซึ่งทำให้ไม่ชัดเจนว่าปริมาณที่ต้องการแท้จริงมีจำนวนเท่าไหร่ นักเรียนที่สามารถลงทะเบียนเรียน 6 วิชาต่อภาคการศึกษาเป็นได้ทุกคนหรือไม่ ซึ่งทำให้นักพัฒนาระบบอาจตีความหมายของความต้องการคลาดเคลื่อนได้ ส่งผลให้การสร้างกรณีทดสอบไม่ถูกต้องทำให้ระบบไม่ได้มาตรฐาน จึงเป็นสาเหตุสำคัญในการวิเคราะห์ความต้องการเพื่อลดความซับซ้อนในการพัฒนาซอฟต์แวร์ให้มีประสิทธิภาพตรงตามความต้องการที่แท้จริง โดยหลักการและเป้าหมายของการทดสอบ คือ ความเป็นไปได้ในการหาจุดบกพร่องและข้อผิดพลาดให้เจอเพื่อทำการแก้ไขได้อย่างตรงจุด ดังนั้นผู้วิจัยจึงได้ทำการศึกษาค้นคว้าเพื่อหาวิธีการและเทคนิคต่าง ๆ ที่สามารถทำให้การตีความเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) เป็นไปในทิศทางเดียวกันและตรงตามความต้องการของผู้ใช้ เพื่อให้กระบวนการสร้างกรณีทดสอบ (test case) ในการทดสอบซอฟต์แวร์มีความถูกต้องแม่นยำก่อนส่งต่อให้นักทดสอบระบบ (tester) นำไปใช้งาน

จากปัญหาที่กล่าวมาข้างต้น ผู้วิจัยได้มีการศึกษาค้นคว้าและให้ความสนใจในเรื่องการนำแนวคิดและเทคนิคการพัฒนาออนโทโลยี (ontology) มาประยุกต์ใช้กับศาสตร์ทางด้านการทดสอบซอฟต์แวร์เพื่อเพิ่มโครงสร้างให้กับข้อกำหนดความต้องการในเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ที่เป็นภาษาธรรมชาติให้อยู่ในรูปแบบเชิงโครงสร้างมากขึ้น โดยได้นำเครื่องมือ (Rabbit to OWL Ontologies Authoring: ROO) (Denaux, et al., 2008) ซึ่งเป็นเครื่องมือในการพัฒนาออนโทโลยีที่สามารถทำความเข้าใจได้ง่ายและไม่ซับซ้อน เครื่องมือนี้มีการกำหนดโครงสร้างของออนโทโลยีโดยใช้ภาษาธรรมชาติควบคุม (Controlled Natural Language: CNL) ที่มีชื่อเรียกว่า Rabbit language ด้วยไวยากรณ์ที่มีรูปแบบเฉพาะและถูกกำหนดไว้อย่างชัดเจนสามารถขจัดความคลุมเครือของคำศัพท์ที่ใช้ภายในออนโทโลยี อีกทั้งยังสามารถใช้โครงสร้างความรู้ที่เป็นนามธรรมซึ่งสามารถช่วยลดความซับซ้อนของการสร้างความรู้และง่ายต่อการออกแบบและพัฒนาโดยออนโทโลยีเป็นรูปแบบการแทนองค์ความรู้ของคุณสมบัติและความสัมพันธ์ในขอบเขตของสิ่งที่สนใจสามารถนำมาใช้เป็นรูปแบบข้อกำหนดซึ่งประกอบไปด้วย คำศัพท์ภายในโดเมน (domain vocabularies) แนวคิดที่สนใจ (concepts) การจำแนกแนวคิด (classifications) ลำดับชั้นของแนวคิด (taxonomies หรือ concept hierarchies) ความสัมพันธ์ระหว่างแนวคิด (relationships) ขอบเขตและข้อจำกัด (constraints) รวมถึงสัจพจน์ของโดเมน (domain axioms) (Keet, 2020) ดังนั้นการนำออนโทโลยีมาใช้ในการแทนความหมายของเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) จึงส่งผลให้เกิดประโยชน์ในการเพิ่มโครงสร้างที่ชัดเจนให้กับข้อกำหนดความต้องการที่นำมาใช้ในการสร้างกรณีทดสอบ โดยมีงานวิจัยมากมาย (Barbosa, et al., 2006; Castañeda, et al., 2010; Siegemund, et al., 2011; Sim and Brouse, 2014; Souza, et al., 2013; Tarasov, et al., 2016; Thongglin, et al., 2013; Ul Haq and Qamar, 2019; Wang, et al., 2007) ที่ให้ความสนใจในการแก้ไขปัญหาดังกล่าวข้างต้น และเพื่อให้การดำเนินการทดสอบมีความถูกต้องมากยิ่งขึ้น ในกระบวนการแทนความรู้เพื่อให้คำอธิบายเชิงความหมายสามารถแทนได้หลากหลาย

รูปแบบ (Russell and Norvig, 2003) จากการศึกษางานวิจัยอื่น ๆ ที่เกี่ยวข้องพบว่างานวิจัยส่วนใหญ่ได้กล่าวถึงการนำออนโทโลยีมาประยุกต์ใช้กับศาสตร์ทางด้านต่าง ๆ มากมาย โดยมีงานวิจัยของ Castañeda และคณะ (Castañeda, et al., 2010) งานวิจัยของ Ul Haq และ Qamar (Ul Haq and Qamar, 2019) และงานวิจัยของ Wang และคณะ (Wang, et al., 2007) ได้นำรูปแบบการแทนแบบโครงข่ายความหมาย (semantic network) ซึ่งเป็นการแทนโดยเลียนแบบหน่วยความจำของมนุษย์ที่มีความสัมพันธ์กันของโหนดแต่ละโหนดในรูปของลำดับชั้น (hierarchy) ในงานวิจัยของ Barbosa และคณะ (Barbosa, et al., 2006) ได้นำออนโทโลยีมาใช้ในกระบวนการทดสอบซอฟต์แวร์เพื่อเพิ่มประสิทธิภาพให้กับซอฟต์แวร์ และมีรูปแบบการแทนแบบกราฟความคิด (conceptual graph) ซึ่งเป็นการแทนโดยอธิบายในรูปสัญลักษณ์ของการสืบถอดคลาสแม่และคลาสลูก ในงานวิจัยของ Souza และคณะ (Souza, et al., 2013) เป็นการนำออนโทโลยีมาใช้ในกิจกรรมการทดสอบและขั้นตอนการทดสอบเพื่อเชื่อมโยงและเก็บองค์ความรู้ในการสร้างความเข้าใจสำหรับความต้องการซอฟต์แวร์ โดยมีลักษณะการแทนความรู้ในรูปแบบของเฟรม (frame) เพื่อแสดงโครงสร้างของข้อมูลแทนสถานการณ์ เหตุการณ์ การกระทำ บุคคล และสถานที่ภายในเฟรมนั้น ๆ นอกจากนี้ยังมีงานวิจัยของ Tarasov และคณะ (Tarasov, et al., 2016) ที่ได้มีการแทนความรู้ในรูปแบบของสคริปต์ (script) โดยนำออนโทโลยีมาประยุกต์ใช้กับกฎการอนุมาน (inference rules) เพื่อใช้ในการแทนโครงสร้างของเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) และแปลงเป็นไวยากรณ์ Prolog สำหรับการสร้างกรณีทดสอบ เป็นต้น

นอกเหนือจากนั้นยังมีงานวิจัยที่คล้ายคลึงกับงานวิจัยที่ผู้วิจัยได้นำเสนอ โดยในงานวิจัยของ Ul Haq และ Qamar (Ul Haq and Qamar, 2019) ได้นำออนโทโลยีมาใช้ในการแทนความหมายของโครงสร้างเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) สำหรับการสร้างกรณีทดสอบของการทดสอบแบบกล่องดำ ซึ่งเป็นขั้นตอนและบทภาพเหตุการณ์การทำงานหลัก ๆ ของการทดสอบ เช่น ผู้เกี่ยวข้องที่มีส่วนได้ส่วนเสียกับระบบ (stakeholder) เงื่อนไขก่อนดำเนินการ (pre-condition) เงื่อนไขหลังดำเนินการ (post-condition) เป็นต้น แต่ยังไม่ครอบคลุมกระบวนการทำงานทั้งหมดของการทดสอบ รวมถึงความครอบคลุมในการสร้างกรณีทดสอบและข้อมูลทดสอบ งานวิจัยของ Tarasov และคณะ (Tarasov, et al., 2016) ได้นำออนโทโลยีและกฎการอนุมานมาใช้ในการแทนโครงสร้างของข้อกำหนดความต้องการสำหรับการสร้างกรณีทดสอบ โดยกรณีทดสอบที่ได้จากการดำเนินการจะเป็นข้อมูลนำเข้าเพื่อใช้ในการทดสอบ (test input) ขั้นตอนการทดสอบ (test procedure) และผลลัพธ์ที่คาดหวัง (expected test results) จากขั้นตอนของการทดสอบนั้น ๆ แต่ไม่ได้กล่าวถึงการสร้างกรณีทดสอบและข้อมูลทดสอบให้มีความครอบคลุมสำหรับการทดสอบซอฟต์แวร์ งานวิจัยของ Souza และคณะ (Souza, et al., 2013) ได้นำเสนอ ROoST (Reference Ontology on Software Testing) เป็นการพัฒนาออนโทโลยีที่เกี่ยวข้องกับกิจกรรมการทดสอบซอฟต์แวร์ ขั้นตอนการทดสอบ การออกแบบกรณีทดสอบ รวมถึงเทคนิคการทดสอบที่เหมาะสมเพื่อให้ครอบคลุมในกระบวนการทดสอบแต่ไม่ได้เน้นในเรื่องความครอบคลุมของการสร้างกรณีทดสอบและข้อมูลทดสอบสำหรับการทดสอบซอฟต์แวร์ งานวิจัยของ Castañeda และคณะ (Castañeda, et al., 2010) ได้นำเสนอการทบทวนแนวคิดกระบวนการต่าง ๆ เกี่ยวกับออนโทโลยีเพื่อนำไปใช้ในวิศวกรรมความต้องการ วิธีการและเทคนิคต่าง ๆ ในการ

ดำเนินงาน หรือกิจกรรมอื่น ๆ ที่เกี่ยวข้องกับกระบวนการวิศวกรรมความต้องการ โดยมุ่งเน้นไปที่การให้ข้อมูลเชิงทฤษฎีมากกว่าข้อมูลในเชิงของการปฏิบัติหรือการนำไปใช้งาน งานวิจัยของ Wang และคณะ (Wang, et al., 2007) ได้เสนอแนวทางในการนำออนโทโลยีไปใช้กับการสร้างกรณีทดสอบแบบอัตโนมัติบนเว็บเซอร์วิส (web service) โดยใช้ภาษา (Web Ontology Language for Web Service: OWL-S) เพื่ออธิบายกระบวนการทำงานต่าง ๆ ภายในแอปพลิเคชันบนเว็บเซอร์วิส โดยนำ Petri-Net มาใช้ในการแทนความหมายของขั้นตอนการทดสอบ และใช้ OWL-S สำหรับนำมาสร้างข้อมูลทดสอบ แต่ถึงอย่างไรในงานวิจัยนี้ไม่ได้ให้ความสำคัญในเรื่องความครอบคลุมสำหรับการสร้างกรณีทดสอบซอฟต์แวร์ งานวิจัยของ Barbosa และคณะ (Barbosa, et al., 2006) ได้นำเสนอการพัฒนาแอปพลิเคชัน OntoTest ที่เกี่ยวข้องกับกิจกรรมการทดสอบ การกำหนดคำศัพท์ที่ใช้กันอย่างแพร่หลายในกระบวนการทดสอบ เพื่อเป็นประโยชน์ในการพัฒนาเครื่องมือที่ช่วยเพิ่มความสามารถในการสื่อสารระหว่างผู้มีความเกี่ยวข้องภายในองค์กรเพื่อให้มีความเข้าใจตรงกัน และยังมีงานวิจัยอื่น ๆ (Thongglin, et al., 2013) ที่ได้มีการกล่าวถึงโครงสร้างของหลักไวยากรณ์เพื่อหารูปแบบของคำที่ทำให้การระบุข้อกำหนดความต้องการในเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ฉบับภาษาไทยไม่เกิดความกำกวม โดยได้มีการนำรูปแบบไวยกรณ์ควบคุมมาใช้ (controlled syntax) เพื่อให้วิศวกรซอฟต์แวร์สามารถเขียนข้อกำหนดความต้องการได้อย่างถูกต้องและตรงตามความต้องการของผู้ใช้งาน

แต่ถึงอย่างไรก็ตามในงานวิจัยส่วนใหญ่มีข้อจำกัดในกระบวนการทดสอบซอฟต์แวร์สำหรับการสร้างกรณีทดสอบและข้อมูลทดสอบให้มีความครอบคลุมเพื่อใช้ในการทดสอบซอฟต์แวร์ (Barbosa, et al., 2006; Castañeda, et al., 2010; Tarasov, et al., 2016; Ul Haq and Qamar, 2019; Wang, et al., 2007) มีเพียงงานวิจัยส่วนน้อยที่ให้ความสนใจในเรื่องนี้ เช่น งานวิจัย Souza และคณะ (Souza, et al., 2013) ที่เสนอการนำออนโทโลยีมาใช้ในการแทนกิจกรรมการทดสอบและขั้นตอนการทดสอบเพื่อเชื่อมโยงและเก็บองค์ความรู้ในการสร้างความเข้าใจสำหรับความต้องการซอฟต์แวร์ตามโครงสร้างของเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) เพื่อนำเสนอแนวทางในการทดสอบที่เกี่ยวข้อง เช่น ขั้นตอนการทดสอบ การออกแบบกรณีทดสอบ เทคนิคการทดสอบที่เหมาะสม เป็นต้น โดยในกระบวนการสร้างกรณีทดสอบและข้อมูลทดสอบผู้วิจัยได้นำเสนอเทคนิคการทดสอบแบบกล่องดำ (black-box testing technique) จำนวน 2 เทคนิค คือ เทคนิคการแบ่งชั้นสมมูล (Equivalence Class Partitioning: ECP) และเทคนิคต้นไม้การจำแนก (Classification Tree Method: CTM) มาประยุกต์ใช้ร่วมกัน หรือที่เรียกว่า เทคนิคการรวมกันของชั้นสมมูลและต้นไม้การจำแนก (Combination of Equivalence and Classification Tree Method: CCTM) เพื่อให้การสร้างกรณีทดสอบและข้อมูลทดสอบมีความถูกต้องแม่นยำ และครอบคลุมทุกกรณีของการทดสอบซอฟต์แวร์ตามความต้องการของผู้ใช้งาน

ในงานวิจัยนี้ผู้วิจัยได้นำเสนอกรอบแนวคิดที่รวมการทำงานระหว่างเทคนิคการพัฒนาออนโทโลยีจากความต้องการของซอฟต์แวร์ที่อยู่ในรูปแบบภาษาธรรมชาติควบคุม (CNL) กับเทคนิคการทดสอบระบบ โดยผู้วิจัยได้พิจารณานำเครื่องมือ ROO มาช่วยในการออกแบบและพัฒนาออนโทโลยีซึ่งจะดำเนินการด้วยภาษาธรรมชาติควบคุม (CNL) หรือภาษา Rabbit Language ที่สามารถช่วยลดความซับซ้อนของความต้องการซึ่งอธิบายในรูปแบบของภาษาธรรมชาติสำหรับ

ข้อกำหนดความต้องการที่ระบุในเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ไวยกรณ์ที่มีรูปแบบเฉพาะของเครื่องมือจะช่วยเพิ่มโครงสร้างและขจัดความคลุมเครือของข้อกำหนดความต้องการให้มีความชัดเจนก่อนจะนำไปพัฒนาออนโทโลยีที่สามารถเรียนรู้และทำความเข้าใจได้ง่ายขึ้น ส่งผลให้การสร้างกรณีทดสอบมีความถูกต้องมากยิ่งขึ้น ผลลัพธ์ที่ได้จากเครื่องมือนี้จะเป็นออนโทโลยีที่อยู่ในรูปแบบ Web Ontology Languages (OWL) ซึ่งจะถูกนำไปแปลงให้เป็นพจนานุกรมข้อมูล (data dictionary) ที่อยู่ในรูปแบบโครงสร้างมาตรฐาน XML ในกระบวนการสร้างกรณีทดสอบกรอบแนวคิดนี้ได้นำเทคนิคการทดสอบที่ช่วยเพิ่มประสิทธิภาพการทดสอบด้วยเทคนิคการรวมกันของชั้นสมมูลและต้นไม้การจำแนก หรือเทคนิค CCTM ที่มีการสร้างกรณีทดสอบจากโครงสร้าง XML ของพจนานุกรมข้อมูลพิจารณาร่วมกับเงื่อนไขการตัดสินใจจากต้นไม้การตัดสินใจ (decision tree) ผลของการใช้เทคนิค CCTM จะช่วยให้การสร้างกรณีทดสอบมีความครอบคลุมทุกกรณีที่เป็นไปได้และสามารถลดจำนวนกรณีทดสอบที่ซ้ำซ้อน ซึ่งจะช่วยให้การทดสอบมีข้อผิดพลาดเกิดขึ้นน้อยลงและยังสามารถช่วยลดค่าใช้จ่ายในการแก้ไขข้อผิดพลาดที่เกิดขึ้นได้ รวมถึงยังช่วยแบ่งเบาภาระงานของนักทดสอบระบบและช่วยลดระยะเวลาในกระบวนการทดสอบ ส่งผลให้ซอฟต์แวร์ที่พัฒนามีประสิทธิภาพตรงตามความต้องการและน่าเชื่อถือมากยิ่งขึ้น เพื่อแสดงให้เห็นถึงกระบวนการทำงานของกรอบแนวคิดที่ต้นแบบ (prototype) ได้ถูกพัฒนาขึ้น โดยต้นแบบดังกล่าวจะถูกประเมินความถูกต้องและประสิทธิภาพด้วยกรณีศึกษาที่แตกต่างกัน 2 กรณีศึกษา

1.2 คำถามงานวิจัย

1) สามารถสร้างกรณีทดสอบจากออนโทโลยีของข้อกำหนดความต้องการในเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ได้อย่างไร

2) สามารถขยายขีดความสามารถในการสร้างกรณีทดสอบให้มีความครอบคลุมทุกกรณีของการทดสอบได้อย่างไร

3) สามารถยืนยันความถูกต้องในการทำงานของกรอบแนวคิดสำหรับการสร้างกรณีทดสอบได้อย่างไร

4) สามารถประเมินความถูกต้องและประสิทธิภาพของต้นแบบที่พัฒนาตามกรอบแนวคิดและการนำไปใช้งานในอนาคตได้อย่างไร

1.3 วัตถุประสงค์งานวิจัย

1) เพื่อพัฒนากรอบแนวคิดในการสร้างกรณีทดสอบจากออนโทโลยีความต้องการของซอฟต์แวร์ที่ได้ถูกพัฒนามาจากความต้องการที่อยู่ในรูปแบบภาษาธรรมชาติควบคุม (CNL) ให้มีความชัดเจนก่อนจะนำไปสร้างกรณีทดสอบ

2) เพื่อขยายขีดความสามารถในการสร้างกรณีทดสอบให้มีความครอบคลุมในทุกกรณีของการทดสอบด้วยเทคนิคการรวมกันของชั้นสมมูลและต้นไม้การจำแนก

3) เพื่อพัฒนาต้นแบบในการยืนยันการทำงานและความถูกต้องของกรอบแนวคิดสำหรับการสร้างกรณีทดสอบ

4) เพื่อประเมินความถูกต้องและประสิทธิภาพของต้นแบบในการนำไปใช้งานในอนาคตด้วยกรณีศึกษาที่กำหนดและผู้เชี่ยวชาญที่เกี่ยวข้อง

1.4 ขอบเขตของงานวิจัย

1) สร้างกรณีทดสอบจากออนโทโลยีที่ได้มีการออกแบบและพัฒนามาจากข้อกำหนดความต้องการในเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ที่เขียนด้วยภาษาอังกฤษ

2) พิจารณาข้อมูลส่วนที่เป็นความต้องการฟังก์ชันการทำงาน (functional requirement) ของเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ในการออกแบบและพัฒนาออนโทโลยีข้อกำหนดความต้องการซอฟต์แวร์

3) ต้นแบบที่พัฒนาสามารถสร้างกรณีทดสอบได้เฉพาะความต้องการที่ใช้เทคนิคการรวมกันของชั้นสมมูลและต้นไม้การจำแนกในการสร้างกรณีทดสอบเท่านั้น

4) การประเมินความถูกต้องของต้นแบบจะทำการเปรียบเทียบกรณีทดสอบที่สร้างด้วยวิธีการมือกับกรณีทดสอบที่ได้จากต้นแบบ และมีผู้เชี่ยวชาญด้านวิศวกรรมซอฟต์แวร์และการทดสอบซอฟต์แวร์เป็นผู้ประเมินการทำงานของต้นแบบ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้กรอบแนวคิดสำหรับการสร้างกรณีทดสอบด้วยออนโทโลยีความต้องการซอฟต์แวร์ที่ได้มีการเพิ่มโครงสร้างของข้อกำหนดความต้องการให้มีความชัดเจนก่อนจะนำไปสร้างกรณีทดสอบ

2) ได้ต้นแบบสำหรับสร้างกรณีทดสอบแบบอัตโนมัติจากออนโทโลยีความต้องการซอฟต์แวร์ที่ตรงตามความต้องการและครอบคลุมทุกกรณีของการทดสอบซอฟต์แวร์

3) ต้นแบบที่พัฒนาสามารถช่วยลดระยะเวลาและภาระงานในการสร้างกรณีทดสอบสำหรับนักทดสอบระบบได้

4) ต้นแบบที่พัฒนาสามารถช่วยลดการเกิดข้อผิดพลาดจากข้อกำหนดความต้องการสำหรับการสร้างกรณีทดสอบในการทดสอบซอฟต์แวร์ได้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในส่วนเนื้อหาของบทนี้จะกล่าวถึงการศึกษาทฤษฎีและผลงานวิจัยอื่น ๆ ที่มีความเกี่ยวข้องกับงานวิจัย รวมถึงเทคนิค และกระบวนการต่าง ๆ ที่มีความสำคัญกับงานวิจัยที่ผู้วิจัยได้นำเสนอ เพื่อนำองค์ความรู้ที่ได้จากการสืบค้น ค้นคว้า มาประกอบการดำเนินงานวิจัย ซึ่งเนื้อหาที่นำเสนอในบทนี้จะแบ่งออกเป็น 3 ส่วน ได้แก่ ทฤษฎีที่เกี่ยวข้อง ผลงานวิจัยที่เกี่ยวข้อง และการเปรียบเทียบงานวิจัยที่เกี่ยวข้อง ดังนี้

2.1 ทฤษฎีที่เกี่ยวข้อง

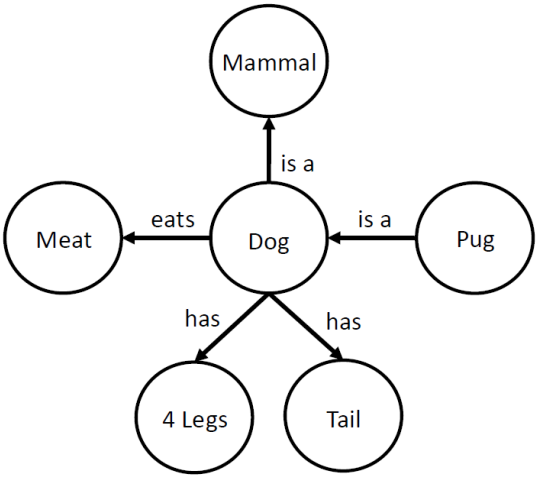
2.1.1 การแทนความรู้ (Knowledge Representation)

การแทนความรู้ คือ กระบวนการในการจัดรูปแบบองค์ความรู้หรือแทนค่าของความรู้ ซึ่งจะถูกจัดเก็บไว้ในหน่วยความจำของคอมพิวเตอร์เพื่อให้คอมพิวเตอร์เรียนรู้และทำความเข้าใจองค์ความรู้นั้น ๆ ได้ การนำเสนอความรู้ให้อยู่ในรูปแบบที่เครื่องจักรสามารถนำไปใช้ได้นั้นเป็นสาขาหลักที่สำคัญสาขาหนึ่งของปัญญาประดิษฐ์ (artificial intelligence) โดยจะนำความรู้ที่ได้มาจัดระเบียบก่อนการนำไปใช้งาน ในการแทนความรู้สามารถแทนด้วยวิธีการเขียนโปรแกรม หรืออธิบายเหตุผลที่สามารถสรุปข้อเท็จจริงต่าง ๆ ได้ โดยสามารถแบ่งประเภทของการแทนความรู้ได้หลายประเภท (Luger, 2009) ดังนี้

2.1.1.1 โครงข่ายความหมาย (Semantic Network)

โครงข่ายความหมายเป็นการแทนความรู้โดยเลียนแบบหน่วยความจำของมนุษย์ ซึ่งมีความสัมพันธ์กันระหว่างโหนดแต่ละโหนด เป็นการอธิบายแนวคิดที่มีความสัมพันธ์เชื่อมโยงกันทำให้เห็นความสัมพันธ์ในแต่ละวัตถุ (object) ได้ชัดเจนขึ้น โดยงานวิจัยเหล่านี้ (Castañeda, et al., 2010; Ul Haq and Qamar, 2019; Wang, et al., 2007) ได้นำโครงข่ายความหมายมาใช้ในการแทนความรู้ ตัวอย่างเช่น การแทนโครงข่ายความหมายของสุนัข (Dog) ซึ่งบอกคุณลักษณะ (properties) ของสุนัขว่าเป็นสัตว์เลี้ยงลูกด้วยนม (Mammal) กินเนื้อ (Meat) เป็น

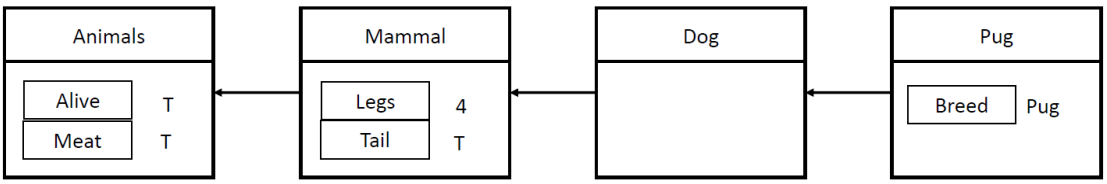
อาหาร มีขา 4 ขา (4 Legs) และมีหาง (Tail) สังเกตว่าลูกศรจากสุนัขที่ชี้ออกไปยัง Mammal, Meat, 4 Legs และ Tail เพื่อบอกลักษณะ object ของสุนัข แต่ ปั๊ก (Pug) เป็นชนิดของสุนัข ดังนั้น ลูกศรจะชี้จากปั๊กไปที่สุนัขเพื่อบ่งบอกว่าปั๊กเป็นส่วนหนึ่ง (instance) ของสุนัขนั่นเอง (Ramirez and Valdes, 2012) ดังรูปที่ 2.1



รูปที่ 2.1 แสดงการแทนแบบโครงข่ายความหมาย

2.1.1.2 กรอบ (Frame)

กรอบ หรือ เฟรม เป็นแนวคิดที่ถูกพัฒนาโดย Minsky (1974) เป็นการแทนความรู้เพื่อแสดงโครงสร้างข้อมูลแทนสถานการณ์ เหตุการณ์ การกระทำ บุคคล สถานที่ ที่เป็นต้นแบบ โดยมีส่วนประกอบ คือ ชื่อและฟิลด์ต่าง ๆ ซึ่งบอกลักษณะ (properties) ที่อธิบายถึง object ภายในกรอบหรือเฟรมนั้น ๆ และ คำอธิบายคุณลักษณะ (slot) ของสิ่งที่เป็นต้นแบบ โดยมีงานวิจัยของ Souza และคณะ (Souza, et al., 2013) ได้นำรูปแบบการแทนแบบกรอบหรือเฟรมมาใช้ในงานวิจัย ตัวอย่างเช่น การแทนวัตถุภายในเฟรมแต่ละเฟรมของสัตว์ (Animals) เลี้ยงลูกด้วยนม (Mammal) สุนัข (Dog) และสายพันธุ์ของสุนัข (Pug) ซึ่งภายในกรอบจะอธิบายคุณลักษณะของแต่ละวัตถุนั้น ๆ ที่สืบทอดมาจากคลาสที่สูงกว่า เช่น ปั๊ก (Pug) เป็นสายพันธุ์ที่สืบทอดมาจากคลาสของสุนัข เป็นต้น ดังรูปที่ 2.2



รูปที่ 2.2 แสดงการแทนแบบกรอบหรือเฟรม

2.1.1.3 สคริปต์ (Script)

สคริปต์ได้รับการออกแบบโดย Schank and Abelson (1977) เป็นการอธิบายโดยภาษาโปรแกรมใช้แทนเหตุการณ์ต่าง ๆ ที่เป็นลำดับต่อเนื่องกัน สามารถแทนความรู้ได้เฉพาะบางลักษณะเท่านั้น สคริปต์เป็นตัวแทนที่มีโครงสร้างในการอธิบายลำดับเหตุการณ์ที่กำหนดไว้ในบริบทเฉพาะ เพื่อจัดโครงสร้างแนวคิด และจัดระเบียบฐานความรู้ให้ระบบสามารถเข้าใจได้ สคริปต์มีส่วนประกอบต่าง ๆ ดังนี้ (1) Entry condition การกำหนดเงื่อนไข เพื่ออธิบายสถานการณ์ต่าง ๆ ก่อนที่เหตุการณ์นั้นจะเกิดขึ้นหรือใช้งานจริง (2) Results ผลลัพธ์หรือข้อเท็จจริงเมื่อจบสคริปต์ (3) Props ส่วนขยายที่สนับสนุนเนื้อหาและเป็น object ที่ใช้สร้างลำดับเหตุการณ์ (4) Roles บทบาทหรือการกระทำของผู้ที่เกี่ยวข้องกับการทำงานภายในระบบ (5) Scenes ฉากในการอธิบายลำดับเหตุการณ์ที่เกิดขึ้นจริง ซึ่งแต่ละฉากจะแสดงมุมมองต่าง ๆ ภายในสคริปต์ ในงานวิจัยของ Tarasov และคณะ (Tarasov, et al., 2016) ได้นำรูปแบบการแทนแบบสคริปต์มาใช้ ตัวอย่างเช่น สคริปต์ของกระบวนการร้านอาหาร ซึ่งสคริปต์สามารถทำนายได้ว่าถ้าเกิดเหตุการณ์นี้ผลที่ได้จากเหตุการณ์ควรเป็นอย่างไร จากตัวอย่างนี้สคริปต์สามารถให้ผลลัพธ์ที่สอดคล้องกับเงื่อนไขของเหตุการณ์ โดยมีเงื่อนไข คือ “Customer is hungry” และเมื่อดำเนินการตามลำดับเหตุการณ์ของสคริปต์แล้ว ผลลัพธ์ที่ได้หลังจากจบสคริปต์ คือ “Customer not hungry” เป็นต้น ดังรูปที่ 2.3

SCRIPT
Name: RESTAURANT
Roles: Customer, Waiter, Cook, Cashier
Entry condition: Customer is hungry
Props: Food, Table, Money, Menu, Tip
Events:
1/ Customer enters restaurant
2/ Customer goes to table
3/ Waiter brings menu
4/ Customer orders food
5/ Waiter brings food
6/ Customer eats food
...
10/ Customer leaves restaurant

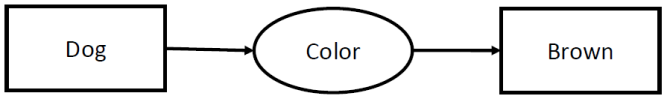
Main concept: 6
Results: Customer not hungry,
Customer has less money,
Restaurant has more money

รูปที่ 2.3 แสดงการแทนแบบสคริปต์

2.1.1.4 กราฟความคิด (Conceptual Graphs)

กราฟความคิดถูกคิดค้นโดย Sowa (2008) เป็นการอธิบายภาษาธรรมชาติในรูปของสัญลักษณ์เชิงตรรกศาสตร์และเป็นตัวกลางในการสื่อความหมายระหว่างคอมพิวเตอร์กับ

มนุษย์ให้สามารถเข้าใจตรงกัน ในการแทนความรู้โดยใช้กราฟเชิงความคิดประกอบด้วย โหนดความคิด (concept nodes) แทนด้วยสัญลักษณ์รูปสี่เหลี่ยม และโหนดความคิดสัมพันธ์ (conceptual relation nodes) แทนด้วยสัญลักษณ์รูปวงกลม แสดงความสัมพันธ์ระหว่างโหนดความคิดแต่ละโหนด เป็นแนวคิดที่เฉพาะเจาะจงสามารถแสดงความเกี่ยวข้องระหว่างกลุ่มขององค์ประกอบและใช้ในระบบค้นคืนสารสนเทศ (Montes, et al., 2001) โดยการเปรียบเทียบกันระหว่างกราฟ (graph matching) ในงานวิจัยของ Barbosa และคณะ (Barbosa, et al., 2006) ได้นำรูปแบบการแทนแบบกราฟความคิดมาใช้ ตัวอย่างเช่น โหนดของ “Dog” และ “Brown” เป็นโหนดความคิดที่ถูกเชื่อมด้วยโหนดความสัมพันธ์ คือ “Color” และสามารถอธิบายความสัมพันธ์ได้ว่า “สุนัขสีน้ำตาล” ดังรูปที่ 2.4

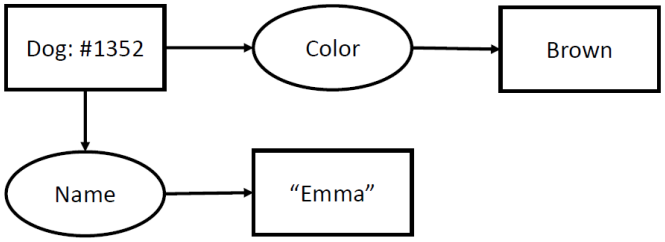


รูปที่ 2.4 แสดงการแทนแบบกราฟความคิด

เพื่อให้การแทนความรู้มีความชัดเจนขึ้นในกรณีที่ต้องการทราบว่าสุนัขตัวไหนมีสีน้ำตาล โดยสามารถทำการกำหนดชื่อให้กับสุนัขเพื่อบ่งชี้ถึงตัวสุนัขนั้น ๆ ทำให้สามารถระบุได้ว่า สุนัขตัวที่ชื่อ “Emma” มีสีน้ำตาล ดังรูปที่ 2.5

รูปที่ 2.5 แสดงการขยายกราฟความคิด

จากรูปที่ 2.5 ถึงแม้จะทราบแล้วว่าสุนัขที่ชื่อ “Emma” มีสีน้ำตาล แต่ยังไม่ชัดเจนว่าเป็นสุนัขที่ชื่อ “Emma” ตัวไหน เนื่องจากสุนัขที่มีสีน้ำตาล และชื่อ “Emma” อาจจะมีมากกว่า 1 ตัว จึงต้องมีการกำหนดตัวบ่งชี้ (markers) ที่เฉพาะเจาะจงให้กับโหนดความคิดเพื่อให้สามารถระบุได้เจาะจงถึงสิ่งที่ต้องการ เช่น การกำหนดให้สุนัขที่มีตัวบ่งชี้ #1352 มีชื่อว่า “Emma” และมีสีน้ำตาล ดังรูปที่ 2.6



รูปที่ 2.6 แสดงการบ่งชี้กราฟความคิด

2.1.2 กระบวนการออกแบบและพัฒนาออนโทโลยี

ออนโทโลยี (ontology) เป็นเทคโนโลยีที่ใช้ในการกำหนดรูปแบบโครงสร้างให้มีความหมายตามขอบเขตของสิ่งที่สนใจ ซึ่งมีการกำหนดการอธิบายความหมายที่มีโครงสร้างและความสัมพันธ์ที่เชื่อมโยงกัน ทำให้ออนโทโลยีมีความสามารถในการถ่ายทอดคุณสมบัติ (inheritance) โดยการกำหนดนิยามความคิดให้อยู่ในรูปแบบของโหนดหรือคลาส (class) ออนโทโลยีสามารถนำมาใช้เป็นตัวแทนความรู้เฉพาะด้านเพื่อเชื่อมโยงความรู้ที่มีอยู่อย่างกระจัดกระจายให้รวมอยู่ในที่เดียวกันและมีความสัมพันธ์กัน ออนโทโลยีถูกนำมาแทนความรู้ของแนวคิดด้วยข้อกำหนด (specification) เพื่อแสดงมโนภาพ (conceptualization) ของสิ่งต่าง ๆ ที่มีอยู่ในขอบเขตความรู้หรือโดเมน (domain) นั้น ๆ โดยข้อกำหนดจะถูกอธิบายโครงสร้างฐานความรู้ทางด้านใดด้านหนึ่งในการใช้งานข้อมูลร่วมกัน (information sharing) ระหว่างผู้ที่มีส่วนเกี่ยวข้องในโดเมนให้เกิดความเข้าใจตรงกัน (อิสรา ชื่นตา, จารี ทองคำ, และจิรัฏฐา ภูบุญอบ, 2558) ออนโทโลยีถือเป็นหัวใจหลักของการแทนความรู้ (knowledge representation) โดยเป็นรูปแบบหนึ่งของการแทนความรู้แบบโครงข่ายความหมายซึ่งสามารถกำหนดคำที่ใช้เป็นตัวแทนความรู้ของสิ่งที่สนใจได้ นอกจากนี้ออนโทโลยียังสามารถนำไปประยุกต์ใช้ในกระบวนการประมวลผลภาษาเพื่อเพิ่มประสิทธิภาพในการทำงานและยังสามารถนำไปประยุกต์ใช้กับระบบการทำงานต่าง ๆ ที่เกี่ยวข้องกับมโนภาพเพื่อให้การทำงานมีประสิทธิภาพมากยิ่งขึ้น เช่น ระบบการทำงานของ Search Engine ในการค้นหาและเข้าถึงข้อมูลที่ผู้ใช้งานต้องการ ซึ่งจากข้อมูลที่มีอยู่อย่างมหาศาลออนโทโลยีสามารถค้นหาโดยอ้างอิงจากฐานความรู้เพื่อให้ระบบสืบค้นสามารถค้นหาคำที่เกี่ยวข้องกับคำค้นนั้น ๆ ได้ ทั้งคำที่มีความหมายเหมือนกันและคำที่เกี่ยวข้องกัน โดยภาษาที่ใช้ในการออกแบบและพัฒนาออนโทโลยีมีอยู่หลากหลายภาษา ตัวอย่างเช่น ภาษา XML, RDF, RDFS และ OWL เป็นต้น

กระบวนการออกแบบและพัฒนาออนโทโลยีเป็นการกำหนดแนวคิดและความสัมพันธ์ให้ใกล้เคียงกับสิ่งที่สนใจมากที่สุด (Kabmala, Manmart, and Malaivongs, 2006) ซึ่งความเหมาะสมของขั้นตอนในการออกแบบและพัฒนาออนโทโลยีจะขึ้นอยู่กับบริบทแวดล้อมและวัตถุประสงค์ในการนำออนโทโลยีไปใช้งาน โดยกระบวนการออกแบบและพัฒนาออนโทโลยีมีอยู่หลากหลายวิธี ได้แก่

1) Noy and Mcguinness (2001) ได้นำเสนอการออกแบบและพัฒนาออนโทโลยีออกเป็น 7 ขั้นตอน

1.1) กำหนดโดเมนและวัตถุประสงค์ของออนโทโลยี
1.2) พิจารณาออนโทโลยีที่มีอยู่ว่าสารมารถนำมาปรับใช้ใหม่ได้หรือไม่
1.3) กำหนดคำศัพท์ที่ต้องการใช้ในออนโทโลยี
1.4) กำหนดคลาสหรือแนวคิด และลำดับชั้นของคลาส
1.5) กำหนดคุณสมบัติของคลาส
1.6) กำหนดเงื่อนไขของคลาสและความสัมพันธ์
1.7) สร้างชุดข้อมูลอินสแตนส์ (instances) เพื่อใช้ในการประมวลผล

2) Sugumaran and Storey (2002) ได้นำเสนอการออกแบบและพัฒนาออนโทโลยีด้วยแนวคิดฮิวริสติก (heuristic-based) หรือแนวคิดของความเป็นเหตุเป็นผล ออกเป็น 4 ขั้นตอน

2.1) การกำหนดคำศัพท์พื้นฐานทั่วไปที่ใช้บ่อยที่สุด รวมถึงการกำหนดคำเหมือน (synonym)

2.2) การกำหนดประเภทของความสัมพันธ์ระหว่างคำศัพท์และความสัมพันธ์ระหว่างออนโทโลยี

2.3) การกำหนดเงื่อนไขความสัมพันธ์และกฎเบื่องต้น

การกำหนดกฎระดับสูงในการจัดเก็บรวบรวมองค์ความรู้ ในขอบเขตของกฎที่ชัดเจนและกฎที่มีความเกี่ยวข้องกัน

3) Hakimpour and Geppert (2001, 2005) ได้นำเสนอการออกแบบและพัฒนาออนโทโลยีเพื่อแก้ปัญหาความหลากหลายของความหมาย (semantic heterogeneity) ออกเป็น 5 ขั้นตอน

3.1) กำหนดขอบเขตให้ชัดเจนรวมถึงโดเมนและแหล่งที่มาในการออกแบบและพัฒนาออนโทโลยี

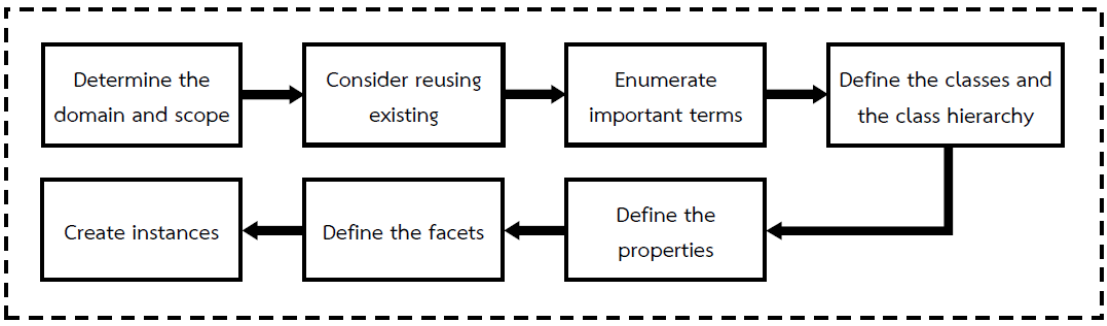
3.2) การสกัดคำศัพท์และแนวคิดจากแหล่งข้อมูลภายในบริบทแวดล้อม

3.3) การสกัดนิยาม กฎเกณฑ์ และความสัมพันธ์จากกระบวนการก่อนหน้า

3.4) กำหนดคำศัพท์ให้สมบูรณ์ทั้งแนวคิดและความสัมพันธ์

3.5) สร้างรูปแบบหลังการสกัดนิยามด้วยภาษาเชิงตรรกะและนำเสนอรูปแบบที่ใช้

จากการศึกษากระบวนการออกแบบและพัฒนาออนโทโลยีข้างต้นพบว่ากระบวนการที่ถูกกล่าวถึงมากที่สุด คือ ขั้นตอนของการกำหนดโดเมนและวัตถุประสงค์ของโอนโทโลยีภายใต้โดเมนของสิ่งที่สนใจเพื่อใช้ในการกำหนดคำศัพท์และนำความรู้ที่มีอยู่มาใช้ใหม่ได้ ซึ่งผู้วิจัยได้ประยุกต์ใช้ขั้นตอนในการออกแบบและพัฒนาออนโทโลยีข้างต้นโดยอิงจากขั้นตอนของ Noy and Mcguinness (2001) ดังรูปที่ 2.7



รูปที่ 2.7 แสดงขั้นตอนการออกแบบและพัฒนาออนโทโลยีของ Noy and Mcguinness

2.1.3 เครื่องมือสำหรับออกแบบและพัฒนาออนโทโลยี (Ontology Authoring)

เครื่องมือในการออกแบบและพัฒนาออนโทโลยีเป็นเครื่องมือที่พัฒนาขึ้นมาเพื่อช่วยให้ผู้เชี่ยวชาญด้านโดเมนสามารถจัดการองค์ความรู้ให้เหมาะสมต่อการนำองค์ความรู้ไปใช้งานด้วยการนำองค์ความรู้ที่มีอยู่มาปรับใช้ได้โดยอัตโนมัติ เนื่องจากเครื่องมือถูกออกแบบให้สามารถสร้างออนโทโลยีโดยใช้ภาษาธรรมชาติควบคุม (CNL) ซึ่งจะใช้คำหรือประโยคเฉพาะตามหลักไวยกรณ์ของภาษาควบคุมเพื่อช่วยให้ผู้ใช้สามารถออกแบบและพัฒนาออนโทโลยีได้ง่ายขึ้น เครื่องมือนี้มีชื่อเรียกอย่างเป็นทางการว่า Rabbit to OWL Ontology Authoring Tool (ROO) เป็นเครื่องมือสำหรับออกแบบและพัฒนาออนโทโลยีอย่างง่ายโดยผู้เชี่ยวชาญด้านโดเมนและสามารถแบ่งปันองค์ความรู้ด้วยภาษา Web Ontology Language (OWL) ในการนำองค์ความรู้ไปใช้งาน (Denaux, et al., 2009)

2.1.3.1 ภาษาธรรมชาติควบคุม (Controlled Natural Language: CNL)

ภาษาธรรมชาติควบคุม (Controlled Natural Language: CNL) เป็นกลุ่มย่อยของภาษาธรรมชาติที่ได้มีการจำกัดไวยากรณ์และคำศัพท์เพื่อลดหรือขจัดความคลุมเครือหรือความกำกวมของภาษาธรรมชาติ การกำหนดภาษาย่อยของภาษาธรรมชาติสำหรับการประมวลผลด้วยคอมพิวเตอร์ถูกบุกเบิกในทศวรรษที่ 1970 (Denaux, 2013) ตัวอย่างเช่น รูปแบบแรกของภาษาธรรมชาติควบคุม (CNL) ถูกใช้ในการสร้างโดเมนของการสนทนาซึ่งทำการแปลระหว่างประโยคภาษาธรรมชาติเข้ากับโครงสร้างอย่างเป็นทางการผ่านสื่อกลางของมนุษย์ อีกตัวอย่างหนึ่งของภาษาควบคุมถูกนำมาใช้เป็นส่วนติดต่อสำหรับการเขียนคำสั่ง SQL รวมถึงการกำหนดภาษาย่อยของภาษาอังกฤษและภาษาฝรั่งเศสสำหรับการแปลภาษาอัตโนมัติในโดเมนเฉพาะของกรมอุตุนิยมวิทยาและคู่มือการบิน ภาษาธรรมชาติควบคุม (CNL) ยังเป็นส่วนหนึ่งของภาษาธรรมชาติที่ใช้งานได้ง่ายและช่วยให้สามารถอธิบายองค์ความรู้และกำหนดโดเมนด้วยรูปประโยคที่ง่ายต่อการเข้าใจ โดยมีรูปแบบประโยคเฉพาะที่มีความกระชับ ช่วยให้ผู้ใช้หลีกเลี่ยงการใช้คำสรรพนาม และพิจารณาใช้เฉพาะคำที่บัญญัติในพจนานุกรมเพื่อให้เป็นมาตรฐานเดียวกัน ดังนั้นภาษาธรรมชาติควบคุม (CNL) จึงมีส่วนช่วยในการลดความคลุมเครือและความซับซ้อนของภาษาธรรมชาติ ทำให้องค์ความรู้ที่ได้มีโครงสร้างและมีความชัดเจนมากยิ่งขึ้น

2.1.3.2 เครื่องมือ (Rabbit to OWL Ontology Authoring Tool: ROO)

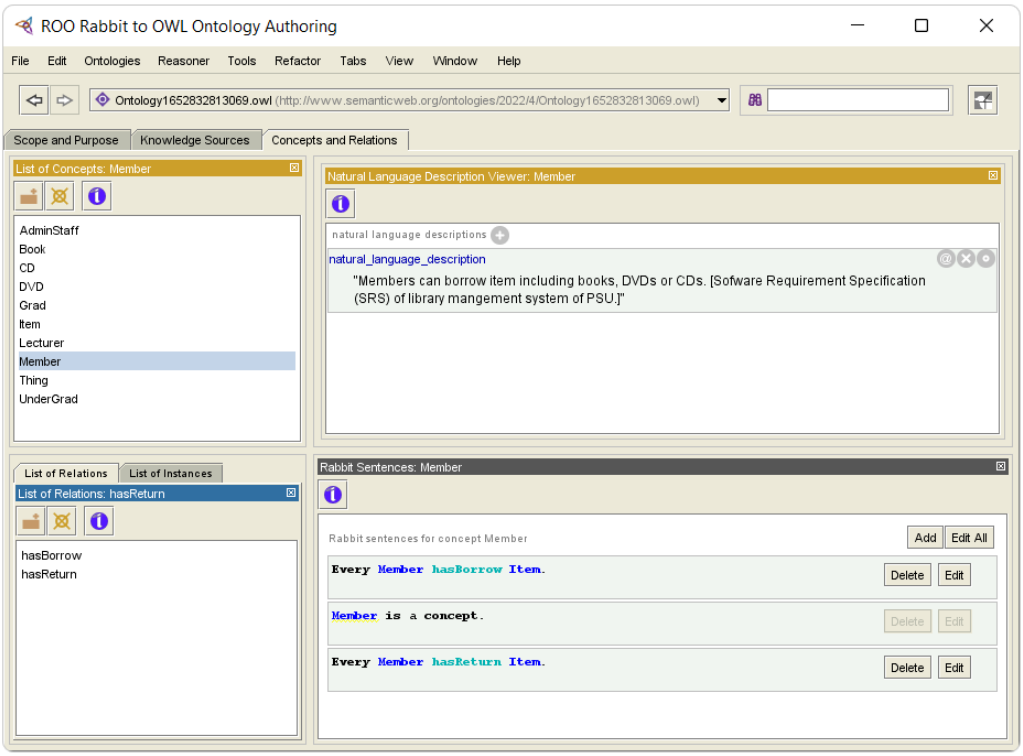
เครื่องมือ ROO (Rabbit to OWL Ontology authoring) เป็นเครื่องมือในการออกแบบและพัฒนาออนโทโลยีเพื่อสนับสนุนการสร้างแนวคิดโดยผู้เชี่ยวชาญด้านโดเมนได้อย่างง่ายและสามารถอธิบายโดเมนของสิ่งที่สนใจได้อย่างอิสระ (Denaux, et al., 2008) โดยการสร้างออนโทโลยีในรูปแบบประโยคที่เป็นภาษาอังกฤษตามธรรมชาติซึ่งมีไวยากรณ์ที่กำหนดไว้อย่าง

ชัดเจน เครื่องมือ ROO มีรูปแบบโครงสร้างของภาษาแบบภาษาธรรมชาติควบคุม (CNL) ซึ่งมีชื่อเรียกว่า Rabbit language หรือการใช้ภาษาธรรมชาติในการดำเนินงานเพื่อให้สามารถใช้โครงสร้างความรู้ที่เป็นนามธรรมซึ่งสามารถช่วยลดความซับซ้อนของการสร้างความรู้และง่ายต่อการออกแบบและพัฒนาออนโทโลยี อีกทั้งยังช่วยให้ผู้เชี่ยวชาญโดเมนสามารถพัฒนาออนโทโลยีได้ง่ายโดยไม่จำเป็นต้องมีพื้นฐานหรือความรู้มากนัก โดยเครื่องมือ ROO มีการกำหนดโครงสร้างพื้นฐานที่สำคัญหลัก ๆ ได้แก่ แนวคิด (concept) ความสัมพันธ์ (relationship) และข้อมูล (instance หรือ individual) ที่ใช้ในการออกแบบและพัฒนาออนโทโลยี สามารถแสดงตัวอย่างและอธิบายรูปแบบไวยกรณ์ของเครื่องมือ ROO ดังตารางที่ 2.1

ตารางที่ 2.1 ตัวอย่างและคำอธิบายความหมายในโครงสร้าง ROO

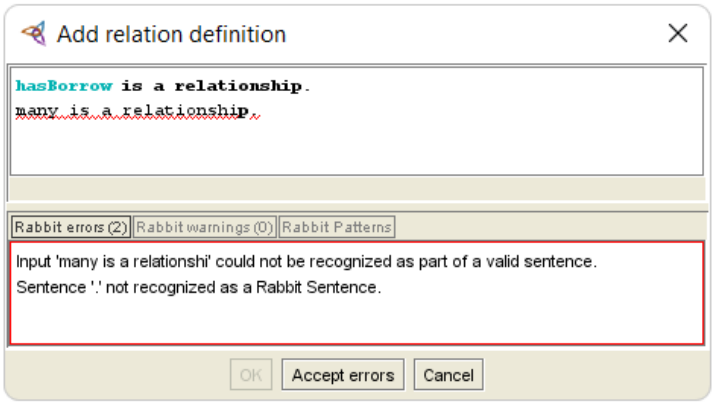
ไวยากรณ์	ความหมาย	ตัวอย่าง
<concept> is a concept.	การกำหนด concept หรือ class ของสิ่งที่สนใจ	- Member is a concept - Item is a concept
Every <concept> is a kind of <concept>.	การกำหนดให้ concept หนึ่ง เป็น subclass ของอีก concept หนึ่ง	Every Book is a kind of Item Every CD is a kind of Item Every DVD is a kind of Item
<relation> is a relationship.	การกำหนดความสัมพันธ์ที่ใช้เชื่อม concept	hasReturn is a relationship.
Every <concept> <relation> <concept>.	การกำหนดความสัมพันธ์ระหว่าง concept	Every Member hasReturn Item
<individual> is a <concept >	การกำหนดข้อมูลให้กับคลาส	Peter is a Member

จากตารางที่ 2.1 เป็นรูปแบบพื้นฐานในการออกแบบออนโทโลยีด้วยภาษา Rabbit นอกจากนี้ยังสามารถเรียนรู้โครงสร้างเพิ่มเติมและการกำหนดรูปแบบประโยคอื่น ๆ ของ ROO ได้ในงานวิจัยของ Denaux (2013) โดยความคงเส้นคงวาของคำศัพท์ที่ใช้และการกำจัดคำอภิธานศัพท์ที่ไม่ชัดเจนในโครงสร้างของ ROO สามารถช่วยลดความกำกวมของภาษาธรรมชาติและช่วยให้คำแนะนำในการสร้างออนโทโลยีที่เข้าใจได้ง่ายโดยรูปแบบไวยกรณ์ที่มีความเฉพาะ อีกทั้งยังสามารถช่วยให้ผู้ใช้ป้อนโครงสร้างที่ถูกต้องในการออกแบบและพัฒนาออนโทโลยี ส่งผลให้ออนโทโลยีที่ได้มีความชัดเจนทำให้เกิดประโยชน์และมีความเข้าใจตรงกันของผู้ใช้ภายในโดเมน สามารถแสดงตัวอย่างหน้าจอการทำงานของเครื่องมือ ROO ดังรูปที่ 2.8



รูปที่ 2.8 ตัวอย่างหน้าจอการใช้งานเครื่องมือ ROO

ในการประมวลผลภาษาธรรมชาติด้วยเครื่องมือ ROO จะหลีกเลี่ยงการใช้คำศัพท์ทางภาษาศาสตร์ เช่น คำนำหน้านาม (determiner) หรือคำคุณศัพท์ (adjective) ซึ่งเป็นคำในกลุ่มคำที่ใช้แทนคน สัตว์ สิ่งของ หรือสถานที่ หรือคำในการบ่งบอกปริมาณ เช่น a, an, the, each, some, any, many เป็นต้น ซึ่งจะช่วยให้ผู้ใช้หลีกเลี่ยงการนำเสนอความไม่ชัดเจนในออนโทโลยี นอกจากนี้ภายในโครงสร้างของ ROO ยังช่วยให้ผู้ใช้สามารถหลีกเลี่ยงการกำหนดคุณสมบัติหรือ entity ที่ทำให้ออนโทโลยีมีความคลุมเครือ โดยเครื่องมือ ROO จะแสดงคำเตือนเมื่อประโยคของผู้ใช้มีการใช้คำหรือวลีที่อาจนำไปสู่ความกำกวม ดังรูปที่ 2.9



รูปที่ 2.9 หน้าจอแสดงคำเตือนของเครื่องมือ ROO

2.1.4 ข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (Software Requirement Specification: SRS)

ในเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) เป็นข้อกำหนดที่ระบุขึ้นในกระบวนการพัฒนาซอฟต์แวร์ โดยก่อนพัฒนาจะต้องทำการวิเคราะห์ความต้องการของระบบก่อนว่ามีความต้องการหรือฟังก์ชันการทำงานอะไรบ้างที่จำเป็นต่อระบบ นักวิเคราะห์ระบบจะทำการเก็บความต้องการจากผู้ใช้โดยเขียนเป็นเอกสารที่เรียกว่า SRS หรือ Software Requirement Specification เพื่อระบุข้อกำหนดความต้องการของผู้ใช้ (user requirement) และความต้องการของระบบ (system requirement) โดยใช้ภาษาธรรมชาติในการเขียนเอกสาร ซึ่งรายละเอียดต่าง ๆ ภายในเอกสารจะขึ้นอยู่กับชนิดของระบบที่ทำการพัฒนาและกระบวนการในการพัฒนา เพื่อความเข้าใจตรงกันของผู้ที่เกี่ยวข้อง โดย IEEE Standard 830-1998 (1998) ได้กำหนดโครงสร้างของเอกสารเป็นมาตรฐานเบื้องต้นไว้ ดังตารางที่ 2.2

ตารางที่ 2.2 โครงสร้างเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS)

ข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS)
1. บทนำ (Introduction) 1.1 วัตถุประสงค์ของเอกสาร 1.2 ขอบเขตของระบบ 1.3 คำจำกัดความ ตัวย่อ และอภิธานศัพท์ 1.4 เอกสารอ้างอิง 1.5 ภาพรวมของเอกสาร 2. รายละเอียดทั่วไป (General Description) 2.1 มุมมองเกี่ยวกับระบบ 2.2 ฟังก์ชันของระบบ 2.3 คุณสมบัติของผู้ใช้ 2.4 ข้อจำกัดทั่วไป 2.5 สมมติฐานและอ้างอิง 3. ข้อกำหนดความต้องการ (Specific Requirement) กล่าวถึงความต้องการของฟังก์ชันการทำงาน และความต้องการที่ไม่ใช่ฟังก์ชันการทำงาน เช่น ความต้องการฟังก์ชันการทำงานที่ระบบจำเป็นต้องมี และความต้องการด้านประสิทธิภาพและความน่าเชื่อถือ เป็นต้น 4. ภาพผนวก (Appendices) 5. ดัชนีคำสำคัญ (Index)

2.1.4.1 ความต้องการของฟังก์ชันการทำงาน (Functional Requirement)

ความต้องการฟังก์ชันการทำงาน เป็นการวิเคราะห์ความต้องการของระบบ ซึ่งเป็นความต้องการหลักที่ระบบต้องดำเนินการและจำเป็นต้องทำ กล่าวคือ เป็นฟังก์ชันของระบบที่มีการปฏิสัมพันธ์กับผู้ใช้ เป็นฟังก์ชันที่สนับสนุนและตอบสนองการทำงานของผู้ใช้โดยตรง ตัวอย่างฟังก์ชันของระบบลงทะเบียนเรียน เช่น นักศึกษาสามารถลงทะเบียนเรียนได้ อาจารย์สามารถเปลี่ยนแปลงตารางเวลาเรียนได้ หรือนักศึกษาสามารถตรวจสอบผลการเรียนและสถานภาพของนักศึกษาได้ เป็นต้น

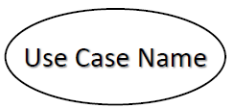
2.1.4.2 ความต้องการที่ไม่ใช่ฟังก์ชันการทำงาน (Non-Functional Requirement)

ความต้องการที่ไม่ใช่ฟังก์ชันการทำงาน เป็นความต้องการอื่น ๆ ที่ไม่ใช่ความต้องการหลักของระบบที่จำเป็นต้องทำ ซึ่งไม่ได้เป็นฟังก์ชันการทำงานแต่เป็นความคาดหวังว่าระบบควรจะต้องมี เช่น ความปลอดภัย ความน่าเชื่อถือ ประสิทธิภาพ เวลาในการตอบสนอง ฯลฯ ตัวอย่างความต้องการที่ไม่ใช่ฟังก์ชันการทำงาน เช่น ระบบไม่ควรเปิดเผยข้อมูลส่วนตัวของผู้ใช้ ระบบสามารถประมวลผลได้รวดเร็ว หรือระบบสามารถใช้งานได้ง่ายและไม่ซับซ้อน เป็นต้น

2.1.5 แผนภาพยูสเคส (Use Case Diagram)

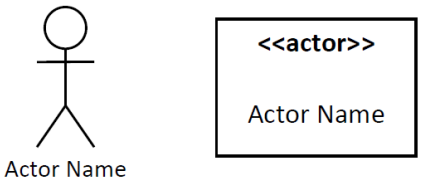
แผนภาพยูสเคส (Cockburn, 2000) คือ แผนภาพที่แสดงการทำงานระหว่างผู้ใช้กับระบบ และความสัมพันธ์ของระบบงานย่อยภายในระบบงานหลัก ในการเขียนแผนภาพยูสเคสผู้ใช้หรือผู้กระทำจะถูกกำหนดให้เป็น Actor และระบบงานย่อย คือ Use Case โดยมีจุดประสงค์ในการแสดงภาพรวมเพื่ออธิบายการทำงานภายในระบบว่ามีหน้าที่หรือฟังก์ชันการทำงานอะไรบ้าง โดยมีสัญลักษณ์ต่าง ๆ ในแผนภาพ ดังนี้ (Schneider and Winters, 2001)

1) Use Case หรือ ยูสเคส คือ หน้าที่ที่ระบบต้องกระทำ สามารถแทนด้วยสัญลักษณ์รูปวงรีและมีชื่อของยูสเคสอยู่ภายในวงรีเป็นคำกริยาหรือกริยาวลีก็ได้ เช่น ลงทะเบียนเรียน ตรวจสอบรายวิชา เป็นต้น ดังรูปที่ 2.10



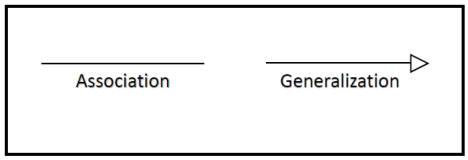
รูปที่ 2.10 สัญลักษณ์แทนยูสเคส

2) Actor คือ ผู้กระทำ หรือผู้ที่เกี่ยวข้องกับระบบทั้ง Primary Actor และ Stakeholder Actor โดยผู้กระทำที่เป็นมนุษย์จะใช้สัญลักษณ์รูปคน (Stick Man Icon) ส่วนผู้กระทำที่ไม่ใช่มนุษย์ เช่น ระบบงานอื่นที่อยู่นอกเหนือจากสิ่งที่สนใจจะใช้รูปสี่เหลี่ยมแล้วเขียน “<<actor>>” ไว้ด้านบนภายในสี่เหลี่ยม ดังรูปที่ 2.11



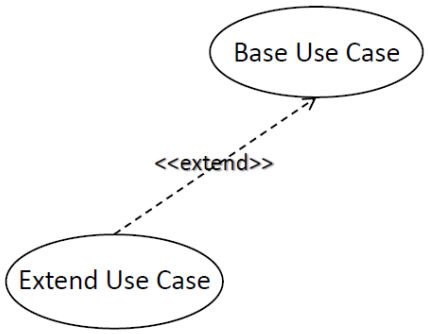
รูปที่ 2.11 สัญลักษณ์แทนผู้กระทำ

3) Connection คือ เส้นที่ลากเชื่อมต่อระหว่างผู้กระทำกับยูสเคสที่มีปฏิสัมพันธ์กัน โดยใช้สัญลักษณ์เส้นตรงไม่มีหัวลูกศร หรือ Association ส่วนเส้นที่ใช้เชื่อมต่อระหว่างยูสเคสกับยูสเคส หรือผู้กระทำกับผู้กระทำ ในกรณีที่เป็นองค์ประกอบหรือการแยกประเภท จะใช้สัญลักษณ์เส้นตรงมีหัวลูกศรโปร่ง หรือที่เรียกว่า Generalization ดังรูปที่ 2.12



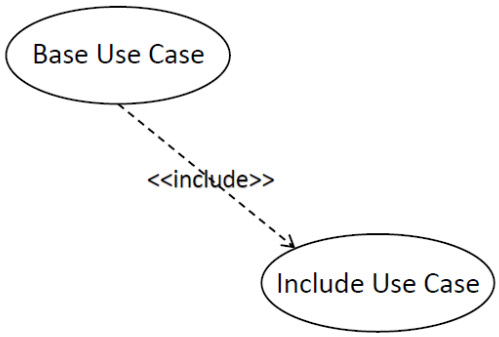
รูปที่ 2.12 สัญลักษณ์การเชื่อมความสัมพันธ์

4) Extend Relationship เป็นความสัมพันธ์ที่เป็นส่วนขยายกิจกรรมของยูสเคสจะเกิดขึ้นในกรณีที่มีเงื่อนไขบางอย่างส่งผลให้กิจกรรมภายในยูสเคสเปลี่ยนแปลงไปจากเดิม โดยยูสเคสที่ดำเนินการตามปกติจะเรียกว่ายูสเคสหลัก (base use case) ส่วนยูสเคสที่มาขยายยูสเคสหลักเรียกว่ายูสเคสขยาย (extend use case) สามารถแทนด้วยสัญลักษณ์ <<extend>> ซึ่งจะเขียนไว้บนเส้นเชื่อมความสัมพันธ์และหัวลูกศรจะชี้มายังยูสเคสหลัก ดังรูปที่ 2.13



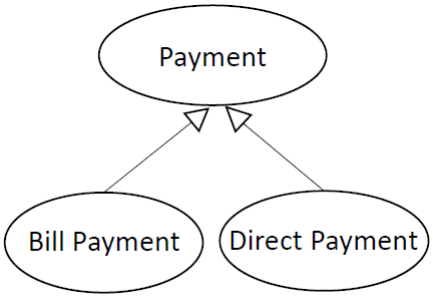
รูปที่ 2.13 ความสัมพันธ์แบบส่วนขยาย

5) Include Relationship เป็นความสัมพันธ์ที่เกิดขึ้นในกรณีที่ยูสเคสหนึ่งไปเรียกหรือดึงกิจกรรมของอีกยูสเคสมาใช้จะเรียกว่ายูสเคสหลัก (base use case) และยูสเคสที่ถูกเรียกใช้จะเรียกว่ายูสเคสร่วม (Include use case) ซึ่งยูสเคสหลักต้องพึ่งพาความสามารถหรือฟังก์ชันการทำงานของยูสเคสร่วม โดยยูสเคสที่มีความสัมพันธ์กันสามารถทำงานได้ตามปกติแม้จะไม่ได้ถูกเรียกใช้ สามารถเขียนแทนด้วยเส้นสัญลักษณ์ในทิศทางตรงข้ามกับ “Extend Relationship” โดยเส้นเชื่อมความสัมพันธ์จะชี้หัวลูกศรไปที่ยูสเคสร่วม และเขียน <<include>> ไว้บนเส้นความสัมพันธ์ ดังรูปที่ 2.14



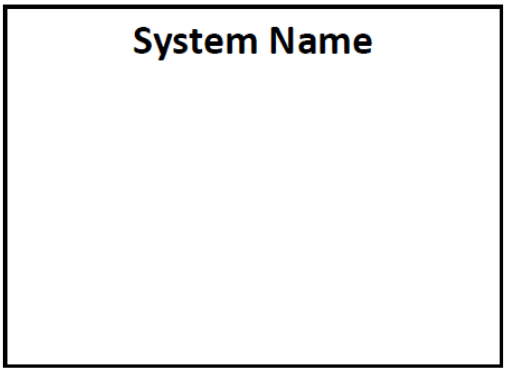
รูปที่ 2.14 ความสัมพันธ์แบบการเรียกใช้งาน

6) Generalization Relationship เป็นความสัมพันธ์อีกรูปแบบหนึ่งของยูสเคสซึ่งเป็นองค์ประกอบหรือการจำแนกประเภทที่มีการสืบถอดคุณสมบัติของยูสเคสหลัก โดยสามารถเป็นได้ทั้งความสัมพันธ์แบบผู้กระทำกับผู้กระทำ และความสัมพันธ์แบบยูสเคสกับยูสเคส โดยให้ยูสเคส Payment (การชำระเงิน) เป็นยูสเคสหลัก และมียูสเคสที่เป็นองค์ประกอบหรือตัวเลือก 2 รูปแบบ ได้แก่ Bill Payment (ชำระโดยใช้ใบแจ้งยอดชำระ) และ Direct Payment (ชำระอัตโนมัติโดยตัดผ่านบัญชีธนาคาร) เป็นต้น แสดงความสัมพันธ์ ดังรูปที่ 2.15



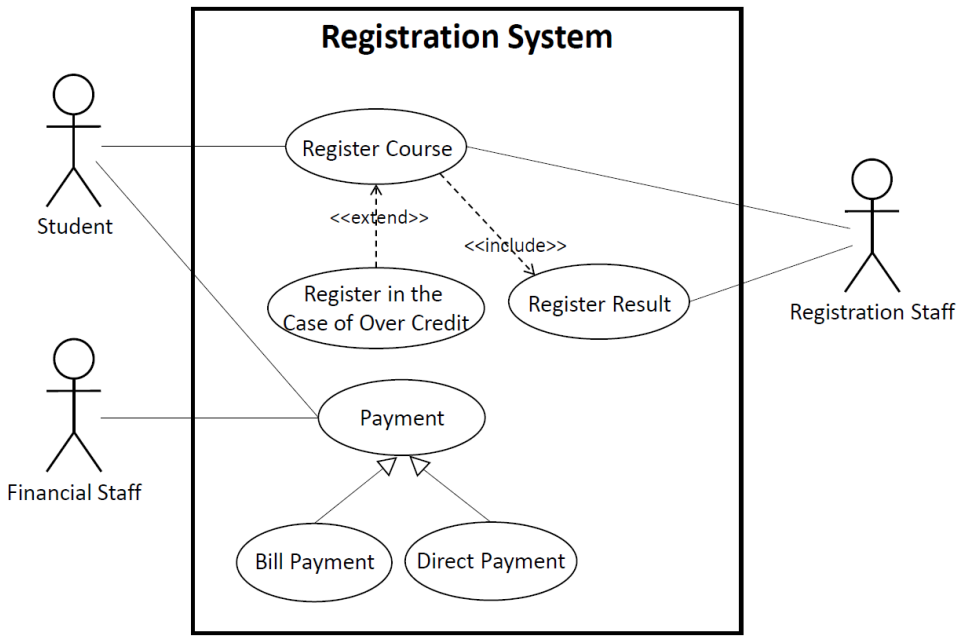
รูปที่ 2.15 ความสัมพันธ์แบบองค์ประกอบหรือการจำแนกประเภท

7) System Name เป็นเส้นแบ่งขอบเขตระหว่างยูสเคสกับผู้กระทำ โดยใช้รูปสี่เหลี่ยมเป็นสัญลักษณ์พร้อมทั้งเขียนชื่อของระบบไว้ด้านบนของกรอบสี่เหลี่ยมซึ่งมีความจำเป็นและสำคัญมาก ดังรูปที่ 2.16



รูปที่ 2.16 สัญลักษณ์แทนชื่อระบบ

8) จากสัญลักษณ์ของยูสเคสที่อธิบายข้างต้น สามารถออกแบบแผนภาพยูสเคสของระบบลงทะเบียนเรียนประกอบด้วย 4 ยูสเคส และ 3 ผู้กระทำ เมื่อพิจารณาแล้วยูสเคส “Register Result” จะถูกเรียกใช้งานเมื่อ Student มีการ “Register Course” โดย Registration Staff จะเป็นผู้ตรวจสอบหลังจากได้ทำการลงทะเบียนแล้ว ซึ่งเป็นความสัมพันธ์แบบเรียกใช้งานหรือ Include Relationship สำหรับยูสเคส “Register in the Case of Over Credit” จะเป็นความสัมพันธ์ในรูปแบบ Extend Relationship ซึ่งเป็นส่วนขยายการทำงานของยูสเคสหลัก คือ “Register Course” และจะไม่เกิดขึ้นทุกครั้งแต่จะเกิดเฉพาะเมื่อมีการลงทะเบียนเรียนเกินหน่วยกิตที่กำหนดเท่านั้น ซึ่งแตกต่างจาก Include Relationship ที่จะเกิดขึ้นทุกครั้งเมื่อมีการเรียกใช้งาน ส่วนยูสเคส “Payment” จะมียูสเคสที่เป็นองค์ประกอบ 2 ยูสเคส ได้แก่ “Bill Payment” และ “Direct Payment” ซึ่งเป็นรูปแบบของการชำระเงินและถูกใช้งานโดย Student และ Financial Staff ดังรูปที่ 2.17



รูปที่ 2.17 แผนภาพยูสเคสระบบลงทะเบียนเรียน

2.1.6 เอกซ์เอ็มแอล (XML)

เอกซ์เอ็มแอล (XML) ย่อมาจาก Extensible Markup Language เป็นภาษา metalanguage ที่ใช้สำหรับการเขียนเอกสาร markup (markup document) ซึ่งเป็นภาษามาตรฐานในการอธิบายข้อมูลและโครงสร้างการจัดเก็บข้อมูลแบบลำดับชั้น (hierarchy) โดยมีการนำแนวคิดของ metadata มาใช้ในการกำหนดหน้าที่ของข้อมูลแต่ละประเภท ทำให้มองเห็นโครงสร้างของเอกสารได้อย่างชัดเจนขึ้น (Antoniou and Van Harmelen, 2004) XML ถูกพัฒนาต่อยอดมาจากภาษา SGML (Standard Generalized Markup Language) ที่มีการดัดแปลงโครงสร้างให้สามารถเข้าใจได้ง่ายและมีความซับซ้อนน้อยลง เป็นภาษาที่ถูกนำมาใช้ในการแลกเปลี่ยนข้อมูลของแอปพลิเคชันต่าง ๆ ระหว่างเครื่องคอมพิวเตอร์ที่มีความแตกต่างกันผ่านเครือข่ายอินเทอร์เน็ต นอกจากนี้ XML ยังเป็นภาษาที่มีความยืดหยุ่นเนื่องจากผู้ใช้สามารถกำหนดและตั้งค่า metadata หรือ tags ให้เหมาะกับเอกสารเฉพาะที่ต้องการได้อย่างอิสระ และยังสามารถเพิ่ม metadata ในภายหลังได้โดยไม่มีผลกระทบต่อโปรแกรมอื่น ๆ มีรูปแบบโครงสร้างที่สามารถเรียกใช้งานได้อย่างอิสระจึงทำให้การจัดการข้อมูล หรือเรียกใช้ข้อมูลจากแอปพลิเคชันต่าง ๆ เป็นไปในมาตรฐานเดียวกันทำให้สามารถอ่านและประมวลผลเอกสารได้ง่าย สะดวก และรวดเร็ว ในการสร้าง XML จะมีการกำหนดกฎและโครงสร้างของเนื้อหาภายในเอกสารซึ่งมีรายละเอียด ดังนี้

1) Root Element หรือ อิลีเมนต์รากเป็นอิลีเมนต์ระดับบนสุดซึ่งจะมีเพียงอิลีเมนต์เดียวเท่านั้น โดยอิลีเมนต์ย่อย (sub-elements) อื่น ๆ จะอยู่ภายใต้อิลีเมนต์นี้ การซ้อนกันของอิลีเมนต์จะต้องเป็นลำดับต่อเนื่องกันและทุกอิลีเมนต์ต้องมี tag เปิด และ tag ปิด โดยชื่อของอิลีเมนต์ภายใน tag จะต้องเป็นชื่อเดียวกันทุกประการ เนื่องจากเป็น case sensitive ของการตั้งชื่อ โดยอิลีเมนต์รากคือ <Bookstore>...</Bookstore> และมีอิลีเมนต์อื่น ๆ อยู่ภายใน ดังรูปที่ 2.18

```
<Bookstore>
    <book>
        <title></title>
    </book>
</Bookstore>
```

รูปที่ 2.18 ตัวอย่างการประกาศ Element

2) Attribute เป็นการระบุค่า (value) หรือลักษณะพิเศษของอิลีเมนต์ เช่น สี ขนาดตัวอักษร รูปแบบตัวอักษร เป็นต้น โดยค่าของอิลีเมนต์จะอยู่ภายใต้เครื่องหมาย ‘’ หรือ “” โดยให้ category เป็นแอตทริบิวต์ของ children และมีค่าของข้อมูลอยู่ภายในอิลีเมนต์ ดังรูปที่ 2.19

```
<book category="children">
      <title>Harry Potter</title>
      <author>J K. Rowling</author>
      <year>2005</year>
      <price>229.99</price>
</book>
```

รูปที่ 2.19 ตัวอย่างการประกาศ Attribute

3) โครงสร้างของเอกสาร XML ประกอบด้วย 2 ส่วนหลัก ๆ คือ โปรล็อค (prolog) ซึ่งเป็นการประกาศหัวของเอกสาร และอิลีเมนต์เอกสาร (Document Element) ซึ่งเป็นส่วนของเนื้อหาภายในเอกสาร โดยตัวอย่างของ XML Schema ดังรูปที่ 2.20

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="Item">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Book" type="xs:string"/>
                <xs:element name="CD" type="xs:string"/>
                <xs:element name="DVD" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Prolog

Document Element

รูปที่ 2.20 ตัวอย่างโครงสร้าง XML Schema

จากรูปที่ 2.20 มีรายละเอียดของโครงสร้าง XML Schema ในแต่ละส่วน ดังนี้

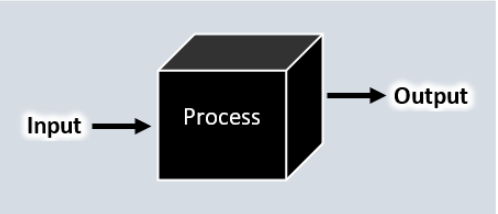
3.1) Prolog เป็นโครงสร้างส่วนแรกของเอกสาร ภายใน Prolog จะมีองค์ประกอบที่เรียกว่า XML Declaration หรือการประกาศ XML เป็นการระบุเวอร์ชันของ XML Specification โดยพิจารณาใช้เวอร์ชัน 1.0 ซึ่งควรมีการประกาศไว้เสมอ หรืออาจจะมีการประกาศ encoding declaration เพื่อระบุการเข้ารหัสตัวอักษร เช่น UTF-8 ซึ่งจะเป็นรูปแบบของตัวอักษรส่วนใหญ่ในภาษาอังกฤษรวมถึงภาษาไทยด้วย

3.2) Document Element เป็นส่วนของเนื้อหา (content) ภายในเอกสาร ซึ่งประกอบไปด้วย tag เนื้อหาภายใน tag และ element ต่าง ๆ เนื้อหาส่วนนี้จะเป็น single element ที่สามารถประกอบไปด้วย sub-elements และ external entities

2.1.7 เทคนิคการทดสอบแบบกล่องดำ (Black-box Testing Techniques)

เทคนิคการทดสอบแบบกล่องดำ เป็นการทดสอบฟังก์ชันการทำงานของระบบโดยโปรแกรมจะถูกมองในรูปของกล่องดำเพื่อป้องกันไม่ให้เห็นกลไกภายในสำหรับการทดสอบข้อมูลการทดสอบแบบกล่องดำนั้นนักทดสอบระบบ (tester) จะทำการศึกษาเรียนรู้เฉพาะสถาปัตยกรรม

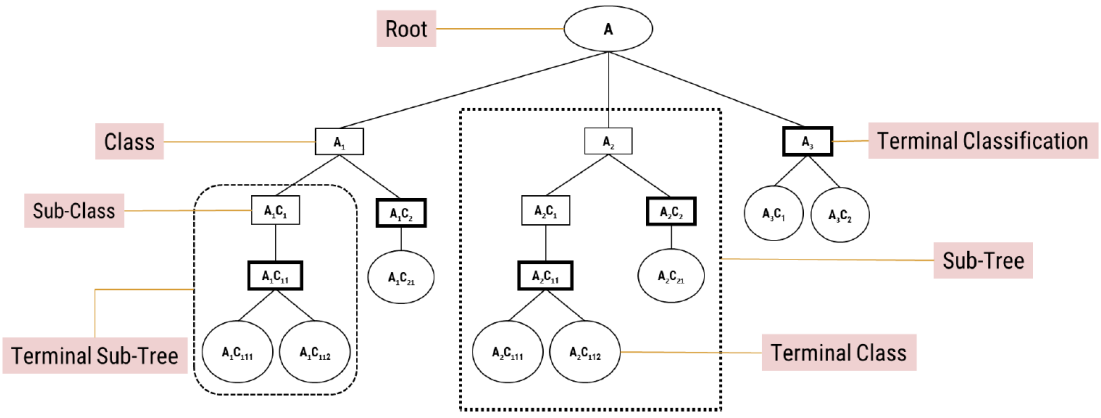
ของระบบเท่านั้นไม่จำเป็นต้องเข้าถึงรหัสคำสั่งของซอฟต์แวร์ (source code) (Khan, 2011) ซึ่งการทดสอบแบบกล่องดำไม่คำนึงถึงคำสั่งภายในโปรแกรมจะทำการทดสอบฟังก์ชันการทำงานต่าง ๆ ของโปรแกรมตามความต้องการของระบบ (requirement) ที่มีอยู่ โดยจะทดสอบจากค่าข้อมูลผลลัพธ์ (output) ที่ได้จากข้อมูลนำเข้า (input) ซึ่งค่าทั้งสองต้องมีความสอดคล้องกัน สามารถแสดงกระบวนการทำงานของการทดสอบแบบกล่องดำ ดังรูปที่ 2.21



รูปที่ 2.21 กระบวนการทำงานของการทดสอบแบบกล่องดำ

2.1.7.1 ต้นไม้การจำแนก (Classification Tree Method: CTM)

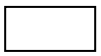
การจำแนกโดยใช้ต้นไม้การจำแนก (Grochtmann, et al., 1993) เป็นวิธีการหนึ่งของเทคนิคการทดสอบแบบกล่องดำ (black-box testing) ซึ่งวิธีการจำแนกโดยใช้ต้นไม้การจำแนกจะทำการแบ่งโดเมนของข้อมูลนำเข้า (input domain) ออกเป็นหมวดหมู่และแตกย่อยในแต่ละหมวดหมู่ลงไปเป็นชั้น ๆ (class) โดยในการแบ่งแบบต้นไม้การจำแนกจะแตกกิ่งก้านออกไปเรื่อย ๆ ซึ่งมีลักษณะคล้ายกับต้นไม้จึงเรียกว่าต้นไม้การจำแนก (classification tree) การสร้างกรณีทดสอบด้วยวิธีนี้จะเป็นการรวมโหนดแต่ละโหนดของกิ่งต้นไม้เข้าด้วยกัน เพื่อพิจารณาเลือกกรณีทดสอบที่เป็นไปได้ทำให้การสร้างกรณีทดสอบมีความครอบคลุมยิ่งขึ้น โดยในแต่ละกรณีทดสอบจะประกอบไปด้วยโหนดของหมวดหมู่แต่ละหมวดซึ่งใน 1 โหนด จะได้เป็น 1 กรณีทดสอบ (Ramadoss, et al., 2011) ในการจำแนกแบบต้นไม้การจำแนกสามารถอธิบาย ดังรูปที่ 2.22



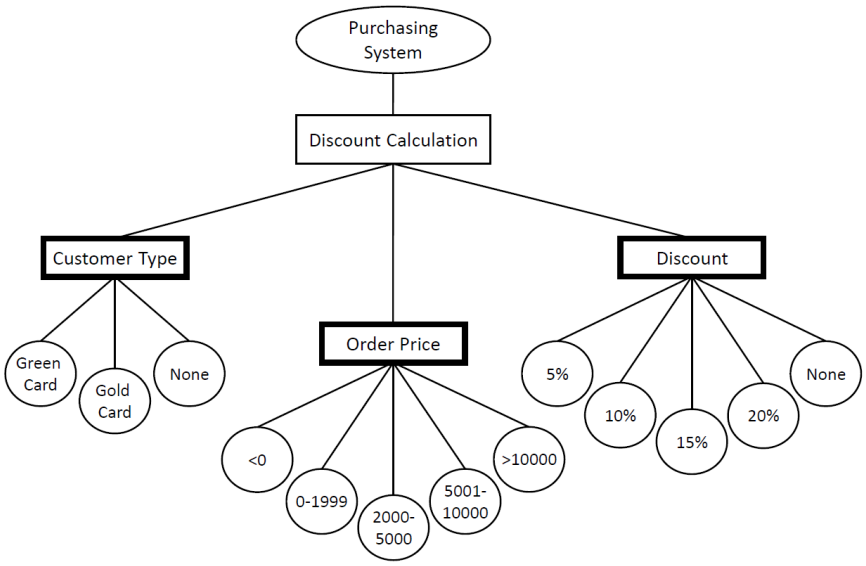
รูปที่ 2.22 แผนภาพการจำแนกแบบต้นไม้การจำแนก

จากรูปที่ 2.22 แสดงโครงสร้างการจำแนกโดยใช้ต้นไม้การจำแนก สามารถอธิบายรายละเอียดและสัญลักษณ์ที่ใช้ ดังตารางที่ 2.3

ตารางที่ 2.3 อธิบายสัญลักษณ์การจำแนกโดยใช้ต้นไม้การจำแนก

สัญลักษณ์	ความหมาย
Root	ภาพรวมของระบบงานหลัก
 Class	คลาสที่สามารถแตกย่อยลงมาภายในโดเมนเดียวกันของระบบงาน
Terminal Classification	คลาสของข้อมูลที่สามารถแตกย่อยเป็นลำดับสุดท้าย
Terminal Class	ข้อมูลชุดสุดท้ายของกิ่งก้านนั้น ๆ ที่ไม่สามารถแตกย่อยได้อีก ซึ่งจะใช้เป็นข้อมูลในการสร้างกรณีทดสอบ
Sub-Tree	ข้อมูลที่ถูกจัดหมวดหมู่ให้อยู่ในประเภทเดียวกัน สามารถแตกย่อยรายละเอียดลงไปได้อีก ซึ่งเป็นเรื่องราวที่ต่อเนื่องและมีความสอดคล้องกัน
Terminal Sub-Tree	ข้อมูลที่ถูกจัดหมวดหมู่ให้อยู่ในประเภทเดียวกัน สามารถแตกย่อยรายละเอียดที่จะจัดเก็บเป็นลำดับสุดท้าย เพื่อใช้ในการสร้างกรณีทดสอบ

การจำแนกโดยต้นไม้การจำแนกมีวัตถุประสงค์เพื่อจัดหมวดหมู่ของข้อมูลให้อยู่ภายใต้โดเมนเดียวกัน เป็นเรื่องราวที่มีความต่อเนื่องกัน จากรูปแบบต้นไม้การจำแนกข้างต้น สามารถแสดงตัวอย่างต้นไม้การจำแนกของระบบการจัดซื้อสินค้า (Purchasing System) ในส่วนของฟังก์ชันคำนวณส่วนลดของการซื้อสินค้า โดยส่วนลดที่ได้รับในแต่ละช่วงราคาสินค้าจะแตกต่างกันไปตามส่วนลดของลูกค้าแต่ละประเภท และต้องมียอดการสั่งซื้อขั้นต่ำตั้งแต่ 2,000 บาทขึ้นไป เช่น ลูกค้าที่ถือบัตร Green Card มียอดการสั่งซื้อตั้งแต่ 2,000 บาทขึ้นไป จะได้รับส่วนลด 10% เป็นต้น แสดงตัวอย่างต้นไม้การจำแนก ดังรูปที่ 2.23



รูปที่ 2.23 ตัวอย่างต้นไม้การจำแนก

2.1.7.2 การแบ่งชั้นสมมูล (Equivalence Class Partitioning: ECP)

การแบ่งชั้นสมมูล (Jorgensen, 2003) เป็นอีกเทคนิคหนึ่งของการทดสอบแบบกล่องดำ (black-box testing) ซึ่งจะทำการแบ่งข้อมูลการทดสอบออกเป็นส่วน ๆ เพื่อทดสอบทั้งส่วนที่ถูกต้องและไม่ถูกต้อง โดยมีวัตถุประสงค์ในการลดจำนวนข้อมูลทดสอบเป็นหลัก การทดสอบโดยใช้ชั้นสมมูลเป็นวิธีการออกแบบกรณีทดสอบเพื่อตรวจหาข้อผิดพลาดโดยนำข้อมูลมาแบ่งกลุ่มเป็นชั้นหรือเป็นช่วงที่เรียกว่าชั้นสมมูล (equivalence class) ซึ่งข้อมูลแต่ละชั้นจะไม่ซ้ำกันตามหลักเกณฑ์ในการกำหนดชั้นสมมูล โดยจะพิจารณาทั้งค่าข้อมูลที่มีความถูกต้อง (valid) และค่าข้อมูลที่ไม่ถูกต้อง (invalid) ตัวอย่างแสดงการแบ่งชั้นสมมูลของช่วงราคาสินค้าในการคำนวณส่วนลดของลูกค้าแต่ละประเภท ดังรูปที่ 2.24

Invalid	Invalid	Valid	Valid	Valid
<0	**0-1999**	**2000-5000**	**5001-10000**	**>10000**

รูปที่ 2.24 ตัวอย่างการแบ่งชั้นสมมูล

จากรูปที่ 2.24 เป็นช่วงของราคาสินค้าที่จะได้รับส่วนลดเมื่อลูกค้าทำการสั่งซื้อ ซึ่งแต่ละช่วงราคาจะได้รับส่วนลดที่แตกต่างกัน สามารถแบ่งชั้นสมมูลของช่วงข้อมูลที่ถูกต้อง (Valid) ได้ 3 ช่วง ตัวอย่างเช่น ช่วงราคาตั้งแต่ 2,000-5,000 เป็นต้น และช่วงของข้อมูลที่ไม่ถูกต้อง (Invalid) 2 ช่วง ได้แก่ ช่วงราคาน้อยกว่า 2,000 และช่วงราคาติดลบ

ตัวอย่าง: โปรแกรมคำนวณส่วนลดของการซื้อสินค้า โดยลูกค้าจะมี 2 ประเภท ได้แก่ ลูกค้าที่ถือบัตร Green Card และ Gold Card จะได้รับส่วนลดเมื่อมียอดการสั่งซื้อขั้นต่ำตั้งแต่ 2,000 บาทขึ้นไป ในการสร้างกรณีทดสอบโดยใช้ชั้นสมมูลสามารถจำแนกวิธีการทดสอบออกเป็น 4 รูปแบบ (Jorgensen, 2003) โดยรายละเอียดในแต่ละรูปแบบสามารถอธิบายได้ ดังนี้

กำหนดให้ชั้นสมมูลที่มีค่าถูกต้องประกอบไปด้วย

V_1 = {Customer Type: Customer Type = “Green Card”}

V_2 = {Customer Type: Customer Type = “Gold Card”}

V_3 = {Order Price: 2000 <= Order Price <= 5000}

V_4 = {Order Price: 5001 <= Order Price <= 10000}

V_5 = {Order Price: Order Price >= 10001}

และกำหนดให้ชั้นสมมูลที่มีค่าไม่ถูกต้องประกอบไปด้วย

IV_6 = {Customer Type: Customer Type = “None”}

IV_7 = {Order Price: 0 <= Order Price <= 1999}

IV_8 = {Order Price: Order Price < 0}

1) การทดสอบโดยชั้นสมมูลแบบวีคนอร์มัล (Weak Normal)

การทดสอบโดยใช้ชั้นสมมูลแบบวีคนอร์มอลจะพิจารณาจากสมมติฐานที่ว่า ความล้มเหลวของโปรแกรมจะไม่เกิดจากข้อผิดพลาด 2 ที่พร้อม ๆ กัน แต่จะเกิดจากข้อผิดพลาดเดียวเท่านั้น (single fault assumption) และการพิจารณาสร้างข้อมูลทดสอบจะสร้างจากชั้นสมมูลเพียงชั้นใดชั้นหนึ่งของทุกตัวแปรที่ถูกต้อง (valid) เพราะฉะนั้น จากตัวอย่างที่นำมาประกอบการอธิบายจะพิจารณาจากช่วงข้อมูลของประเภทลูกค้าและยอดการสั่งซื้อทุกช่วง ดังตารางที่ 2.4

ตารางที่ 2.4 ตัวอย่างกรณีทดสอบแบบวีคนอร์มัล

Test Case ID	Customer Type	Order Price	Valid/Invalid
TC-ID01	Green Card	2500	Valid
TC-ID02	Green Card	7000	Valid
TC-ID03	Gold Card	30000	Valid

2) การทดสอบโดยชั้นสมมูลแบบสตรองนอร์มัล (Strong Normal)

การทดสอบโดยใช้ชั้นสมมูลแบบสตรองนอร์มอลจะพิจารณาจากสมมติฐานที่ว่า ความล้มเหลวของโปรแกรมจะเกิดจากข้อผิดพลาด 2 ที่พร้อม ๆ กัน (multiple fault assumption) โดยกรณีทดสอบที่สร้างจะครอบคลุมทุกชั้นสมมูลที่เป็นไปได้ทั้งหมด หรือทุก ๆ ผลคูณคาร์ทีเซียน (Cartesian Product) แต่เนื่องจากการออกแบบชั้นสมมูลแบบสตรองนอร์มัลจะพิจารณาเฉพาะชั้นสมมูลที่ถูกต้อง (valid) เท่านั้น ดังนั้นการสร้างกรณีทดสอบรูปแบบนี้จะครอบคลุมทุกชั้นสมมูลที่ถูกต้องทั้งหมด ซึ่งจะได้กรณีทดสอบทั้งหมด 6 กรณีทดสอบ ดังตารางที่ 2.5

ตารางที่ 2.5 ตัวอย่างกรณีทดสอบแบบสตรองนอร์มัล

Test Case ID	Customer Type	Order Price	Valid/Invalid
TC-ID01	Green Card	4800	Valid
TC-ID02	Green Card	9990	Valid
TC-ID03	Green Card	30000	Valid
TC-ID04	Gold Card	4800	Valid
TC-ID05	Gold Card	9990	Valid
TC-ID06	Gold Card	30000	Valid

3) การทดสอบโดยชั้นสมมูลแบบวีคโรบัส (Weak Robust)

การทดสอบโดยใช้ชั้นสมมูลแบบวีคโรบัสจะพิจารณาจากสมมติฐานที่ว่า ความล้มเหลวของโปรแกรมจะไม่เกิดจากข้อผิดพลาด 2 ที่พร้อม ๆ กัน แต่จะเกิดจากข้อผิดพลาดเดียวเท่านั้น (single fault assumption) ซึ่งในการสร้างกรณีทดสอบจะพิจารณาทุกชั้นสมมูลที่เป็นไปได้และเป็นไปไม่ได้ทั้งที่ถูกต้อง (valid) และไม่ถูกต้อง (invalid) โดยการสร้างกรณีทดสอบสำหรับกรณีที่ถูกต้องจะสร้างเหมือนกันกับแบบวีคนอร์มัล แต่จะมีการเพิ่มเติมการสร้างกรณีทดสอบที่ไม่ถูกต้อง โดยจะสร้างจากชั้นสมมูลที่ไม่ถูกต้องเพียงชั้นเดียวของทุกตัวแปรเท่านั้น และจะต้องเป็นค่าใดค่าหนึ่งที่ไม่ซ้ำกัน เนื่องจากความล้มเหลวจะเกิดจากข้อผิดพลาดเดียวเท่านั้น ดังตารางที่ 2.6

ตารางที่ 2.6 ตัวอย่างกรณีทดสอบแบบวีคโรบัส

Test Case ID	Customer Type	Order Price	Valid/Invalid
TC-ID01	Green Card	2500	Valid
TC-ID02	Green Card	7000	Valid
TC-ID03	Gold Card	30000	Valid
TC-ID04	Green Card	500	Invalid
TC-ID05	Gold Card	1999	Invalid
TC-ID06	None	-200	Invalid

4) การทดสอบโดยชั้นสมมูลแบบสตรองโรบัส (Strong Robust)

การทดสอบโดยใช้ชั้นสมมูลแบบสตรองโรบัสจะพิจารณาจากสมมติฐานที่ว่า ความล้มเหลวของโปรแกรมจะเกิดจากข้อผิดพลาด 2 ที่พร้อม ๆ กัน (multiple fault assumption) โดยกรณีทดสอบที่สร้างจะครอบคลุมทุกชั้นสมมูลที่เป็นไปได้และเป็นไปไม่ได้ทั้งหมด

หรือทุก ๆ ผลคูณคาร์ทีเซียน (Cartesian Product) โดยชั้นสมมูลที่ถูกต้องจะสร้างเหมือนกันกับแบบสตรองนอร์มัล แต่การสร้างกรณีทดสอบที่ไม่ถูกต้องจะพิจารณาทุกชั้นสมมูลของทุก ๆ ตัวแปรทั้งหมด

โดยในงานวิจัยนี้ผู้วิจัยได้นำการทดสอบโดยชั้นสมมูลแบบสตรองโรบัสมาใช้ในการสร้างกรณีทดสอบร่วมกับต้นไม้การจำแนกเพื่อให้กรณีทดสอบมีความครอบคลุมทั้งข้อมูลทดสอบที่ถูกต้องและไม่ถูกต้องในทุกช่วงของชั้นสมมูลที่เป็นไปได้ ดังตารางที่ 2.7

ตารางที่ 2.7 ตัวอย่างกรณีทดสอบแบบสตรองโรบัส

Test Case ID	Customer Type	Order Price	Valid/Invalid
TC-ID01	Green Card	4800	Valid
TC-ID02	Green Card	9990	Valid
TC-ID03	Green Card	30000	Valid
TC-ID04	Gold Card	4800	Valid
TC-ID05	Gold Card	9990	Valid
TC-ID06	Gold Card	30000	Valid
TC-ID07	Green Card	1999	Invalid
TC-ID08	Green Card	-12500	Invalid
TC-ID09	Gold Card	1999	Invalid
TC-ID10	Gold Card	-12500	Invalid
TC-ID11	None	4800	Invalid
TC-ID12	None	9990	Invalid
TC-ID13	None	30000	Invalid
TC-ID14	None	1999	Invalid
TC-ID15	None	-12500	Invalid

2.1.7.3 การรวมกันของชั้นสมมูลและต้นไม้การจำแนก (Combination of Equivalence and Classification Tree Method: CCTM)

การรวมกันของชั้นสมมูลและต้นไม้การจำแนก เป็นการรวมกันของ 2 เทคนิคการทดสอบแบบกล่องดำ (black-box testing) ซึ่งช่วยให้การสร้างกรณีทดสอบมีความครอบคลุมทุกกรณีของการทดสอบ และยังสามารถลดจำนวนกรณีทดสอบในการทดสอบซอฟต์แวร์ให้น้อยลงได้ (Ramadoss, et al., 2011) ในการจำแนกโดยใช้ต้นไม้การจำแนกจะใช้ข้อมูลของโหนดที่มีความแตกต่างกันของข้อมูลในคลาสที่เรียกว่าเทอร์มินัลคลาสซึ่งเป็นโหนดลำดับสุดท้ายของต้นไม้การจำแนกที่ไม่

สามารถแตกย่อยได้แล้ว ซึ่งในคลาสนี้จะมีลักษณะข้อมูลซึ่งเป็นค่าของตัวแปรที่สามารถนำไปแบ่งชั้นสมมูลได้ โดยในการสร้างกรณีทดสอบจากการแบ่งชั้นสมมูลสามารถแบ่งได้เป็น 4 ลักษณะ โดยจะแสดงตัวอย่างจากข้อมูลของนักศึกษา ดังตารางที่ 2.8

ตารางที่ 2.8 แนวทางการสร้างกรณีทดสอบจากการแบ่งชั้นสมมูล

ค่าชั้นสมมูล	**ตัวอย่าง**	**ค่าที่เป็นไปได้**
ค่าที่เป็นช่วง	ช่วงของคะแนนในการคำนวณเกรด	0-49, 50-59, 60-69, 70-79, 80-100 เป็นต้น
ค่าที่เป็นจำนวน	หน่วยกิตการศึกษา	3 หน่วยกิต, 6 หน่วยกิต เป็นต้น
ค่าที่เป็นเซต หรือกลุ่มที่ถูกกำหนดไว้	สถานะของนักศึกษา	ปกติ (normal), วิกฤต (critical), รอพินิจครั้งที่ 1 (1-Pro), รอพินิจครั้งที่ 2 (2-pro), รอพินิจครั้งที่ 3 (3-pro) เป็นต้น
ค่าที่จำเป็นต้องมี	ข้อมูลนักศึกษา	ชื่อ-นามสกุล, สาขา, คณะ เป็นต้น

2.2 วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

2.2.1 งานวิจัย “การสร้างกรณีทดสอบด้วยออนโทโลยีสำหรับการทดสอบแบบกล่องดำ (Ontology Based Test Case Generation for Black Box Testing)”

งานวิจัยของ Ul Haq และ Qamar (Ul Haq and Qamar, 2019) ได้นำเสนอการสร้างกรณีทดสอบจากออนโทโลยีแบบอัตโนมัติสำหรับการทดสอบแบบกล่องดำ โดยนำออนโทโลยีมาใช้แทนความหมายของเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ในการสร้างกรณีทดสอบ โดยใช้ Web Ontology Language (OWL) ในการแทนความหมายของออนโทโลยีสำหรับข้อกำหนดความต้องการที่เป็นบทภาพเหตุการณ์การทำงานของการทดสอบ เช่น ผู้เกี่ยวข้องที่มีส่วนได้ส่วนเสียกับระบบ (stakeholder) เงื่อนไขก่อนดำเนินการ (pre-condition) เงื่อนไขหลังดำเนินการ (post-condition) เป็นต้น โดยจะเป็นบทภาพเหตุการณ์การทดสอบที่เป็นส่วนสำคัญหลัก ๆ ของการทดสอบซอฟต์แวร์ ซึ่งในงานวิจัยนี้ยังไม่ครอบคลุมส่วนการทำงานทั้งหมดของการทดสอบ

ในงานวิจัยนี้เป็นการนำออนโทโลยีมาใช้แทนความหมายโครงสร้างเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) โดยมีการแทนในรูปแบบโครงข่ายความหมายในส่วนที่เป็นกระบวนการดำเนินงานและผู้ที่เกี่ยวข้องกับระบบ สำหรับงานวิจัยที่ผู้วิจัยนำเสนอจะใช้ออนโทโลยีเพื่อแทนความหมายของความต้องการในเอกสารเพื่อนำไปใช้ในการสร้างกรณีทดสอบ และในงานวิจัยนี้ไม่ได้กล่าวถึงความครอบคลุมสำหรับการสร้างกรณีทดสอบและข้อมูลทดสอบ

2.2.2 งานวิจัย “การประยุกต์ใช้กฎการอนุมานกับออนโทโลยีความต้องการซอฟต์แวร์ในการสร้างกรณีทดสอบซอฟต์แวร์ (Application of Inference Rules to a Software Requirements Ontology to Generate Software Test Cases)”

งานวิจัยของ Tarasov และคณะ (Tarasov, et al., 2016) ได้นำเสนอแนวทางในการสร้างกรณีทดสอบด้วยออนโทโลยีและกฎการอนุมาน (inference rule) จากเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) และความรู้เพิ่มเติมเกี่ยวกับส่วนประกอบของระบบซอฟต์แวร์ที่ต้องการพัฒนา กฎการอนุมานถูกสร้างขึ้นจากการตรวจสอบเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ที่มีอยู่และการสอบถามเพิ่มเติมจากนักทดสอบระบบที่มีประสบการณ์ โดยกฎการอนุมานและออนโทโลยีถูกนำมาใช้ในการแทนความหมายของโครงสร้างเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ในส่วนที่เป็นบทภาพเหตุการณ์ในการสร้างกรณีทดสอบ โดยได้มีการแปลงออนโทโลยีจากไวยกรณ์แบบ Web Ontology Language (OWL) เป็นไวยกรณ์ Prolog เพื่อสร้างกรณีทดสอบ โดยกรณีทดสอบที่ได้จากการดำเนินการจะเป็นข้อมูลนำเข้าเพื่อใช้ทดสอบ (test input) ขั้นตอนการทดสอบ (test procedure) และผลลัพธ์ที่คาดหวัง (expected test results) จากขั้นตอนของการทดสอบนั้น ๆ

งานวิจัยนี้มีการแทนความหมายในรูปแบบของสคริปต์ โดยนำออนโทโลยีมาใช้ในการแทนโครงสร้างของเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ในส่วนของบทภาพเหตุการณ์สำหรับการสร้างกรณีทดสอบ ซึ่งแตกต่างจากงานวิจัยที่ผู้วิจัยนำเสนอ ที่มีการใช้ออนโทโลยีในการแทนความหมายของข้อกำหนดความต้องการในเอกสารเพื่อนำมาใช้ในการสร้างกรณีทดสอบ และข้อมูลทดสอบให้มีความครอบคลุมสำหรับการทดสอบซอฟต์แวร์

2.2.3 งานวิจัย “การใช้รูปแบบออนโทโลยีสำหรับอ้างอิงการทดสอบซอฟต์แวร์ (Using Ontology Patterns for Building a Reference Software Testing Ontology)”

งานวิจัยของ Souza และคณะ (Souza, et al., 2013) ได้นำเสนอ ROoST (Reference Ontology on Software Testing) เป็นการพัฒนาออนโทโลยีเพื่ออ้างอิงการทดสอบซอฟต์แวร์ในการสร้างรูปแบบออนโทโลยีใหม่ ๆ จากซอฟต์แวร์ SP-OPL (Software Process Ontology Pattern Language) โดยจะกล่าวถึงวิธีการพัฒนา ROoST และนำเสนอส่วนต่าง ๆ ของ ROoST ที่เกี่ยวข้องกับกิจกรรมกระบวนการทดสอบซอฟต์แวร์และขั้นตอนการทดสอบที่ครอบคลุมกระบวนการทดสอบซอฟต์แวร์ รวมถึงเทคนิคการทดสอบสำหรับการออกแบบกรณีทดสอบและสภาพแวดล้อมในการทดสอบซอฟต์แวร์ เพื่อเป็นแนวทางสำหรับนักพัฒนาและผู้ที่เกี่ยวข้องในการลดข้อผิดพลาดที่อาจเกิดขึ้นและสามารถออกแบบกรณีทดสอบได้อย่างมีประสิทธิภาพ

งานวิจัยนี้ได้นำออนโทโลยีมาใช้ในการแทนกิจกรรมการทดสอบและขั้นตอนการทดสอบเพื่อเชื่อมโยงและเก็บองค์ความรู้ในการสร้างความเข้าใจสำหรับความต้องการซอฟต์แวร์ โดยมีการแทนความรู้ในรูปแบบของเฟรม ซึ่งในงานวิจัยนี้ได้กล่าวถึงการพัฒนาออนโทโลยีตามโครงสร้าง

ของเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) เพื่อนำเสนอแนวทางในการทดสอบที่เกี่ยวข้อง เช่น ขั้นตอนการทดสอบ การออกแบบกรณีทดสอบ เทคนิคการทดสอบที่เหมาะสม เป็นต้น เพื่อให้มีความครอบคลุมในกระบวนการทดสอบ แต่ในงานวิจัยที่ผู้วิจัยนำเสนอจะให้ความสำคัญในการสร้างกรณีทดสอบให้มีความครอบคลุมด้วยเทคนิคการทดสอบแบบกล่องดำ ซึ่งแตกต่างจากงานวิจัยนี้ที่ให้ความสำคัญในส่วนความครอบคลุมของกระบวนการทดสอบซอฟต์แวร์ รวมถึงเทคนิคการทดสอบที่มีความเหมาะสมในการสร้างกรณีทดสอบของแต่ละกิจกรรมการทดสอบซฮฟต์แวร์

2.2.4 งานวิจัย “การนำออนโทโลยีมาใช้ในวิศวกรรมความต้องการ (The Use of Ontologies in Requirements Engineering)”

งานวิจัยของ Castañeda และคณะ (Castañeda, et al., 2010) ได้เสนอการนำออนโทโลยไปใช้ในกระบวนการวิศวกรรมซอฟต์แวร์ในโดเมนต่าง ๆ มากมาย เช่น การนำออนโทโลยีไปใช้ในวิศวกรรมความต้องการ (requirement engineering) การดำเนินการของซอฟต์แวร์ (software implementation) และการบำรุงรักษาซอฟต์แวร์ (software maintenance) โดยมีงานวิจัยมากมายที่ใช้ประโยชน์จากออนโทโลยีในด้านวิศวกรรมซอฟต์แวร์และวิศวกรรมความต้องการโดยเฉพาะ รวมถึงวิธีการกำหนดความหมายของข้อมูลและความรู้ภายในโดเมน และลักษณะของวิศวกรรมความต้องการที่เกี่ยวข้องกับการรวบรวมความรู้จากหลาย ๆ แหล่งเข้าด้วยกัน งานวิจัยนี้จึงได้เสนอการทบทวนออนโทโลยีเพื่อนำไปใช้งานในโดเมนต่าง ๆ เหล่านี้ในการจำแนกแนวทางทางด้านวิศวกรรมความต้องการเพื่อชี้แจงวิธีการและเทคนิคต่าง ๆ ในการดำเนินงาน หรือกิจกรรมต่าง ๆ ให้ได้รับประโยชน์จากกระบวนการดังกล่าวในวิศวกรรมความต้องการแบบดั้งเดิมได้อย่างทั่วถึง

งานวิจัยนี้ได้นำเสนอการทบทวนออนโทโลยีเกี่ยวกับแนวคิด และกระบวนการต่าง ๆ เพื่อนำไปใช้ในวิศวกรรมความต้องการ โดยได้นำออนโทโลยีมาใช้ในการแทนความหมายของโครงสร้างเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) และมีการแทนความหมายในรูปแบบของโครงข่ายความหมาย โดยมุ่งเน้นไปที่การให้ข้อมูลเชิงทฤษฎีในส่วนของวิศวกรรมความต้องการในการเก็บรวบรวมความรู้และวิเคราะห์ความต้องการ ซึ่งมีความแตกต่างจากงานวิจัยที่ผู้วิจัยได้นำเสนอตรงที่จะนำความต้องการในเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) มาแทนความหมายด้วยออนโทโลยีเพื่อให้มีโครงสร้างในการนำไปต่อยอดสำหรับนำมาใช้ในการสร้างกรณีทดสอบเพื่อทดสอบซอฟต์แวร์

2.2.5 งานวิจัย “การสร้างกรณีทดสอบด้วยออนโทโลยีสำหรับการทดสอบเว็บเซอร์วิส (Ontology-Based Test Case Generation for Testing Web Services)”

งานวิจัยของ Wang และคณะ (Wang, et al., 2007) ได้เสนอแนวทางในการนำออนโทโลยีไปใช้กับการสร้างกรณีทดสอบแบบอัตโนมัติบนเว็บเซอร์วิส (web service) โดย

มีวัตถุประสงค์เพื่อปรับปรุงกระบวนการทดสอบและข้อมูลทดสอบของข้อกำหนดที่เกี่ยวกับความหมายของเว็บเซอร์วิส โดยใช้ภาษา (Web Ontology Language for Web Service: OWL-S) เพื่ออธิบายตรรกะ (logic) ต่าง ๆ ภายในแอปพลิเคชันของกระบวนการเว็บเซอร์วิส และ Petri-Net ถูกสร้างขึ้นเพื่อเป็นตัวแทนของออนโทโลยีในการแทนความหมายของ IOPE (Inputs, Outputs, Preconditions and Effects) สำหรับการสร้างกรณีทดสอบและข้อมูลทดสอบ เพื่อเพิ่มประสิทธิภาพและการเรียนรู้ของการสร้างกรณีทดสอบแบบอัตโนมัติบนเว็บเซอร์วิส

งานวิจัยนี้ได้เสนอการใช้ Petri-Net เพื่อนำมาใช้ในการสร้างขั้นตอนการทดสอบและออนโทโลยี (OWL-S) ในการสร้างข้อมูลทดสอบเว็บเซอร์วิส โดยตัวอย่างออนโทโลยีที่ใช้ในงานวิจัยนี้ใช้เพื่อแทนความหมายของความต้องการสำหรับเว็บเซอร์วิสในการจองห้องพักของโรงแรม โดยมีการแทนความหมายในรูปแบบของโครงข่ายความหมาย นอกเหนือจากนั้นในงานวิจัยนี้จะกล่าวถึงการสร้างกรณีทดสอบด้วยออนโทโลยีความต้องการซอฟต์แวร์ในกระบวนการทดสอบ แต่ไม่ได้ให้ความสำคัญในส่วนของการสร้างกรณีทดสอบและข้อมูลทดสอบให้มีความครอบคลุมตามความต้องการของผู้ใช้งาน ซึ่งแตกต่างจากงานวิจัยที่ผู้วิจัยได้นำเสนอที่จะให้ความสำคัญในเรื่องของความครอบคลุมของการสร้างกรณีทดสอบสำหรับการทดสอบซอฟต์แวร์

2.2.6 งานวิจัย “การสร้างออนโทโลยีสำหรับการทดสอบซอฟต์แวร์ (Towards the Establishment of an Ontology of Software Testing)”

งานวิจัยของ Barbosa และคณะ (Barbosa, et al., 2006) ได้นำเสนอการพัฒนาแอปพลิเคชัน OntoTest-ontology ของการทดสอบซอฟต์แวร์ที่ได้รับการพัฒนาเพื่อสนับสนุนการนำความรู้มาใช้ใหม่และแบ่งปันความรู้สำหรับการทดสอบภายในองค์กร OntoTest ได้รับการสร้างขึ้นตามมาตรฐาน ISO/IEC 12207 ซึ่งมีการสำรวจด้านต่าง ๆ ที่เกี่ยวข้องกับกิจกรรมการทดสอบ โดยเฉพาะอย่างยิ่งความสนใจในการกำหนดคำศัพท์ที่ใช้กันอย่างแพร่หลายในกระบวนการทดสอบซึ่งจะเป็นประโยชน์ในการพัฒนาเครื่องมือที่ช่วยเพิ่มความสามารถในการสื่อสารระหว่างกันของผู้มีความเกี่ยวข้องภายในองค์กรในกระบวนการทดสอบเพื่อให้เข้าใจตรงกันได้เป็นอย่างดี

งานวิจัยนี้ได้มีการกล่าวถึงการนำความรู้ที่มีอยู่ในโดเมนกลับมาใช้ใหม่และมีการแบ่งปันความรู้สำหรับใช้งานร่วมกันภายในโดเมนเดียวกัน โดยมีการแทนในรูปแบบของกราฟความคิด แต่ถึงอย่างไรก็ตามในงานวิจัยนี้พัฒนาออนโทโลยีโดยเน้นไปที่โครงสร้างของเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) โดยผู้วิจัยให้ความสำคัญในส่วนการแทนความหมายของความต้องการในเอกสารมากกว่าโครงสร้างของเอกสาร เพื่อให้การตีความตรงตามความต้องการที่แท้จริงของผู้ใช้งาน และในงานวิจัยนี้ไม่ได้กล่าวถึงความสำคัญของความครอบคลุมในกระบวนการสร้างกรณีทดสอบและข้อมูลทดสอบ

2.3 เปรียบเทียบงานวิจัยที่เกี่ยวข้อง

ในการเปรียบเทียบระหว่างงานวิจัยที่เกี่ยวข้องกับงานวิจัยที่ผู้วิจัยนำเสนอภายใต้โดเมนของการทดสอบซอฟต์แวร์ ผู้วิจัยจะเปรียบเทียบตามเทคนิคการแทนความรู้ และเทคนิคการสร้างกรณีทดสอบ ซึ่งมีรายละเอียดต่าง ๆ ดังตารางที่ 2.9

ตารางที่ 2.9 ตารางเปรียบเทียบงานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้อง	เทคนิคการแทนความรู้				ความครอบคลุมของกรณีทดสอบ	เทคนิคการสร้างกรณีทดสอบ		โดเมน
	Semantic Network	Frame	Script	Conceptual Graphs		การแบ่งชั้นสมมูล	ต้นไม้การจำแนก	
งานวิจัยที่นำเสนอ	✓	-	-	-	✓	✓	✓	**การทดสอบซอฟต์แวร์**
Ul Haq และ Qamar อธิบายในหัวข้อที่ 2.2.1	✓	-	-	-	-	-	-	การทดสอบซอฟต์แวร์
Tarasov และคณะ อธิบายในหัวข้อที่ 2.2.2	-	-	✓	-	-	-	-	การทดสอบซอฟต์แวร์
Souza และคณะ อธิบายในหัวข้อที่ 2.2.3	-	✓	-	-	✓	-	-	การทดสอบซอฟต์แวร์
Castañeda และคณะ อธิบายในหัวข้อที่ 2.2.4	✓	-	-	-	-	-	-	วิศวกรรมซอฟต์แวร์
Wang และคณะ อธิบายในหัวข้อที่ 2.2.5	✓	-	-	-	-	-	-	การทดสอบเว็บเซอร์วิส
Barbosa และคณะ อธิบายในหัวข้อที่ 2.2.6	-	-	-	✓	-	-	-	การทดสอบซอฟต์แวร์

จากตารางที่ 2.9 ผู้วิจัยได้พิจารณาปัจจัยในการแบ่งกลุ่มการเปรียบเทียบงานวิจัยที่เกี่ยวข้องกับงานวิจัยที่ผู้วิจัยได้นำเสนอไว้ 2 กลุ่ม ดังนี้

2.3.1 แบ่งตามเทคนิคการแทนความรู้

ในการเปรียบเทียบตามเทคนิคของการแทนความรู้นั้น ผู้วัจัยได้นำเสนอรูปแบบของการแทนความหมายในออนโทโลยีซึ่งประกอบไปด้วย การแทนแบบโครงข่ายความหมาย (semantic network) การแทนแบบกรอบ (frame) การแทนแบบสคริปต์ (script) และการแทนแบบกราฟความคิด (conceptual graphs) เพื่อใช้ในการออกแบบและพัฒนาออนโทโลยีภายใต้ขอบเขตของการทดสอบซอฟต์แวร์ จากการศึกษางานวิจัยที่เกี่ยวข้องมีการแทนความรู้ในรูปแบบต่าง ๆ ข้างต้น โดยในงานวิจัยของ Castañeda และคณะ (Castañeda, et al., 2010) งานวิจัยของ Ul Haq และ Qamar (Ul Haq and Qamar, 2019) และงานวิจัยของ Wang และคณะ (Wang, et al., 2007) ได้เสนอการนำออนโทโลยีมาใช้ในวิศวกรรมซอฟต์แวร์สำหรับการทดสอบซอฟต์แวร์ให้มีประสิทธิภาพ โดยในงานวิจัยเหล่านี้ได้ใช้เทคนิคการแทนความรู้ในรูปแบบของโครงข่ายความหมาย ในงานวิจัยของ Souza และคณะ (Souza, et al., 2013) เป็นการแทนความรู้โดยใช้เทคนิคแบบกรอบหรือเฟรมเพื่ออธิบายออนโทโลยีที่พัฒนาขึ้น และได้นำเสนอเครื่องมือในการออกแบบและพัฒนาออนโทโลยีเพื่อนำมาใช้กับงานทางด้านวิศวกรรมความรู้และการทดสอบซอฟต์แวร์ ในงานวิจัยของ Tarasov และคณะ (Tarasov, et al., 2016) ได้นำเสนอแนวทางในการสร้างกรณีทดสอบด้วยออนโทโลยีและกฎการอนุมานโดยใช้เทคนิคการแทนความรู้แบบสคริปต์ ซึ่งกฎการอนุมานและออนโทโลยีจะถูกนำมาใช้ในการแทนความหมายของโครงสร้างเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ในส่วนที่เกี่ยวกับบทภาพเหตุการณ์ในการสร้างกรณีทดสอบสำหรับทดสอบซอฟต์แวร์ ในงานวิจัยของ Barbosa และคณะ (Barbosa, et al., 2006) ได้นำเสนอการออกแบบและพัฒนาออนโทโลยีสำหรับการทดสอบซอฟต์แวร์ ซึ่งมีรูปแบบการแทนแบบกราฟความคิดเพื่อแสดงโดเมนของสิ่งที่สนใจและเพิ่มประสิทธิภาพของออนโทโลยีในการแบ่งปันความรู้แก่ผู้ที่เกี่ยวข้องภายในโดเมน

จากงานวิจัยที่ได้กล่าวมาข้างต้นมีความคล้ายคลึงกับงานวิจัยที่ผู้วิจัยนำเสนอ โดยในงานวิจัยส่วนใหญ่ (Castañeda, et al., 2010; Ul Haq and Qamar, 2019; Wang, et al., 2007) มีรูปแบบการแทนออนโทโลยีแบบโครงข่ายความหมายสำหรับเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ซึ่งเป็นการแทนรูปแบบเดียวกันกับงานวิจัยที่ผู้วิจัยนำเสนอ อย่างไรก็ตามในงานวิจัยเหล่านี้จะออกแบบและพัฒนาออนโทโลยีโดยเน้นไปที่โครงสร้างของเอกสาร รวมถึงขั้นตอนและกระบวนการทำงานต่าง ๆ ภายในการทดสอบซอฟต์แวร์ ซึ่งแตกต่างจากงานวิจัยที่ผู้วิจัยได้นำเสนอ โดยในการออกแบบและพัฒนาออนโทโลยีจะเน้นไปที่ความหมายของความต้องการในเอกสารเป็นสำคัญ เพื่อให้สามารถตีความหมายของความต้องการที่ระบุในเอกสารได้ตรงตามความต้องการที่แท้จริงของผู้ใช้งาน

2.3.2 แบ่งตามเทคนิคการสร้างกรณีทดสอบ

ในการสร้างกรณีทดสอบเพื่อให้มีความครอบคลุม ผู้วิจัยได้พิจารณาเทคนิคที่ใช้ในการสร้างกรณีทดสอบจำนวน 2 เทคนิคของการทดสอบแบบกล่องดำ (black-box testing technique) คือ เทคนิคการแบ่งชั้นสมมูล (ECP) และเทคนิคต้นไม้การจำแนก (CTM) มาใช้ในการเปรียบเทียบกับงานวิจัยอื่น ๆ จากการศึกษางานวิจัยที่เกี่ยวข้องข้างต้น ทำให้ทราบว่างานวิจัยโดยส่วนใหญ่ไม่ได้ให้ความสำคัญในเรื่องความครอบคลุมของการสร้างกรณีทดสอบและข้อมูลทดสอบ ซึ่งจากตารางเปรียบเทียบงานวิจัยที่เกี่ยวข้อง (ตารางที่ 2.9) มีเพียงงานวิจัยของ Souza และคณะ (Souza, et al., 2013) ที่ให้ความสำคัญในเรื่องความครอบคลุมของการสร้างกรณีทดสอบ ในงานวิจัยที่ผู้วิจัยนำเสนอนี้มีการนำเทคนิคการทดสอบแบบกล่องดำจำนวน 2 เทคนิคมารวมเข้าด้วยกันหรือที่เรียกว่า CCTM (Combination of Equivalence and Classification Tree Method) โดยเทคนิคนี้สามารถช่วยลดความซ้ำซ้อนของกรณีทดสอบให้มีจำนวนน้อยลง เหลือเพียงกรณีทดสอบที่จำเป็นและต้องการนำไปใช้งาน อีกทั้งยังช่วยเพิ่มประสิทธิภาพในการสร้างกรณีทดสอบให้มีความครอบคลุมในทุก ๆ เงื่อนไขของการทดสอบมากยิ่งขึ้น

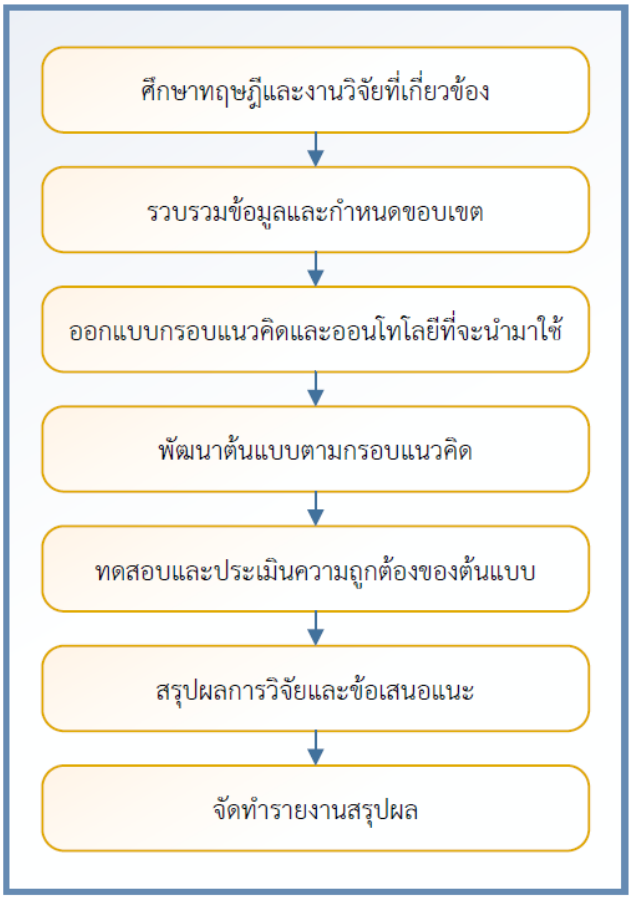
บทที่ 3

ขั้นตอนการดำเนินงานและกรอบแนวคิดของงานวิจัย

การศึกษาวิจัยนี้เป็นการเพิ่มศักยภาพของกระบวนการทดสอบซอฟต์แวร์ (software testing) เกี่ยวกับการสร้างกรณีทดสอบ (test case) โดยได้นำออนโทโลยีมาใช้ในการแทนความหมายของข้อกำหนดความต้องการซอฟต์แวร์ในเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ให้อยู่ในรูปแบบเชิงโครงสร้างก่อนจะนำไปสร้างกรณีทดสอบ และเพื่อเพิ่มประสิทธิภาพให้กับการทดสอบแบบดั้งเดิมโดยจะสร้างกรณีทดสอบจากความต้องการที่เป็นภาษาธรรมชาติซึ่งไม่มีโครงสร้าง ทำให้กรณีทดสอบที่ได้ไม่ตรงตามความต้องการที่แท้จริงของผู้ใช้งาน ดังนั้นกรอบแนวคิดของงานวิจัยนี้มีวัตถุประสงค์ในการสร้างกรณีทดสอบด้วยออนโทโลยีความต้องการซอฟต์แวร์ โดยได้มีการนำเครื่องมือ ROO มาช่วยในการออกแบบและพัฒนาออนโทโลยีทำให้ข้อกำหนดความต้องการมีโครงสร้างที่ชัดเจน โดยเครื่องมือนี้จะมีการนำภาษาธรรมชาติควบคุม (CNL) หรือที่เรียกว่า Rabbit Language มาใช้ในการจัดการภาษาธรรมชาติให้มีโครงสร้างมากยิ่งขึ้น ทำให้การสร้างกรณีทดสอบซอฟต์แวร์ตรงตามความต้องการของผู้ใช้งาน (คำถามงานวิจัย ข้อ 1 และสอดคล้องกับวัตถุประสงค์งานวิจัย ข้อ 1) และเพื่อเพิ่มขีดความสามารถให้การสร้างกรณีทดสอบ ผู้วิจัยได้พิจารณานำเทคนิคการทดสอบแบบกล่องดำ 2 เทคนิค คือ เทคนิคการรวมกันของชั้นสมมูลและต้นไม้การจำแนกมาใช้ในการสร้างกรณีทดสอบให้มีความครอบคลุมในทุก ๆ กรณีของการทดสอบซอฟต์แวร์ (คำถามงานวิจัย ข้อ 2 และสอดคล้องกับวัตถุประสงค์งานวิจัย ข้อ 2) โดยได้มีการอธิบายการทำงานของกรอบแนวคิดไว้ในหัวข้อที่ 3.2 กรอบแนวคิดของงานวิจัย

3.1 ขั้นตอนการดำเนินงานวิจัย

ผู้วัจัยได้มีการดำเนินการและแบ่งขั้นตอนของงานวิจัยออกเป็น 7 ขั้นตอนด้วยกัน ได้แก่ 1) ศึกษาวรรณกรรมและทฤษฎีที่เกี่ยวข้อง 2) รวมรวมข้อมูลและกำหนดขอบเขต 3) ออกแบบกรอบแนวคิดและออนโทโลยีที่จะนำมาใช้ 4) พัฒนาต้นแบบตามกรอบแนวคิด 5) ประเมินความถูกต้องของต้นแบบ 6) สรุปผลการวิจัยและข้อเสนอแนะ 7) จัดทำรายงานสรุปผล แสดงภาพรวมของขั้นตอนและรายละเอียดสำหรับการดำเนินงานวิจัย ดังรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนการดำเนินงานวิจัย

3.1.1 ศึกษาวรรณกรรมและทฤษฎีที่เกี่ยวข้อง

ในกระบวนการเริ่มต้นของการดำเนินงานวิจัยผู้วิจัยได้ทำการศึกษา ค้นคว้า และเก็บรวบรวมข้อมูลที่มีความเกี่ยวข้องกับหัวข้อวิจัยที่ผู้วิจัยได้นำเสนอ และได้มีการทบทวนวรรณกรรมและทฤษฎีต่าง ๆ เพื่อศึกษาแนวทางจากงานวิจัยอื่นที่เกี่ยวข้องและนำมาปรับใช้ในการพัฒนากรอบแนวคิดและการดำเนินการของงานวิจัยนี้

3.1.2 รวบรวมข้อมูลและกำหนดขอบเขต

ขั้นตอนหลังจากการศึกษาวรรณกรรมและทฤษฎีอื่น ๆ ที่เกี่ยวข้องกับงานวิจัย ผู้วิจัยได้เก็บรวบรวมข้อมูลและศึกษาความเป็นไปได้ของวรรณกรรมที่ได้ทำการทบทวนว่ามีความเป็นไปได้ในการนำมาดำเนินการหรือใช้กับงานวิจัยหรือไม่ และได้มีการกำหนดขอบเขตของงานวิจัยให้มีความชัดเจนและเหมาะสมกับการดำเนินงาน เนื่องจากในกระบวนการวิจัยมีระยะเวลาที่จำกัด

ดังนั้นจึงต้องกำหนดขอบเขตให้มีความเหมาะสมชัดเจนเพื่อให้สามารถดำเนินงานวิจัยได้ทันตามระยะเวลาที่กำหนดไว้ และได้มีการค้นหาเครื่องมือในการพัฒนากรอบแนวคิด กรณีศึกษาที่จะนำมาใช้กับต้นแบบ และกำหนดขอบเขตความสามารถของเครื่องมือที่พัฒนาขึ้นจากกรอบแนวคิดของงานวิจัย รวมถึงวิเคราะห์ความเสี่ยงที่อาจส่งผลต่องานวิจัยได้

3.1.3 ออกแบบกรอบแนวคิดและออนโทโลยีที่จะนำมาใช้

กรอบแนวคิดที่ได้นำเสนอในงานวิจัยนี้ ผู้วิจัยได้ทำการพิจารณากระบวนการทำงานออกเป็น 4 ขั้นตอน ดังนี้ 1) การออกแบบและพัฒนาออนโทโลยี 2) การแปลงออนโทโลยีจาก OWL เป็น XML 3) การจัดการตัวแปรและต้นไม้การตัดสินใจ 4) การสร้างกรณีทดสอบ โดยกระบวนการออกแบบและพัฒนาออนโทโลยีจะช่วยให้ข้อกำหนดความต้องการในเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ที่เขียนด้วยภาษาธรรมชาติอยู่ในรูปแบบเชิงโครงสร้าง โดยใช้เครื่องมือ ROO ในการจัดการข้อกำหนดความต้องการด้วยภาษาธรรมชาติควบคุม (CNL) ให้มีความชัดเจนมากขึ้นเพื่อนำมาใช้เป็นพจนานุกรมข้อมูลในการสร้างกรณีทดสอบ โดยจะทำการแปลงไฟล์ .owl ให้เป็นโครงสร้าง XML เพื่อให้สอดคล้องกับไฟล์ยูสเคสซึ่งเป็นไฟล์ .xml ก่อนจะนำไปจัดการตัวแปรและค่าของตัวแปรรวมถึงการออกแบบเงื่อนไขการตัดสินใจในการสร้างกรณีทดสอบด้วย เมื่อจัดการตัวแปรและต้นไม้การตัดสินใจเรียบร้อยแล้วจะเข้าสู่กระบวนการสร้างกรณีทดสอบโดยผู้วิจัยได้พิจารณานำเทคนิคการทดสอบแบบกล่องดำจำนวน 2 เทคนิคมารวมเข้าด้วยกัน คือ เทคนิคการรวมกันของชั้นสมมูลและต้นไม้การจำแนกมาใช้สำหรับการสร้างกรณีทดสอบให้มีความถูกต้องและครอบคลุมตามเงื่อนไขของการทดสอบ ส่งผลให้ระบบที่พัฒนามีประสิทธิภาพและตรงตามความต้องการของผู้ใช้

3.1.4 พัฒนาต้นแบบตามกรอบแนวคิด

หลังจากได้มีการออกแบบกรอบแนวคิดของงานวิจัยเรียบร้อยแล้วผู้วิจัยได้ดำเนินการพัฒนาต้นแบบซึ่งสอดคล้องกับกรอบแนวคิดของงานวิจัย ซึ่งเป็นต้นแบบสำหรับการสร้างกรณีทดสอบแบบอัตโนมัติด้วยออนโทโลยีความต้องการซอฟต์แวร์ โดยการทำงานเริ่มต้นของต้นแบบจะทำการนำเข้าไฟล์จำนวน 2 ไฟล์ เพื่อนำมาใช้ในการประมวลผล ได้แก่ 1) ไฟล์ .xml ของยูสเคส และ 2) ไฟล์ .owl ของออนโทโลยี แล้วหลังจากนั้นจะทำการแปลงไฟล์ .owl ให้เป็นโครงสร้าง XML เพื่อให้สอดคล้องกับยูสเคสและเพื่อนำไปเป็นพจนานุกรมข้อมูลในการสร้างกรณีทดสอบ รวมถึงการออกแบบต้นไม้การตัดสินใจในกรณีที่ระบบมีเงื่อนไขหรือข้อจำกัดบางอย่าง และผู้วิจัยได้มีการพิจารณาเทคนิคการรวมกันของชั้นสมมูลและต้นไม้การจำแนกเพื่อใช้ในการสร้างกรณีทดสอบให้มีความครอบคลุมในทุก ๆ กรณีของการทดสอบซอฟต์แวร์

3.1.5 ทดสอบและประเมินความถูกต้องของต้นแบบ

ในขั้นตอนการทดสอบและประเมินความถูกต้องของต้นแบบจะดำเนินการโดยใช้กรณีศึกษาของระบบจำนวน 2 กรณีศึกษา ได้แก่ 1) กรณีศึกษาของระบบจัดการห้องสมุด และ 2) กรณีศึกษาของระบบวิเคราะห์คำนวณค่าไต ในการทดสอบต้นแบบจะทำการทดสอบฟังก์ชันการทำงานต่าง ๆ ของต้นแบบหลัก ๆ 3 ฟังก์ชันด้วยกัน ได้แก่ 1) ฟังก์ชันการแปลงออนโทโลยีจาก OWL เป็น XML 2) ฟังก์ชันการจัดการตัวแปรและต้นไม้การตัดสินใจ 3) ฟังก์ชันการสร้างกรณีทดสอบ เพื่อตรวจสอบความถูกต้องในการทำงานของต้นแบบว่าสามารถสร้างกรณีทดสอบได้สอดคล้องกับกรณีศึกษาที่นำมาใช้หรือไม่ สำหรับในส่วนของการการประเมินผลต้นแบบผู้วิจัยจะทำการประเมินจากการสร้างกรณีทดสอบด้วยวิธีการมือกับที่ได้จากต้นแบบว่ามีความสอดคล้องและครอบคลุมตามเงื่อนไขของการทดสอบหรือไม่ รวมถึงได้ทำการประเมินผลต้นแบบโดยผู้เชี่ยวชาญที่มีความรู้ทางด้านวิศวกรรมซอฟต์แวร์และการทดสอบซอฟต์แวร์ในการตรวจสอบการทำงานของต้นแบบว่าสามารถดำเนินการได้ตรงตามกรอบแนวคิดที่ได้นำเสนอไว้ในงานวิจัยนี้หรือไม่ รวมถึงความเป็นไปได้ในการนำต้นแบบที่พัฒนาไปใช้งานสำหรับการสร้างกรณีทดสอบในการทดสอบซอฟต์แวร์

3.1.6 สรุปผลการวิจัยและข้อเสนอแนะ

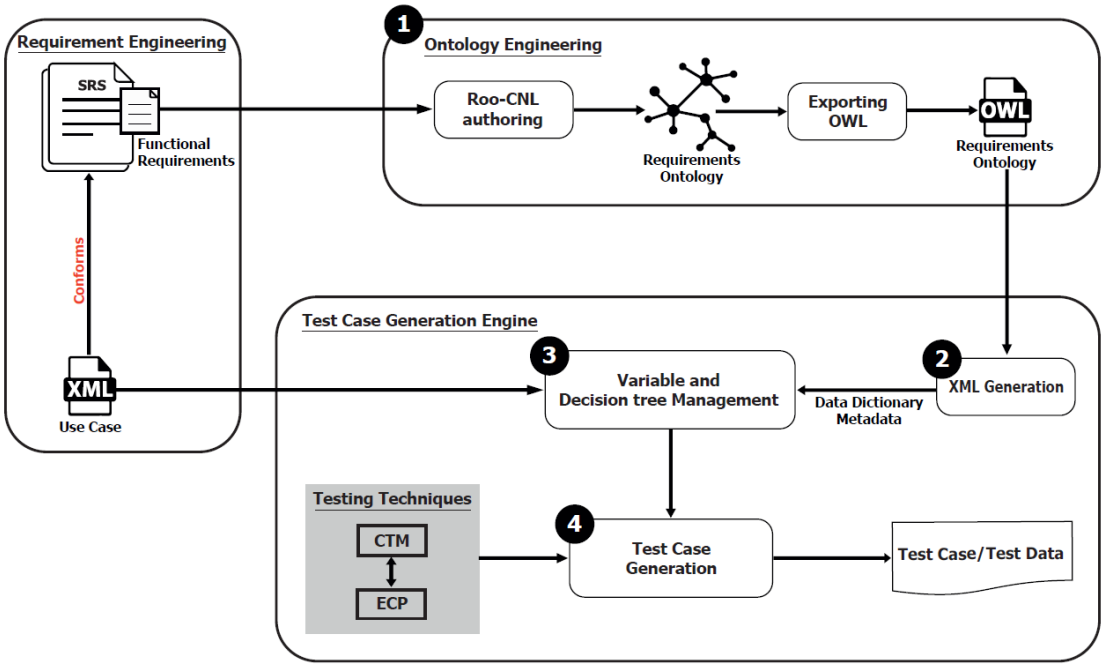
สำหรับการสรุปผลการวิจัยและข้อเสนอแนะต่าง ๆ ของงานวิจัยนี้ หลังจากได้ดำเนินการพัฒนาต้นแบบและประเมินผลการทำงานของต้นแบบเรียบร้อยแล้ว ผู้วิจัยได้ทำการสรุปผลเครื่องมือที่ได้พัฒนาตามกรอบแนวคิดซึ่งผ่านการตรวจสอบความถูกต้องและความน่าเชื่อถือจากผู้เชี่ยวชาญ รวมถึงข้อเสนอแนะต่าง ๆ ซึ่งอาจจะเป็นประโยชน์ต่อการนำไปพัฒนาต่อยอดในอนาคต เมื่อผลสรุปเป็นไปตามความคาดหวังและสอดคล้องกับกรอบแนวคิดของงานวิจัย หลังจากนั้นผู้วิจัยจะทำการนำข้อมูลทั้งหมดจากการดำเนินงานวิจัยมาจัดทำรายงานสรุปผล

3.1.7 จัดทำรายงานสรุปผล

การดำเนินงานวิจัยนี้หลังจากกระบวนการเริ่มต้นตั้งแต่การศึกษาวรรณกรรมและทฤษฎีที่เกี่ยวข้อง การรวบรวมและกำหนดขอบเขตของงานวิจัย การออกแบบกรอบแนวคิด การพัฒนาต้นแบบตามกรอบแนวคิดที่นำเสนอ การประเมินผลต้นแบบ และสรุปผลการวิจัยเป็นที่เรียบร้อยแล้ว ในขั้นตอนสุดท้ายของการดำเนินงานวิจัยผู้วิจัยได้ทำการจัดทำรายงานสรุปผลการดำเนินงานวิจัยในรูปแบบของวิทยานิพนธ์ รวมถึงบทความวิชาการ และวารสารวิชาการ เพื่อเป็นหลักฐานเชิงประจักษ์เกี่ยวกับการดำเนินงานวิจัยในครั้งนี้

3.2 กรอบแนวคิดของงานวิจัย

ในส่วนนี้จะกล่าวถึงภาพรวมการทำงานของกรอบแนวคิดงานวิจัยที่ได้ทำการออกแบบ รวมถึงรายละเอียดของกระบวนการและขั้นตอนต่าง ๆ ที่ผู้วิจัยได้นำเสนอว่ามีการดำเนินการอย่างไร โดยได้นำรายละเอียดของระบบจัดการห้องสมุด (library management system) บางส่วนมาใช้ในการอธิบายกรอบแนวคิดนี้ แสดงภาพรวมการทำงานของกรอบแนวคิด ดังรูปที่ 3.2



รูปที่ 3.2 กรอบแนวคิดของงานวิจัย

จากรูปที่ 3.2 เป็นภาพรวมการทำงานทั้งหมดของกรอบแนวคิดงานวิจัย โดยมีกระบวนการดำเนินงาน 3 ส่วนด้วยกัน ได้แก่ กระบวนการวิศวกรรมความต้องการ (requirement engineering) กระบวนการวิศวกรรมออนโทโลยี (ontology engineering) และกระบวนการสร้างกรณีทดสอบ (test case generation engine) ซึ่งจะประกอบไปด้วยขั้นตอนการทำงาน 4 ขั้นตอน ได้แก่ 1) การออกแบบและพัฒนาออนโทโลยี 2) การแปลงออนโทโลยีจาก OWL เป็น XML 3) การจัดการตัวแปรและต้นไม้การตัดสินใจ 4) การสร้างกรณีทดสอบ โดยมีรายละเอียดในแต่ละกระบวนการ ดังนี้

3.2.1 กระบวนการวิศวกรรมความต้องการ (Requirement Engineering)

กระบวนการวิศวกรรมความต้องการจะเป็นการรวบรวมความต้องการจากผู้ใช้ว่าต้องการให้ระบบทำงานอย่างไร โดยความต้องการจะถูกจัดเก็บและเขียนไว้ในเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ในงานวิจัยนี้ผู้วิจัยจะดำเนินการโดยนำส่วนของความต้องการฟังก์ชันการทำงาน (functional requirement) ในเอกสารมาเป็นข้อมูลนำเข้าของต้นแบบสำหรับการสร้างกรณีทดสอบรวมถึงความต้องการที่เป็นเงื่อนไขการตัดสินใจหรือข้อจำกัดด้วยเช่นกัน โดยความต้องการ (requirement) ที่นำมาใช้ในงานวิจัยนี้จะพิจารณาข้อกำหนดความต้องการที่เป็นภาษาอังกฤษ แสดงตัวอย่างความต้องการของระบบจัดการห้องสมุด ดังตารางที่ 3.1

ตารางที่ 3.1 ตัวอย่างความต้องการระบบจัดการห้องสมุด

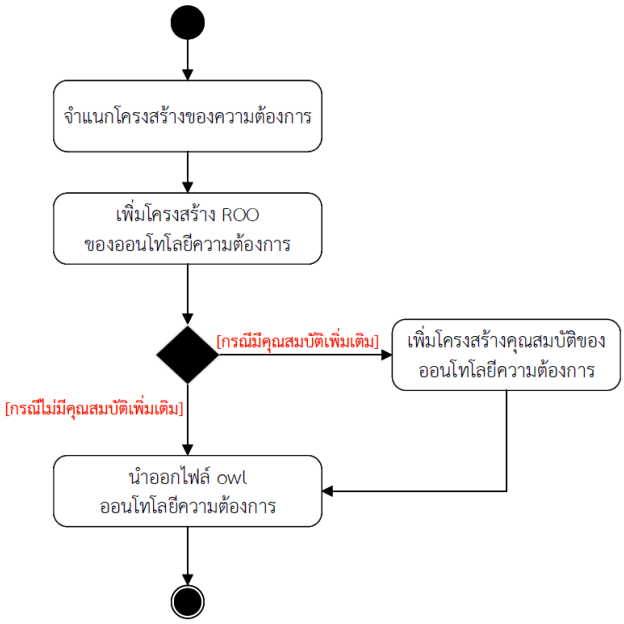
Req. ID	Requirements
LIB-FUN-01	Members can borrow item including books, CDs or DVDs.
LIB-FUN-02	Members are classified into admin staff, graduate student, lecturer and undergraduate.
LIB-CON-01	Books, CDs, DVDs must be disjointed.
LIB-CON-02	Admin, staff, graduate student, lecturer and undergrade must be disjointed.
LIB-CON-03	Maximum borrowing items and borrowing periods: 5 books per 7 days for admin staff and undergraduate students, 10 books per 14 days for graduate students, and 15 books per 30 days for lecturers.
LIB-CON-04	Maximum borrowing items and borrowing periods: 3 discs per 7 days for all members.
LIB-FUN-03	Members can return the borrowed item.
LIB-FUN-04	If there is late return, members must pay fine.

จากตารางที่ 3.1 เป็นตัวอย่างความต้องการของระบบจัดการห้องสมุด ซึ่งจะประกอบไปด้วยคอลัมน์แรก คือ รหัสความต้องการ (Req. ID) และความต้องการ (Requirements) ตามลำดับ ตัวอย่างเช่น ความต้องการรหัส LIB-FUN-01 เป็นความต้องการที่สมาชิกสามารถยืมทรัพยากรของห้องสมุดได้ ประกอบด้วย หนังสือ ซีดี หรือดิวีดี เป็นต้น และผู้วิจัยยังได้มีการพิจารณาในส่วนข้อจำกัดของความต้องการ (constraint) เพื่อให้การสร้างกรณีทดสอบมีความครอบคลุมมากยิ่งขึ้น ตัวอย่างเช่น ความต้องการรหัส LIB-CON-04 เป็นข้อจำกัดความต้องการของสมาชิกแต่ละประเภทสำหรับการยืมทรัพยากรของห้องสมุดในแต่ละครั้งว่าสามารถยืมได้จำนวนสูงสุดเท่าไหร่และได้นานที่สุดกี่วัน เป็นต้น โดยความต้องการเหล่านี้จะถูกนำไปใช้ในกระบวนการออกแบบและพัฒนาออนโทโลยีเพื่อใช้เป็นพจนานุกรมข้อมูล รวมถึงได้พิจารณายูสเคส (use case) ซึ่งจะสอดคล้องกับความต้องการในเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ที่เป็นบทภาพ

เหตุการณ์ของแต่ละยูสเคส โดยข้อมูลเหล่านี้ทั้งในส่วนของออนโทโลยีความต้องการซอฟต์แวร์และตัวยูสเคสจะนำไปใช้เป็นข้อมูลนำเข้าสำหรับกระบวนการในการสร้างกรณีทดสอบต่อไป

3.2.2 กระบวนการวิศวกรรมออนโทโลยี (Ontology Engineering)

กระบวนการวิศวกรรมออนโทโลยีเป็นขั้นตอนเริ่มต้นการทำงานของกรอบแนวคิด โดยมีหนึ่งขั้นตอนที่สำคัญของการสร้างกรณีทดสอบ คือ ขั้นตอนการออกแบบและพัฒนาออนโทโลยี เป็นขั้นตอนในการนำความต้องการจากกระบวนการก่อนหน้ามาทำการจำแนกโครงสร้างและจัดการโครงสร้างให้มีความชัดเจนก่อนนำไปสร้างกรณีทดสอบ สามารถแสดงกิจกรรมการดำเนินงานดังรูปที่ 3.3



รูปที่ 3.3 แผนภาพกิจกรรมกระบวนการวิศวกรรมออนโทโลยี

จากรูปที่ 3.3 กิจกรรมของกระบวนการนี้จะเริ่มจากการนำความต้องการในกระบวนการก่อนหน้ามาทำการจำแนกโครงสร้าง เนื่องจากความต้องการอยู่ในรูปแบบของภาษาธรรมชาติที่ไม่มีโครงสร้าง ดังนั้นในงานวิจัยนี้ผู้วิจัยได้นำเสนอเครื่องมือ ROO ซึ่งเป็นเครื่องมือในการออกแบบและพัฒนาออนโทโลยีโดยใช้ Protégé 4 เข้ามาช่วยในการจัดการข้อกำหนดความต้องการที่เขียนด้วยภาษาธรรมชาติให้อยู่ในรูปแบบเชิงโครงสร้างมากขึ้นก่อนนำไปสร้างกรณีทดสอบ โดยเครื่องมือนี้จะใช้ภาษาธรรมชาติควบคุม (CNL) ช่วยให้ผู้เชี่ยวชาญสามารถออกแบบและพัฒนาออนโทโลยีได้ง่ายและไม่ซับซ้อนด้วยภาษาธรรมชาติที่มีไวยากรณ์กำหนดไว้อย่างชัดเจนซึ่งเป็นรูปแบบเฉพาะของเครื่องมือที่สามารถอธิบายโดเมนของสิ่งที่สนใจได้อย่างอิสระในรูปแบบประโยคภาษาอังกฤษตามธรรมชาติ โดยมีการกำหนดโครงสร้างหลัก ๆ ของแนวคิด (concept) และ

ความสัมพันธ์ (relation) รวมถึงคุณสมบัติต่าง ๆ ของออนโทโลยีความต้องการซอฟต์แวร์ จากความต้องการของระบบจัดการห้องสมุดในกระบวนการก่อนหน้ามาทำการจำแนกตามรูปแบบโครงสร้าง ROO ดังตารางที่ 3.2

ตารางที่ 3.2 โครงสร้าง ROO ของความต้องการระบบจัดการห้องสมุด

Req.ID	**Roo-CNL Structure**	**OWL2 Functional Syntax**
LIB-FUN-01	*Member* is a concept *Item* is a concept	Declaration(Class*(:Member))* Declaration(Class*(:Item))*
	Every *Book* is a kind of *Item* Every *CD* is a kind of *Item* Every *DVD* is a kind of *Item*	SubClassOf*(:Book :Item)* SubClassOf*(:CD :Item)* SubClassOf*(:DVD :Item)*
	hasBorrow is a relationship Every *Member hasBorrow Item*	Declaration(ObjectProperty*(:hasBorrow))* ObjectPropertyDomain*(:hasBorrow :Member)* ObjectPropertyRange(*:hasBorrow :Item)*
LIB-FUN-02	Every *AdminStaff* is a kind of *Member* Every *Grad i*s a kind of *Member* Every *Lecturer* is a kind of *Member* Every *UnderGrad* is a kind of *Member*	SubClassOf*(:AdminStaff :Member)* SubClassOf*(:Grad :Member)* SubClassOf*(:Lecturer :Member)* SubClassOf*(:UnderGrad :Member)*
LIB-CON-01	*DisjointClasses()*	DisjointClasses*(:Book :CD)* DisjointClasses*(:Book :DVD)* DisjointClasses*(:CD :DVD)*
LIB-CON-02	*DisjointClasses()*	DisjointClasses*(:AdminStaff :Grad)* DisjointClasses*(:AdminStaff :Lecturer)* DisjointClasses*(:AdminStaff :UnderGrad)* DisjointClasses*(:Grad :Lecturer)* DisjointClasses*(:Grad :UnderGrad)* DisjointClasses*(:Lecturer :UnderGrad)*
LIB-CON-03	*borrowDaysR1* is a *maxDaysBorrow*	Declaration(NameIndividual*(:borrowDaysR1))* ClassAssertion*(:AdminStaff :borrowDaysR1*) DataPropertyAssertion*(:maxDaysBorrow :borrowDaysR1 "7"^^xsd:integer)*
LIB-CON-04	And, configure data property assertion directly through GUI in the tool.	
LIB-FUN-03	*hasReturn* is a relationship Every *Member hasReturn Item*	Declaration(ObjectProperty*(:hasReturn))* ObjectPropertyDomain*(:hasReturn :Member)* ObjectPropertyRange*(:hasReturn :Item)*
LIB-FUN-04	*DataProperty()* *DataPropertyDomain()* *ObjectSomeValuesFrom()* *DataPropertyRange()*	Declaration(DataProperty*(:fine))* DataPropertyDomain*(:fine* ObjectAllValuesFrom*(:hasReturn :Member))* DataPropertyRange*(:fine xsd:integer)*

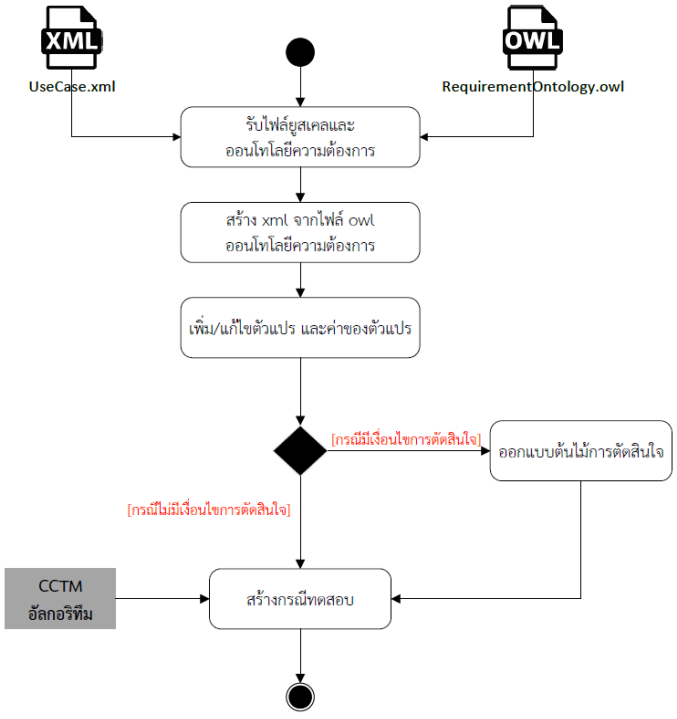
จากตารางที่ 3.2 โครงสร้างของ ROO ข้างต้น มีรายละเอียดแต่ละคอลัมน์ ดังนี้ คอลัมน์แรก (Req.ID) เป็นรหัสของความต้องการ คอลัมน์ (Roo-CNL Structure) เป็นคอลัมน์โครงสร้างของ ROO คอลัมน์สุดท้าย (OWL2 Functional Syntax) เป็นคอลัมน์ของไวยกรณ์ OWL หลังจากทำการส่งออกมาจากเครื่องมือ ตัวอย่างเช่น LIB-FUN-01: Members can borrow item

including books, CDs or DVDs. สามารถออกแบบเป็นคลาส (class) ด้วยโครงสร้างประโยค *<Concept>* is a concept ได้ 2 คลาส คือ *Member* กับ *Item* และออกแบบคลาสย่อย (subclass) ด้วยโครงสร้างประโยค Every *<Concept>* is a kind of *<Concept>* ได้ 3 คลาสย่อย คือ *Book, CD* และ *DVD*

แต่มีข้อจำกัดของคุณสมบัติบางอย่าง เช่น LIB-CON-03 ที่ต้องกำหนดด้วย GUI ภายในเครื่องมือ Protégé เพื่อให้ออนโทโลยีมีความละเอียดและครอบคลุมมากขึ้น เมื่อทำการจำแนกและระบุโครงสร้างลงไปในเครื่องมือ ROO และ Protégé เรียบร้อยแล้วจะทำการส่งออกไฟล์ออนโทโลยีในรูปแบบไฟล์ .owl เพื่อนำไปเป็นพจนานุกรมข้อมูลในกระบวนการสร้างกรณีทดสอบ

3.2.3 กระบวนการสร้างกรณีทดสอบ (Test Case Generation Engine)

ในกระบวนการสร้างกรณีทดสอบมีขั้นตอนการดำเนินกิจกรรมการทำงานของต้นแบบงานวิจัยตามแผนภาพกิจกรรม ดังรูปที่ 3.4



รูปที่ 3.4 แผนภาพกิจกรรมของกระบวนการสร้างกรณีทดสอบ

จากรูปที่ 3.4 ในการสร้างกรณีทดสอบของต้นแบบขั้นตอนแรกจะต้องนำเข้าไฟล์ .xml ซึ่งเป็นไฟล์ยูสเคส และไฟล์ .owl ของออนโทโลยีความต้องการซอฟต์แวร์ โดยไฟล์ยูสเคสที่นำเข้ามาจะแสดงรายละเอียดต่าง ๆ รวมถึงฟังก์ชันที่เกี่ยวข้องในแต่ละยูสเคส ส่วนไฟล์ .owl จะเป็นตัวพจนานุกรมข้อมูลที่จะนำมาใช้ในการสร้างกรณีทดสอบ

3.2.3.1 ขั้นตอนการแปลงออนโทโลยีจาก OWL เป็น XML

ในกระบวนการนี้จะใช้ไฟล์ .owl ที่ได้จากการนำเข้ามาแปลงให้เป็นโครงสร้าง XML เนื่องจากเป็นภาษามาตรฐานที่นิยมใช้ในการแลกเปลี่ยนข้อมูลระหว่างโปรแกรมหรือระบบที่ต้องทำการติดต่อสื่อสารระหว่างกัน และเพื่อให้สอดคล้องกับไฟล์ .xml ของยูสเคสที่ได้มีการนำเข้ามาใช้ในการสร้างกรณีทดสอบของต้นแบบ ดังนั้นผู้วิจัยจึงได้มีการศึกษาค้นคว้าแนวทางในการแปลงออนโทโลยีจาก OWL เป็น XML เพื่อให้ง่ายต่อการแลกเปลี่ยนข้อมูลระหว่างโปรแกรม (Hajjamy, Alaoui, and Bahaj, 2017)

โดยจากการศึกษาค้นคว้าผู้วิจัยได้มีการออกแบบกฎการแปลงที่เกี่ยวข้องกับงานวิจัยไว้ 13 กฎ และได้มีการแสดงกฎการแปลงทั้งหมด รวมถึงตัวอย่างการแปลงของกฎแต่ละกฎในภาคผนวก ก. (กฎการแปลงออนโทโลยีจาก OWL เป็น XML ในส่วน ก1 และ ก2 ตามลำดับ) แสดงตัวอย่างรายละเอียดของการแปลง ดังตารางที่ 3.3

ตารางที่ 3.3 ตัวอย่างกฎการแปลง OWL เป็น XML

Rules	**OWL2 Functional**	**XML Schema**
The structure of Classes and Relations		
1. Class Declaration	Class(*:cname*)	Local: <xs:element name="*cname*"> <xs:complexType> </xs:complexType> </xs:element> Global: <xs:element name="*cname*"/>
2. Subclass Declaration	SubClassOf(*:scname :cname*)	<xs:element name="*scname*" type="*cname*"/>
3. Object Property Declaration	ObjectProperty(*:hasClass*) ObjectPropertyDomain(*:hasClass :cdomain*) ObjectPropertyRange:(*:hasClass :crange*)	<xs:element name="*cdomain*"> <xs:complexType> <xs:sequence> <xs:element name="*hasClass*" ref="*crange*"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="*crange*"> <xs:complexType> </xs:complexType> </xs:element>
4. Data Property Declaration	DataProperty(*:dprop*) DataPropertyDomain(*:dprop :cname*) DataPropertyRange(*:dprop* xsd:*dtype*)	<xs:element name="*cname*"> <xs:complexType> <xs:sequence> <xs:element name="*dprop*" type="xs:*dtype*"/> </xs:sequence> </xs:complexType> </xs:element>

ตารางที่ 3.3 ตัวอย่างกฎการแปลง OWL เป็น XML (ต่อ)

Rules	**OWL2 Functional**	**XML Schema**
The structure of Classes and Relations		
5. Existential	EquivalentClasses(*:cdomain* ObjectSomeValuesFrom(*:hasClass :crange*))	<xs:element name="*cdomain*"> <xs:complexType> <xs:sequence> <xs:element name="*hasClass*" ref="*crange*"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="*crange*"> <xs:complexType> <xs:choice> </xs:choice> </xs:complexType> </xs:element>
…	…	…
Data Property Restrictions		
…	…	…
13. Maximum cardinality	EquivalentClasses(*:cname* DataMaxCardinality(*max :dprop* xsd:*dtype*)	<xs:element name="*cname*"> <xs:complexType> <xs:sequence> <xs:element name="*dprop*" type="xs:*dtype*" maxOccurs="*max*"/> </xs:sequence> </xs:complexType> </xs:element>

จากตารางที่ 3.3 เป็นตัวอย่างกฎการแปลงบางส่วน ประกอบด้วย คอลัมน์แรก (Rules) เป็นกฎของการแปลง คอลัมน์ที่สอง (OWL2 Functional) เป็นไวยกรณ์ OWL และคอลัมน์สุดท้าย (XML Schema) เป็นไวยกรณ์ XML โดยแบ่งเป็น 3 ส่วนหลัก ๆ ของกฎการแปลง ได้แก่ 1) กฎการแปลงคลาสและความสัมพันธ์ (The structure of Classes and Relations) 2) กฎการแปลงข้อจำกัดความสัมพันธ์ (Object Property Restrictions) และ 3) กฎการแปลงข้อจำกัดคุณสมบัติ (Data Property Restrictions) สามารถอธิบายตัวอย่างกฎการแปลงในแต่ละส่วน ดังนี้

กฎข้อที่ 1: Class Declaration

เป็นกฎการประกาศคลาสโดยการกำหนดให้ *cname* เป็นชื่อของคลาสในออนโทโลยี และสามารถแปลงคลาสให้เป็นอิลีเมนต์ของ XML ได้ 2 แบบ คือ อิลีเมนต์แบบ Local และ Global ดังรูปที่ 3.5

```
Local:
<xs:element name="Member">
   <xs:complexType> </xs:complexType>
</xs:element>

Global:
<xs:element name="Member"/>
```

รูปที่ 3.5 ตัวอย่างการแปลงคลาส

กฎข้อที่ 2: Subclass Declaration

เป็นการแปลงคลาสย่อยของคลาส *cname* โดยที่ *scname* จะเป็นชื่อของคลาสย่อย และจะแปลงเป็นอิลีเมนต์ของ XML โดยมี type เป็น *cname* ซึ่งจะอ้างถึงคลาสแม่ของคลาสย่อยนั้น ๆ ดังรูปที่ 3.6

```
<xs:element name="Book" type="Item"/>
```

รูปที่ 3.6 ตัวอย่างการแปลงคลาสย่อย

กฎข้อที่ 3: Object Property Declaration

เป็นการกำหนดความสัมพันธ์ระหว่างคลาส โดยให้ *hasClass* เป็นชื่อของความสัมพันธ์และมี *cdomain* ซึ่งเป็นคลาสโดเมนที่มีความสัมพันธ์กับคลาส *crange* โดยที่สามารถแปลงให้เป็นอิลีเมนต์ภายใน <xs:complexType>...</xs:complexType> และมีการอ้างอิง (ref) ไปยังคลาสความสัมพันธ์ภายในอิลีเมนต์ ดังรูปที่ 3.7

```
<xs:element name="Member">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="hasBorrow" ref="Item"/>
      </xs:sequence>
   </xs:complexType>
</xs:element>
<xs:element name="Item">
   <xs:complexType> </xs:complexType>
</xs:element>
```

รูปที่ 3.7 ตัวอย่างการแปลงความสัมพันธ์ของคลาส

กฎข้อที่ 4: Data Property Declaration

เป็นการกำหนดคุณสมบัติและประเภทของคุณสมบัติให้กับคลาส โดยมี *dprop* เป็นชื่อคุณสมบัติของคลาส และ *xsd:dtype* เป็นประเภทของคุณสมบัติ และสามารถแปลงเป็นอิลีเมนต์ของคุณสมบัติซึ่งจะอยู่ภายใต้อิลีเมนต์ของคลาส *cname* นั้น ๆ ดังรูปที่ 3.8

```
<xs:element name="Member">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="address" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

รูปที่ 3.8 ตัวอย่างการแปลงคุณสมบัติของคลาส

กฎของที่ 5: Object Property Restrictions (Existential)

เป็นการกำหนดองค์ประกอบของคลาสความสัมพันธ์ โดยที่ existential จะถูกแปลงให้เป็นอิลีเมนต์องค์ประกอบที่อยู่ภายใต้ <xs:choice>...</xs:choice> ซึ่งจะมีคลาสความสัมพันธ์ที่เกี่ยวข้องและองค์ประกอบของคลาส ดังรูปที่ 3.9

```
<xs:element name="Member">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="hasBorrow" ref="Item"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Item">
  <xs:complexType>
    <xs:choice>
      <xs:element name="Book" type="xs:string"/>
      <xs:element name="CD" type="xs:string"/>
      <xs:element name="DVD" type="xs:string"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

รูปที่ 3.9 ตัวอย่างการแปลงองค์ประกอบของคลาสความสัมพันธ์

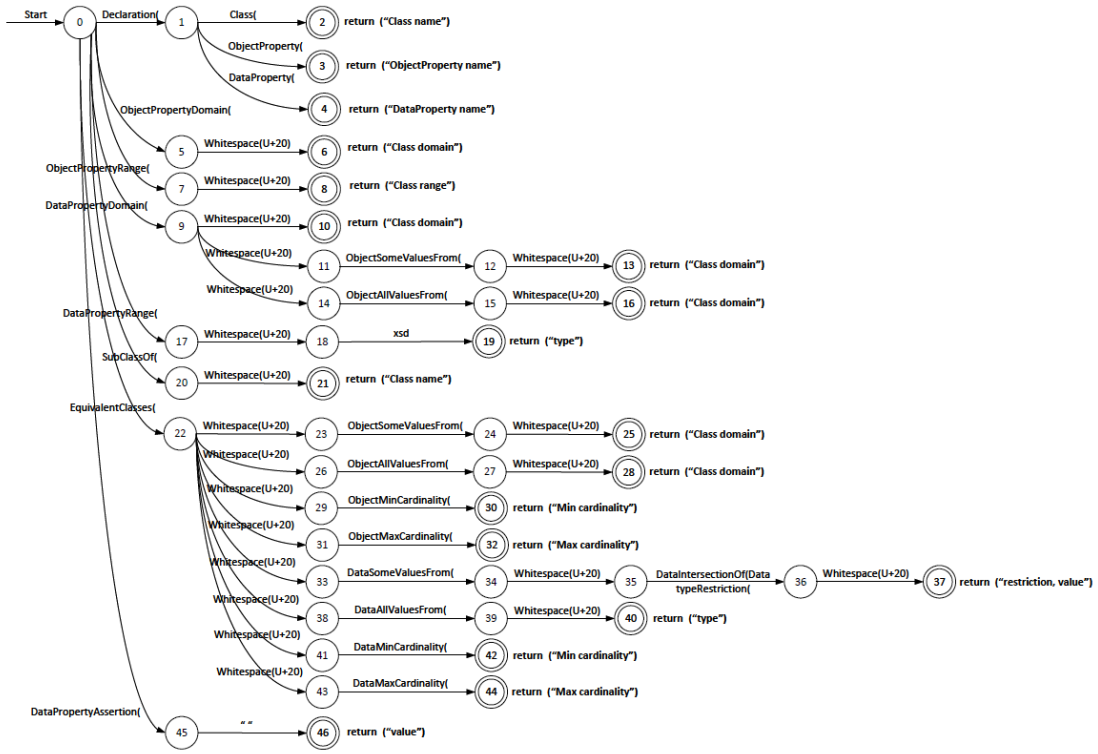
กฎของที่ 13: Data Property Restrictions (Maximum cardinality)

เป็นการแปลงตัวบ่งปริมาณของคุณสมบัติ โดยที่ maximum cardinality จะถูกแปลงให้เป็น macOccurs ในคุณสมบัติของ XML ซึ่งเป็นจำนวนสูงสุดที่กำหนดให้กับคุณสมบัตินั้น ๆ ดังรูปที่ 3.10

```
<xs:element name="Item">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CD" type="xs:string" maxOccurs="3"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

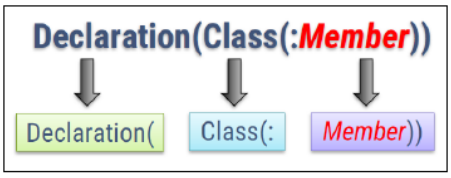
รูปที่ 3.10 ตัวอย่างการแปลงตัวบ่งปริมาณสูงสุดของคุณสมบัติ

ในการแปลงออนโทโลยีจาก OWL เป็น XML ผู้วิจัยได้ทำการวิเคราะห์ไฟล์ .owl ที่ได้จากการออกแบบและพัฒนาออนโทโลยีของข้อกำหนดความต้องการซอฟต์แวร์ เพื่อนำมาตัดคำสำหรับการแปลงของกฎแต่ละกฎที่เกี่ยวข้อง แสดงแผนภาพที่ได้มีการวิเคราะห์คำของไฟล์ .owl ดังรูปที่ 3.11



รูปที่ 3.11 แผนภาพวิเคราะห์คำสำหรับการแปลงออนโทโลยีจาก OWL เป็น XML

จากรูปที่ 3.11 เป็นการวิเคราะห์เพื่อตัดคำในการแปลงออนโทโลยีจาก OWL เป็น XML ตัวอย่างเช่น ประโยค Declaration(Class(:*Member*)) ในไฟล์ .owl ของระบบจัดการห้องสมุด สามารถอธิบายการวิเคราะห์คำในการแปลง ดังรูปที่ 3.12



รูปที่ 3.12 การวิเคราะห์คำสำหรับกฎการแปลง

จากรูปที่ 3.12 เป็นตัวอย่างการวิเคราะห์คำสำหรับกฎการแปลง สามารถแปลงประโยคนี้จากรูปแบบไวยกรณ์ OWL ให้เป็น XML ซึ่งจะสอดคล้องกับกฎการแปลงข้อที่ 1 (Class Declaration) แสดงไวยกรณ์ของ XML ที่ได้จากกฎการแปลง ดังรูปที่ 3.13

```
<xs:element name="Member">
   <xs:complexType> </xs:complexType>
</xs:element>
```

รูปที่ 3.13 ไวยกรณ์ XML ของกฎการแปลงคลาส

3.2.3.2 ขั้นตอนการจัดการตัวแปรและต้นไม้การตัดสินใจ

ในกระบวนการจัดการตัวแปรและต้นไม้การตัดสินใจจะพิจาณาจากไฟล์นำเข้า 2 ไฟล์ ได้แก่ ไฟล์ยูสเคส และไฟล์ออนโทโลยีความต้องการซอฟต์แวร์หลังจากได้ทำการแปลงให้เป็นโครงสร้าง XML ของพจนานุกรมข้อมูล (data dictionary) ซึ่งจะเป็นข้อมูลตัวแปรและค่าของตัวแปรที่เกี่ยวข้องกับระบบ โดยไฟล์ยูสเคสจะถูกออกแบบให้มีความสอดคล้องกับความต้องการที่ระบุในเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ตามแนวทางการพัฒนาของยูเอ็มแอลเวอร์ชัน 2.0 (Pilone and Pitman, 2005) ในการอธิบายภาพรวมระดับบนสุดของบทภาพเหตุการณ์การทำงาน ขั้นตอนต่าง ๆ รวมถึงผู้กระทำที่เกี่ยวข้องและฟังก์ชันการทำงานของระบบสำหรับใช้ในการสร้างกรณีทดสอบ สามารถแสดงตัวอย่างรายละเอียดไฟล์ .xml ของยูสเคสระบบจัดการห้องสมุด ดังรูปที่ 3.14

Use Case	Description
UC001-Borrow Item	Use Case ID: UC001 Use Case Name: Borrow Item Pre-Condition: Checking members of the library Post-Condition: Borrow items successfully Priority: Hight Flow of Event: 1.The system shows GUI for a list of items to borrow 2.Member select items to borrow 3.Member confirm borrow items [A1] 4.The system records the borrowed items 5.The system displays a list of borrowed items Alternative Flow: [A1] If click "Cancel" button, the system will not records [E1] Exception Flow: [E1] The system shows the warning "Confirm cancellation?"

รูปที่ 3.14 ตัวอย่างรายละเอียดไฟล์ยูสเคสระบบจัดการห้องสมุด

ในการสร้างกรณีทดสอบของต้นแบบจะทำการเลือกตัวแปรของฟังก์ชันที่ต้องการสร้างกรณีทดสอบซึ่งสอดคล้องกับฟังก์ชันที่เกี่ยวข้องกับระบบในไฟล์ .xml ของยูสเคสที่ถูกนำเข้ามา และสามารถเพิ่มค่าให้กับตัวแปรซึ่งได้จากพจนานุกรมข้อมูลที่ดึงมาจากการแปลงไฟล์ .owl ของออนโทโลยีความต้องการซอฟต์แวร์ ในการจัดการตัวแปรจะครอบคลุมถึงการระบุเงื่อนไขของระบบ โดยการออกแบบต้นไม้การตัดสินใจในกรณีที่ระบบมีข้อจำกัดหรือเงื่อนไขบางอย่างเพื่อให้ครอบคลุมการทดสอบ แต่ถ้าหากระบบไม่มีเงื่อนไขหรือข้อจำกัดสามารถนำตัวแปรที่จัดการเรียบร้อยแล้วไปสร้างกรณีทดสอบด้วยเทคนิคการจำแนกตัวแปรโดยใช้ต้นไม้การจำแนกและการแบ่งชั้นสมมูลของค่าข้อมูลที่เกี่ยวข้อง แสดงตัวอย่างของตัวแปรจากระบบจัดการห้องสมุดที่นำมาใช้ในการอธิบายดังตารางที่ 3.4

ตารางที่ 3.4 ตัวอย่างตัวแปรระบบจัดการห้องสมุด

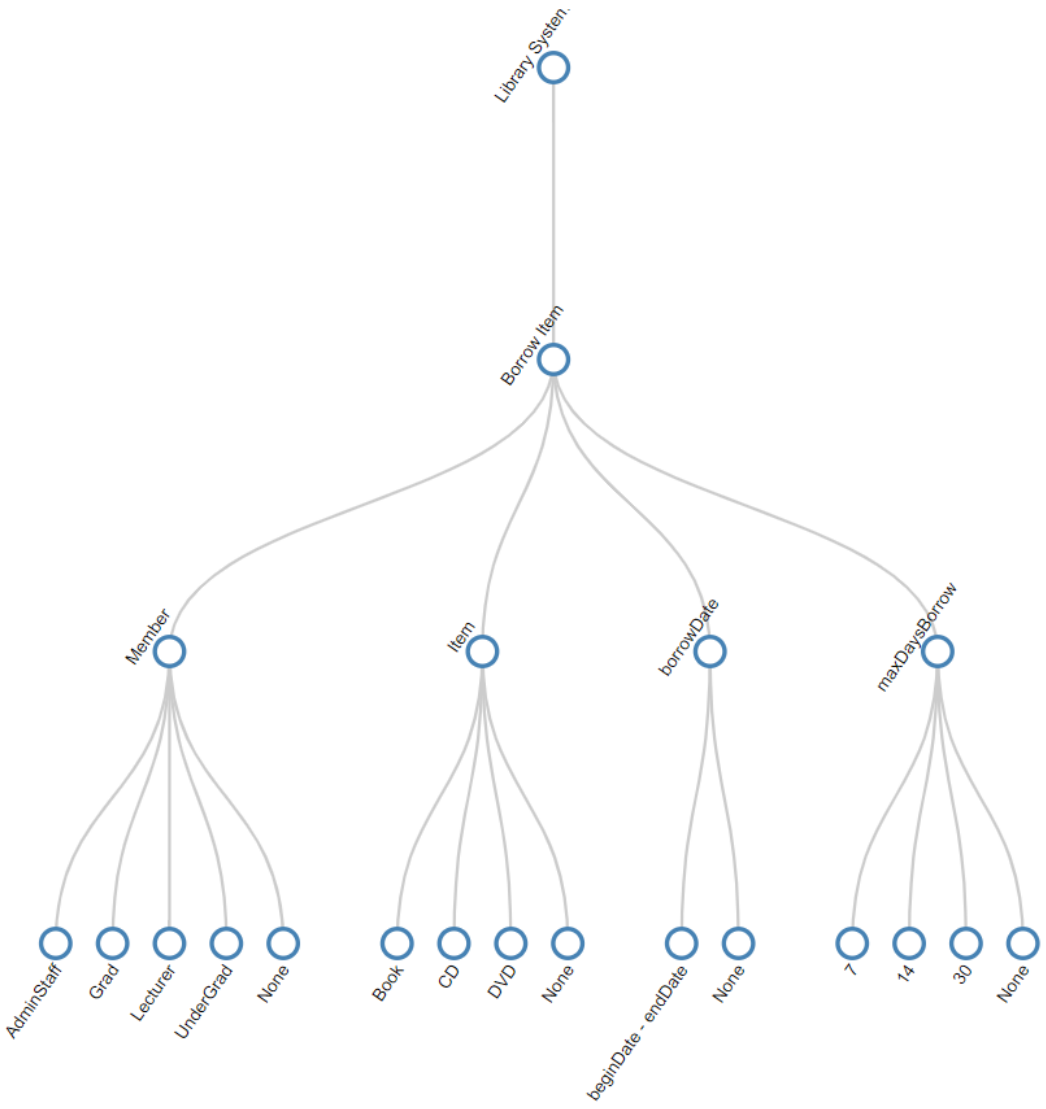
ชื่อตัวแปร	ชนิดตัวแปร	ความหมายตัวแปร	ช่วงข้อมูลที่เป็นไปได้
Member	string	สมาชิกห้องสมุด	AdminStaff, Grad, Lecturer, UnderGrad
Item	string	ทรัพยากรห้องสมุด	Book, CD, DVD
borrowDate	dateTime	วันที่ยืมทรัพยากร	begindate-endDate
maxDaysBorrow	integer	จำนวนวันสูงสุดที่สามารถยืมได้	7, 14, 30

จากตารางที่ 3.4 เป็นตัวอย่างของตัวแปรที่จะนำมาใช้ในการอธิบายกระบวนการสร้างกรณีทดสอบด้วยต้นไม้การจำแนกและการแบ่งชั้นสมมูล ประกอบไปด้วยตัวแปรทั้งหมด 4 ตัวแปร ได้แก่ Member, Item, borrowDate และ maxDaysBorrow ซึ่งได้มีการจัดการค่าของตัวแปร หรือช่วงของข้อมูลที่เป็นไปได้และจำเป็นต้องใช้ในกระบวนการสร้างกรณีทดสอบเรียบร้อยแล้ว รวมถึงการจัดการตัวแปรกรณีที่ระบบมีเงื่อนไขการตัดสินใจในการสร้างกรณีทดสอบเพื่อให้มีความครอบคลุมในการทดสอบมากยิ่งขึ้น

3.2.3.3 ขั้นตอนการสร้างกรณีทดสอบ

ในการสร้างกรณีทดสอบของระบบ ผู้วิจัยได้มีการพิจารณานำเทคนิคการรวมกันของการทดสอบแบบกล่องดำจำนวน 2 เทคนิคมาใช้ คือ เทคนิคการรวมกันของชั้นสมมูลและต้นไม้การจำแนก โดยสามารถอธิบายการสร้างกรณีทดสอบในแต่ละเทคนิคโดยทำการแบ่งหัวข้อในการอธิบายสำหรับกระบวนการสร้างกรณีทดสอบ ดังต่อไปนี้

1) การสร้างกรณีทดสอบด้วยต้นไม้การจำแนก จากตารางที่ 3.4 ตัวอย่างตัวแปรระบบจัดการห้องสมุด สามารถนำมาจำแนกตัวแปรโดยใช้ต้นไม้การจำแนก ซึ่งโหนดบนสุดจะเป็นชื่อของระบบที่ต้องการสร้างกรณีทดสอบ โหนดถัดมาจะเป็นชื่อของฟังก์ชันและชื่อตัวแปรภายในฟังก์ชันตามลำดับ ส่วนในโหนดสุดท้ายจะเป็นค่าของตัวแปร หรือช่วงของข้อมูลสำหรับการสร้างกรณีทดสอบ ซึ่งจะพิจารณาทั้งค่าของข้อมูลที่ถูก (valid) และค่าข้อมูลที่ไม่ถูกต้อง (invalid) จากตัวแปรทั้ง 4 ตัวแปร มีค่าข้อมูลที่ถูกต้องในคอลัมน์สุดท้ายของตารางที่ 3.4 ซึ่งเป็นค่าข้อมูลที่เป็นไปได้สำหรับใช้ในการสร้างกรณีทดสอบ ส่วนค่าข้อมูลที่ไม่ถูกต้องของแต่ละตัวแปรจะเป็นค่าที่อยู่นอกเหนือจากค่าที่สนใจ ในตัวอย่างนี้ค่าที่ไม่ถูกต้องของทั้ง 4 ตัวแปร คือ None ซึ่งการจำแนกตัวแปรโดยเทคนิคต้นไม้การจำแนกสามารถช่วยลดความซ้ำซ้อนของข้อมูลทดสอบทำให้ได้จำนวนกรณีทดสอบที่น้อยลงแต่ยังครอบคลุมทุก ๆ ฟังก์ชันของกระบวนการทดสอบตามเงื่อนไขความต้องการของระบบ สามารถแสดงการสร้างกรณีทดสอบด้วยต้นไม้การจำแนก ดังรูปที่ 3.15



รูปที่ 3.15 แผนภาพต้นไม้การจำแนก

2) การสร้างกรณีทดสอบโดยการแบ่งชั้นสมมูล จากตัวแปรที่จำแนกโดยใช้ต้นไม้การจำแนกข้างต้นสามารถนำมาแบ่งชั้นสมมูลแต่ละช่วงของค่าข้อมูลที่เป็นไปได้ โดยการแบ่งชั้นสมมูลผู้วิจัยได้นำรูปแบบการแบ่งชั้นสมมูลแบบสตรองโรบัส (strong robust) มาใช้กับงานวิจัยนี้ ซึ่งการแบ่งชั้นสมมูลในรูปแบบนี้จะพิจาณาทั้งค่าข้อมูลที่ถูกต้องและไม่ถูกต้องของทุกชั้นสมมูล ซึ่งจะช่วยให้การสร้างกรณีทดสอบครอบคลุมในทุก ๆ ค่าที่เป็นไปได้ของชั้นสมมูลทั้งหมด สามารถแสดงการแบ่งชั้นสมมูล ดังรูปที่ 3.16

รูปที่ 3.16 แผนภาพการแบ่งชั้นสมมูล

จากรูปที่ 3.16 จะเห็นได้ว่ามีการแบ่งชั้นสมมูลในทุก ๆ ค่าข้อมูลทั้งที่ถูกต้องและไม่ถูกต้อง โดยชั้นสมมูลที่ถูกต้องจะถูกสร้างเป็นชั้นสีดำส่วนชั้นสมมูลที่ไม่ถูกต้องจะสร้างเป็นชั้นสีแดง จากการแบ่งชั้นสมมูลของทั้ง 4 ตัวแปร มีค่าข้อมูลของแต่ละตัวแปร ดังนี้

2.1) ตัวแปร Member มีค่าข้อมูล 5 ค่า ได้แก่ AdminStaff, Grad, Lecturer, UnderGrad เป็นค่าข้อมูลที่ถูกต้อง และ None เป็นค่าข้อมูลที่ไม่ถูกต้อง

2.2) ตัวแปร Item มีค่าข้อมูล 4 ค่า ได้แก่ Book, CD, DVD เป็นค่าข้อมูลที่ถูกต้อง และ None เป็นค่าข้อมูลที่ไม่ถูกต้อง

2.3) ตัวแปร borrowDate มีค่าข้อมูล 1 ช่วงที่ถูกต้อง คือ ช่วงเริ่มต้นและสิ้นสุดของวันที่ทำการยืม และมีค่า None เป็นค่าข้อมูลที่ไม่ถูกต้อง

2.4) ตัวแปร maxDaysBorrow มีค่าข้อมูล 4 ค่า ได้แก่ 7, 14, 30 เป็นค่าข้อมูลที่ถูกต้อง และ None เป็นค่าข้อมูลที่ไม่ถูกต้อง โดยการคำนวณกรณีทดสอบจากชั้นสมมูลจะทำการคูณทุก ๆ ชั้นสมมูลด้วยผลคูณคาร์ทีเซียนซึ่งกรณีทดสอบจากการแบ่งชั้นสมมูลแบบสตรองโรบัสจะครอบคลุมทุกกรณีทดสอบที่เป็นไปได้ของชั้นสมมูลทั้งหมด

บทที่ 4

กรณีศึกษาของต้นแบบ

จากกรอบแนวคิดที่ได้มีการออกแบบไว้ เพื่อแสดงการทำงานและสนับสนุนกรอบแนวคิดของงานวิจัยที่นำเสนอ ผู้วิจัยได้แสดงผลลัพธ์ของ 2 กรณีศึกษาที่นำมาใช้ทดสอบกรอบแนวคิดนี้ ได้แก่ กรณีศึกษาที่ 1 ระบบจัดการห้องสมุด และกรณีศึกษาที่ 2 ระบบวิเคราะห์คำนวณค่าไต โดยมีวัตถุประสงค์เพื่อสร้างกรณีทดสอบด้วยออนโทโลยีความต้องการซอฟต์แวร์ และได้มีการนำเครื่องมือ ROO มาช่วยในการออกแบบและพัฒนาออนโทโลยีทำให้ข้อกำหนดความต้องการอยู่ในรูปแบบที่มีโครงสร้างและมีความชัดเจนมากยิ่งขึ้นก่อนจะนำไปสร้างกรณีทดสอบ (คำถามงานวิจัย ข้อ 1 และสอดคล้องกับวัตถุประสงค์งานวิจัย ข้อ 1) รวมถึงได้พิจารณาการรวมเทคนิคการทดสอบแบบกล่องดำจำนวน 2 เทคนิค คือ เทคนิคการรวมกันของชั้นสมมูลและต้นไม้การจำแนกมาใช้ในการสร้างกรณีทดสอบเพื่อให้มีความครอบคลุมในทุก ๆ กรณีของการทดสอบซอฟต์แวร์ (คำถามงานวิจัย ข้อ 2 และสอดคล้องกับวัตถุประสงค์งานวิจัย ข้อ 2)

4.1 กรณีศึกษาที่ 1 ระบบจัดการห้องสมุด (Library Management System)

ระบบจัดการห้องสมุด[1] เป็นระบบสำหรับจัดการทรัพยากรต่าง ๆ ของห้องสมุด ซึ่งสมาชิกภายในระบบสามารถทำการยืม-คืนทรัพยากร ได้แก่ หนังสือ ซีดี หรือดิวีดีได้ โดยสมาชิกแต่ละประเภทจะมีเงื่อนไขในการยืมที่แตกต่างกัน ทางผู้วิจัยได้พิจารณาในส่วนของฟังก์ชันการคืนเพื่อคำนวณหาค่าปรับของทรัพยากรที่ได้ทำการยืมไปแล้วนำกลับมาคืน โดยเงื่อนไขความต้องการไม่ว่าจะเป็นประเภทของสมาชิก เงื่อนไขการยืมของสมาชิกแต่ละประเภท หรือจำนวนค่าปรับต่อวัน จะเป็นความต้องการที่ใช้งานจริงในระบบจัดการห้องสมุดของมหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต

[1] http://library.phuket.psu.ac.th/index.php?option=com_content&view=article&id=93&Itemid=59&lang=th

4.1.1 ความต้องการของระบบ

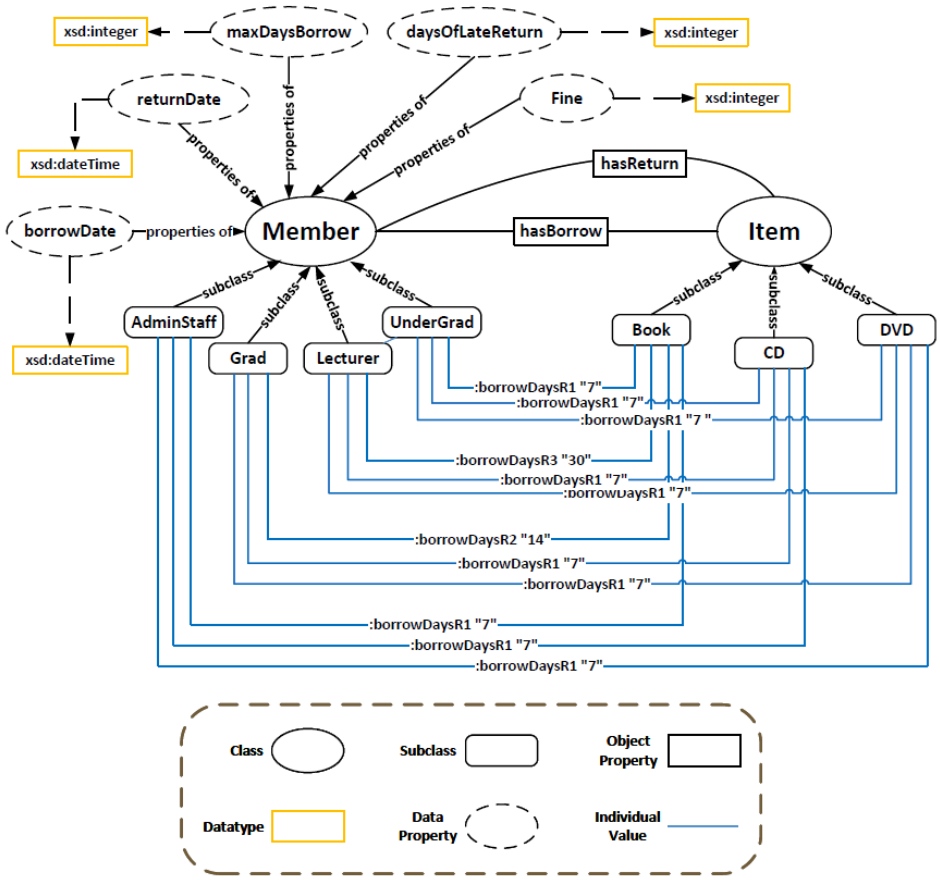
ความต้องการของระบบจัดการห้องสมุดที่นำมาพิจารณาในการออกแบบและพัฒนาออนโทโลยีประกอบไปด้วยความต้องการทั้งหมด 8 ความต้องการ ดังตารางที่ 4.1

ตารางที่ 4.1 ความต้องการระบบจัดการห้องสมุด

Requirement No.	Requirements
Requirement 01:	Members can borrow item including books, DVDs or CDs.
Requirement 02:	Members are classified into admin staff, lecturers, undergraduate and graduate students.
Requirement 03:	Books, DVDs, CDs must be disjointed.
Requirement 04:	Admin, staff, lecturers, undergrade and graduate students must be disjointed.
Requirement 05:	Maximum borrowing items and borrowing periods: 5 books per 7 days for admin staff and undergraduate students, 10 books per 14 days for graduate students, and 15 books per 30 days for lecturers.
Requirement 06:	Maximum borrowing items and borrowing periods: 3 discs per 7 days for all members.
Requirement 07:	Members can return the borrowed item.
Requirement 08:	If there is late return, members must pay fine.

4.1.2 การออกแบบและพัฒนาออนโทโลยี

จากตารางความต้องการระบบจัดการห้องสมุด (ตารางที่ 4.1) สามารถนำมาออกแบบและพัฒนาออนโทโลยีของระบบจัดการห้องสมุด ซึ่งประกอบไปด้วยคลาส ความสัมพันธ์และคุณสมบัติของคลาส สามารถแสดงโครงสร้างออนโทโลยี ดังรูปที่ 4.1



รูปที่ 4.1 ออนโทโลยีระบบจัดการห้องสมุด

จากรูปที่ 4.1 ออนโทโลยีความต้องการของระบบจัดการห้องสมุด ประกอบไปด้วยคลาสทั้งหมด 2 คลาสที่มีความสัมพันธ์กัน โดยคลาส *Member* จะเป็นสมาชิกของห้องสมุด ซึ่งประกอบไปด้วย *AdminStaff, Grad, Lecturer* และ *UnderGrad* ส่วนคลาส *Item* เป็นทรัพยากรห้องสมุดที่สามารยืมได้ ได้แก่ *Book, CD* และ *DVD*

นอกเหนือจากนั้นยังมีความสัมพันธ์ *hasBorrow* ซึ่งเป็นความสัมพันธ์แบบ ObjectProperty ระหว่างคลาส *Member* และ *Item* เพื่อใช้แสดงความสัมพันธ์ที่สมาชิกสามารถยืมทรัพยากรของห้องสมุดได้ และอีกหนึ่งความสัมพันธ์ คือ *hasReturn* เป็นความสัมพันธ์ที่สมาชิกสามารถคืนทรัพยากรห้องสมุดหลังจากได้มีการยืมไปแล้ว สามารถแสดงความสัมพันธ์ของคลาสดังตารางที่ 4.2

ตารางที่ 4.2 ความสัมพันธ์ระหว่างคลาสของระบบจัดการห้องสมุด

Object Property	Domain	Range
hasBorrow	Member	Item
hasReturn	Member	Item

นอกจากนี้ยังมีคุณสมบัติของข้อมูล หรือ DataProperty ของระบบจัดการห้องสมุด โดยมีโดเมน (Domain) คือ คลาส *Member* และมีประเภทข้อมูล (data type) เป็น Rane ของคุณสมบัติข้อมูลสามารถแสดงรายละเอียด ดังตารางที่ 4.3

ตารางที่ 4.3 คุณสมบัติข้อมูลของระบบจัดการห้องสมุด

Data Property	Domain	Range
borrowDate	Member	xsd:dateTime
maxDaysBorrow	Member	xsd:integer
returnDate	Member	xsd:dateTime
daysOfLateReturn	Member	xsd:integer
fine	Member	xsd:integer

รวมถึงค่าข้อมูล หรือ Individual ของระบบจัดการห้องสมุดจำนวน 3 ค่า ได้แก่ *7, 14* และ *30* ซึ่งเป็นเงื่อนไขจำนวนวันสูงสุดในการยืมทรัพยากรของห้องสมุดสำหรับสมาชิกแต่ละประเภทสามารแสดงรายละเอียดค่าข้อมูล ดังตารางที่ 4.4

ตารางที่ 4.4 ค่าข้อมูลของระบบจัดการห้องสมุด

Individual/ Instance	Value
borrowDaysR1	7
borrowDaysR2	14
borrowDaysR3	30

4.1.3 การแปลงออนโทโลยีจาก OWL เป็น XML

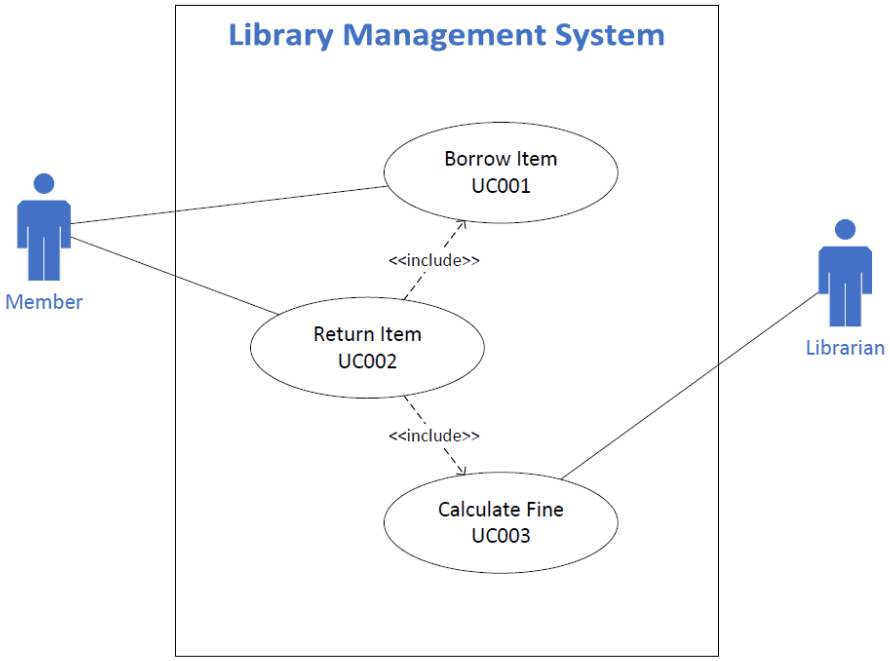
ในขั้นตอนการแปลงออนโทโลยีจาก OWL เป็น XML ผู้วิจัยได้นำออนโทโลยีจากขั้นตอนก่อนหน้าซึ่งมีการส่งออกไฟล์ในรูปแบบของไฟล์ .owl โดยใช้กฎการแปลงที่ได้มีการออกแบบไว้สำหรับรายละเอียดไฟล์ .owl ของออนโทโลยีระบบจัดการห้องสมุดจะแสดงไว้ในภาคผนวก ข. (รูปแบบไฟล์นำเข้าของต้นแบบ ในส่วน ข1) และสามารถแสดงโครงสร้างของ XML หลังจากได้ทำการแปลงเรียบร้อยแล้ว ดังรูปที่ 4.2

```
▼<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="AdminStaff" type="Member"/>
   <xs:element name="Book" type="Item" ref="maxDaysBorrow"/>
   <xs:element name="CD" type="Item" ref="maxDaysBorrow"/>
   <xs:element name="DVD" type="Item" ref="maxDaysBorrow"/>
   <xs:element name="Grad" type="Member"/>
   <xs:element name="Item"/>
   <xs:element name="Lecturer" type="Member"/>
  ▼<xs:element name="Member">
    ▼<xs:complexType>
      ▼<xs:sequence>
        ▼<xs:element name="hasBorrow" ref="Item">
          ▼<xs:complexType>
            ▼<xs:all>
               <xs:element name="borrowDate" type="xs:dateTime"/>
              ▼<xs:element name="maxDaysBorrow">
                ▼<xs:complexType>
                  ▼<xs:choice>
                     <xs:element name="7" type="xs:integer"/>
                     <xs:element name="14" type="xs:integer"/>
                     <xs:element name="30" type="xs:integer"/>
                   </xs:choice>
                 </xs:complexType>
               </xs:element>
             </xs:all>
           </xs:complexType>
         </xs:element>
        ▼<xs:element name="hasReturn" ref="Item">
          ▼<xs:complexType>
            ▼<xs:all>
               <xs:element name="daysOfLateReturn" type="xs:integer"/>
               <xs:element name="fine" type="xs:integer"/>
               <xs:element name="returnDate" type="xs:dateTime"/>
             </xs:all>
           </xs:complexType>
         </xs:element>
       </xs:sequence>
     </xs:complexType>
   </xs:element>
   <xs:element name="UnderGrad" type="Member"/>
 </xs:schema>
```

รูปที่ 4.2 โครงสร้าง XML ระบบจัดการห้องสมุด

4.1.4 การจัดการตัวแปรและต้นไม้การตัดสินใจ

ในขั้นตอนการจัดการตัวแปรและต้นไม้การตัดสินใจจะพิจารณาไฟล์ยูสเคสและไฟล์พจนานุกรมข้อมูลที่ได้จากการแปลงออนโทโลยีความต้องการซอฟต์แวร์ที่เป็นโครงสร้าง OWL ให้เป็นโครงสร้าง XML ซึ่งไฟล์ยูสเคสที่นำเข้ามาจะสอดคล้องกับฟังก์ชันที่ต้องการสร้างกรณีทดสอบ โดยรายละเอียดไฟล์ยูสเคสของระบบจัดการห้องสมุดที่นำเข้ามาจะแสดงไว้ในภาคผนวก ข. (รูปแบบไฟล์นำเข้าของต้นแบบ ในส่วน ข2) และสามารถแสดงแผนภาพยูสเคส ดังรูปที่ 4.3



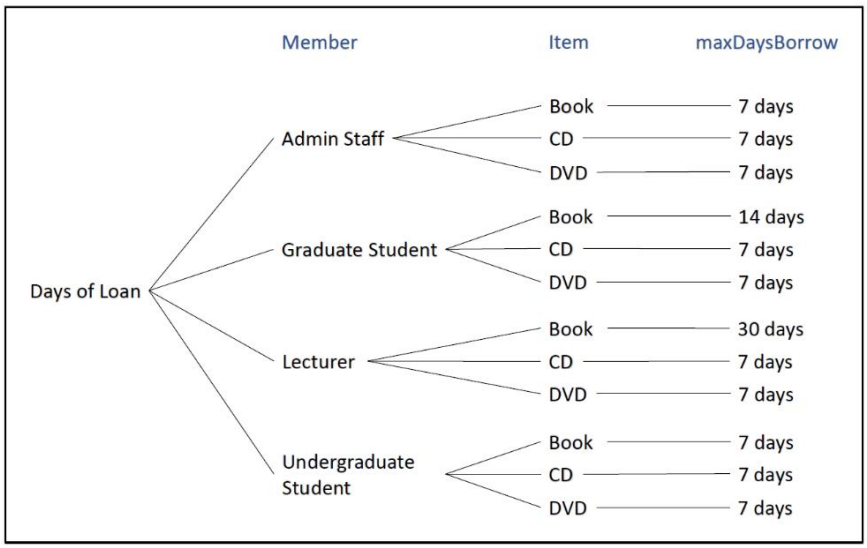
รูปที่ 4.3 แผนภาพยูสเคสระบบจัดการห้องสมุด

จากรูปที่ 4.3 ประกอบไปด้วยยูสเคสฟังก์ชันการทำงานของระบบจัดการห้องสมุดทั้งหมด 3 ยูสเคส ได้แก่ ฟังก์ชันการยืม (UC001: Borrow Item) ฟังก์ชันการคืน (UC002: Return Item) และฟังก์ชันคำนวณค่าปรับ (UC003: Calculate Fine) ซึ่งฟังก์ชัน UC002 มีการเรียกใช้ฟังก์ชัน UC001 และ UC003 โดยในการสร้างกรณีทดสอบจะพิจารณาฟังก์ชัน UC002 ซึ่งมีการเรียกใช้ฟังก์ชันการยืมและคำนวณค่าปรับของทรัพยากรห้องสมุดหลังจากทำการคืน จากพจนานุกรมข้อมูลและยูสเคสของระบบจัดการห้องสมุด สามารถกำหนดตัวแปรที่เกี่ยวข้องกับฟังก์ชัน UC002 ดังตารางที่ 4.5

ตารางที่ 4.5 ตัวแปรระบบจัดการห้องสมุด

ชื่อตัวแปร	ชนิดตัวแปร	ความหมายตัวแปร	ช่วงข้อมูลที่เป็นไปได้
Member	string	สมาชิกห้องสมุด	AdminStaff, Grad, Lecturer, UnderGrad
Item	string	ทรัพยากรห้องสมุด	Book, CD, DVD
borrowDate	dateTime	วันที่ยืมทรัพยากร	begindate-endDate
returnDate	dateTime	วันที่คืนทรัพยากร	borrowDate-endDate
daysOfLateReturn	integer	จำนวนวันคืนล่าช้า	0-2147483647 และ 0
fine	integer	ค่าปรับ	3-2147483647 และ 0
maxDaysBorrow	integer	จำนวนวันสูงสุดที่สามารถยืมได้	7, 14, 30

ในการจัดการตัวแปรจะพิจารณาถึงเงื่อนไขการตัดสินใจของระบบด้วย ซึ่งระบบจัดการห้องสมุดมีเงื่อนการตัดสินใจสำหรับการยืมทรัพยากรห้องสมุดของสมาชิกแต่ละประเภทแตกต่างกันออกไป โดยมีตัวแปรที่เกี่ยวข้องทั้งหมด 3 ตัวแปร ได้แก่ Member, Item และ maxDaysBorrow สามารถแสดงแผนภาพต้นไม้การตัดสินใจ ดังรูปที่ 4.4



รูปที่ 4.4 แผนภาพต้นไม้การตัดสินใจระบบจัดการห้องสมุด

จากรูปที่ 4.4 สามารถอธิบายรายละเอียดเงื่อนไขการตัดสินใจของระบบจัดการห้องสมุดสำหรับสมาชิกแต่ละประเภท ดังตารางที่ 4.6

ตารางที่ 4.6 เงื่อนไขการยืมทรัพยากรของสมาชิกแต่ละประเภท

ประเภทของสมาชิก	ความหมาย	จำนวนวันยืม (วัน)	
		หนังสือ	*ซีดี/ดิวีดี*
Admin Staff	บุคลากร	7	7
Graduate Student	นักศึกษาปริญญาโทและปริญญาเอก	14	
Lecturer	อาจารย์	30	
Undergraduate Student	นักศึกษาปริญญาตรี	7	

จากตารางที่ 4.6 สามารถอธิบายเงื่อนไขการตัดสินใจของสมาชิกแต่ละประเภทสำหรับการยืมหนังสือได้ ดังนี้ บุคลากรและนักศึกษาปริญญาตรี สามารถยืมหนังสือได้สูงสุด 7 วัน นักศึกษาปริญญาโทและปริญญาเอก สามารถยืมหนังสือได้สูงสุด 14 วัน และอาจารย์ สามารถยืมหนังสือได้สูงสุด 30 วัน ส่วนทรัพยากรที่เป็นซีดีและดิวีดี สมาชิกทุกประเภทสามารถยืมได้สูงสุด 7 วัน

เมื่อได้ทำการจัดการตัวแปรและต้นไม้การตัดสินใจที่เกี่ยวข้องเรียบร้อยแล้ว ในลำดับถัดไปจะเป็นกระบวนการสร้างกรณีทดสอบโดยจะนำตัวแปรที่จัดการเรียบร้อยแล้วมาจำแนก

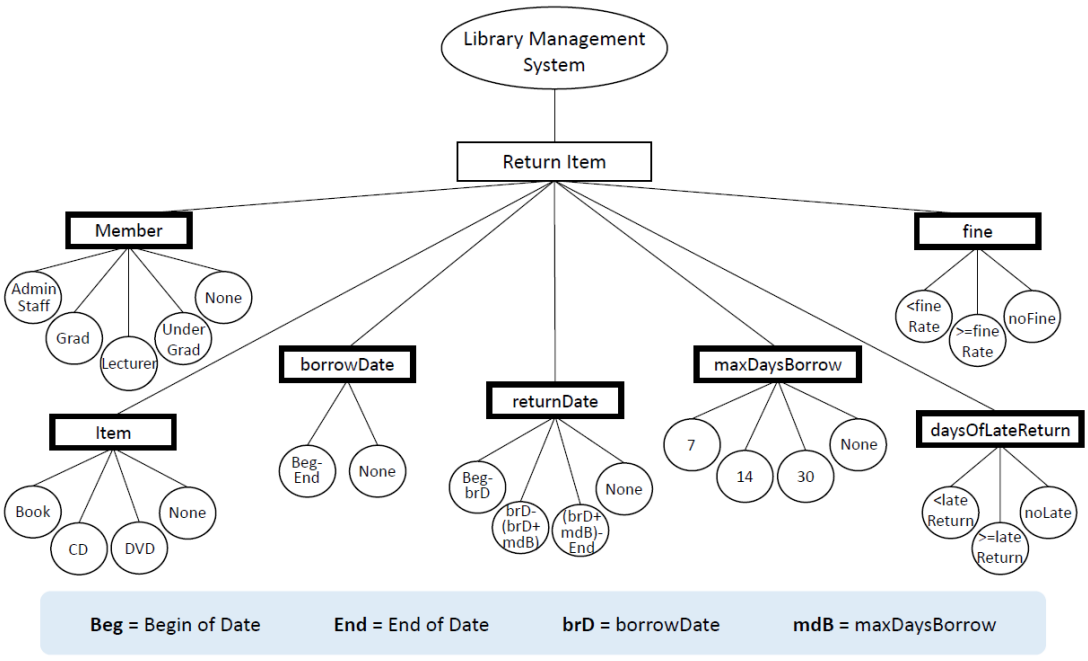
โดยใช้ต้นไม้การจำแนกและการแบ่งชั้นสมมูลของตัวแปรซึ่งเป็นเทคนิคการทดสอบแบบกล่องดำในการสร้างกรณีทดสอบ โดยจะอธิบายกระบวนการดำเนินงานและผลลัพธ์ของขั้นตอนการทำงานในหัวข้อถัดไป

4.1.5 การสร้างกรณีทดสอบ

ในกระบวนการสร้างกรณีทดสอบผู้วิจัยได้นำเทคนิคการรวมกันของชั้นสมมูลและต้นไม้การจำแนกมาใช้ในการสร้างกรณีทดสอบเพื่อให้มีความครอบคลุมในทุก ๆ เงื่อนไขและความต้องการของการทดสอบซอฟต์แวร์ โดยมีรายละเอียดแต่ละกระบวนการสร้างกรณีทดสอบ ดังต่อไปนี้

4.1.5.1 สร้างกรณีทดสอบโดยต้นไม้การจำแนก

สำหรับขั้นตอนการสร้างกรณีทดสอบด้วยต้นไม้การจำแนกจะนำตัวแปรที่ได้มีการจัดการค่าของตัวแปรและต้นไม้การตัดสินใจเรียบร้อยแล้วมาทำการจำแนกในแต่ละโหนดของต้นไม้การจำแนก โดยจะเริ่มสร้างตั้งแต่โหนดรากซึ่งจะเป็นชื่อของระบบที่ต้องการสร้างกรณีทดสอบ โหนดรองลงมาจะเป็นโหนดในระดับของฟังก์ชัน และถัดมาเป็นโหนดตัวแปรและค่าของตัวแปรซึ่งเป็นโหนดในลำดับสุดท้ายที่จะนำไปเป็นค่าข้อมูลสำหรับการสร้างกรณีทดสอบของเทคนิคการแบ่งชั้นสมมูล สามารถแสดงต้นไม้การจำแนกของระบบจัดการห้องสมุด ดังรูปที่ 4.5



รูปที่ 4.5 ต้นไม้การจำแนกระบบจัดการห้องสมุด

จากรูปที่ 4.5 สามารถจำแนกตัวแปรและค่าของตัวแปรได้ทั้งหมด 7 ตัวแปร ซึ่งมีความเกี่ยวข้องกับระบบจัดการห้องสมุด ได้แก่ Member, Item, borrowDate, returnDate, maxDaysBorrow, daysOfLateReturn และ fine โดยในแต่ละตัวแปรจะมีค่าของตัวแปรซึ่งเป็นค่าข้อมูลที่นำไปใช้ในการแบ่งชั้นสมมูลซึ่งจะพิจารณาทั้งค่าที่ถูกต้องและไม่ถูกต้องเพื่อให้ครอบคลุมกระบวนการสร้างกรณีทดสอบ ในส่วนของค่าข้อมูลแต่ละค่าที่เป็นไปได้จะถูกกล่าวถึงในการอธิบายการสร้างกรณีทดสอบด้วยเทคนิคการแบ่งชั้นสมมูลในขั้นตอนถัดไป

4.1.5.2 สร้างกรณีทดสอบจากการแบ่งชั้นสมมูล

จากการจำแนกตัวแปรด้วยเทคนิคต้นไม้การจำแนกข้างต้น สามารถนำค่าข้อมูลของแต่ละตัวแปรมาทำการแบ่งชั้นสมมูลเพื่อสร้างกรณีทดสอบของระบบจัดการห้องสมุด แสดงรายละเอียดการแบ่งช่วงของค่าข้อมูล ดังตารางที่ 4.7

ตารางที่ 4.7 การแบ่งชั้นสมมูลระบบจัดการห้องสมุด

Partition	Num of Partition	Valid Data				Invalid Data	
Member	5	AdminStaff (1)	Grad(2)	Lecturer(3)	UnderGrad(4)	None (5)	
Item	4	Book (6)	CD (7)	DVD (8)		None (9)	
borrowDate	2	Begin of Date – End of Date (10)				None (11)	
returnDate	4	borrowDate – (12) (borrowDate + maxDaysBorrow)		(borrowDate + (13) maxDaysBorrow) – End of Date		Begin of Date – (14) borrowDate	None (15)
maxDaysBorrow (Fix Rate)	4	7 (16)	14 (17)	30 (18)		None (19)	
daysOfLateReturn (Output)	3	No Late (20)		1 –Late Return (21)		Smallest Number – 1 (22)	
fine (Output)	3	No Fine (23)		Fine Rate – (24) (Fine Rate x daysOfLateReturn)		Smallest Number – Fine Rate (25)	

จากตารางที่ 4.7 สามารถอธิบายรายละเอียดของตัวแปรและค่าข้อมูลจากการแบ่งชั้นสมมูลโดยพิจารณาทั้งค่าข้อมูลที่ถูกต้อง (Valid Data) และไม่ถูกต้อง (Invalid Data) ซึ่งจำนวนชั้นสมมูลที่เกี่ยวข้องทั้งหมดมี 25 ชั้นสมมูล และตัวแปรทั้งหมด 7 ตัวแปร โดยมีตัวแปรที่เป็น Input จำนวน 4 ตัวแปร และตัวแปร fix rate (ค่าคงที่ที่สามารถเป็นได้แค่ค่าใดค่าหนึ่งเท่านั้น) จำนวน 1 ตัวแปร และตัวแปร output จำนวน 2 ตัวแปร ดังนี้

<u>ตัวแปร Input ระบบจัดการห้องสมุด</u>

1) ตัวแปร Member สามารถแบ่งชั้นสมมูลได้ 5 ชั้น ซึ่งประกอบด้วยประเภทของสมาชิกที่เป็น AdminStaff, Grad, Lecturer, UnderGrad ซึ่งเป็นค่าข้อมูลที่ถูกต้อง และ None เป็นค่าข้อมูลที่ไม่ถูกต้อง

2) ตัวแปร Item แบ่งชั้นสมมูลได้ 4 ชั้น คือ Book, CD, DVD เป็นค่าข้อมูลที่ถูกต้อง และ None เป็นค่าข้อมูลที่ไม่ถูกต้อง

3) ตัวแปร borrowDate แบ่งชั้นสมมูลได้ 2 ชั้น คือ (beginDate-endDate) เป็นค่าข้อมูลที่ถูกต้องซึ่งเป็นช่วงของวันที่การยืม โดยช่วงของการยืมจะเริ่มจากจุดเริ่มต้นจนถึงจุดสิ้นสุดของวันที่ยืม และ มีค่า None เป็นค่าข้อมูลที่ไม่ถูกต้อง

4) ตัวแปร returnDate แบ่งชั้นสมมูลได้ 4 ชั้น โดยชั้นสมมูลที่ถูกต้องมี 2 ชั้น คือ borrowDate-(borrowDate+maxDaysBorrow) เป็นช่วงของการคืนทรัพยากรหลังจากได้มีการยืมจนถึงจำนวนวันสูงสุดของสมาชิกแต่ละประเภทที่สามารถยืมได้ โดยช่วงนี้จะไม่มีการเรียกเก็บค่าปรับเนื่องจากยังไม่ครบกำหนดวันคืน และ (borrowDate+maxDaysBorrow)-endDate เป็นช่วงการคืนหลังจากครบกำหนดวันคืนไปแล้ว ซึ่งช่วงนี้จะมีการเรียกเก็บค่าปรับตามข้อตกลงของการคืนล่าช้า ส่วนค่าข้อมูลที่ไม่ถูกต้องมี 2 ชั้น คือ (beginDate-borrowDate) เนื่องจากเป็นช่วงการคืนที่อยู่นอกเหนือจากช่วงของข้อมูลที่ถูกต้องหรือช่วงก่อนวันยืม ซึ่งการคืนจะไม่เกิดขึ้นถ้าหากยังไม่ได้มีการยืมทรัพยากร และค่า None เป็นค่าข้อมูลที่ไม่ถูกต้องเช่นเดียวกัน

<u>ตัวแปร Fix Rate ระบบจัดการห้องสมุด</u>

5) ตัวแปร maxDaysBorrow แบ่งชั้นสมมูลได้ 4 ชั้น คือ 7, 14, 30 ซึ่งเป็นค่าวันสูงสุดของการยืมสำหรับสมาชิกแต่ละประเภทและเป็นค่าข้อมูลที่ถูกต้อง ส่วนค่าข้อมูลที่ไม่ถูกต้องมี 1 ชั้น คือ None

<u>ตัวแปร Output ระบบจัดการห้องสมุด</u>

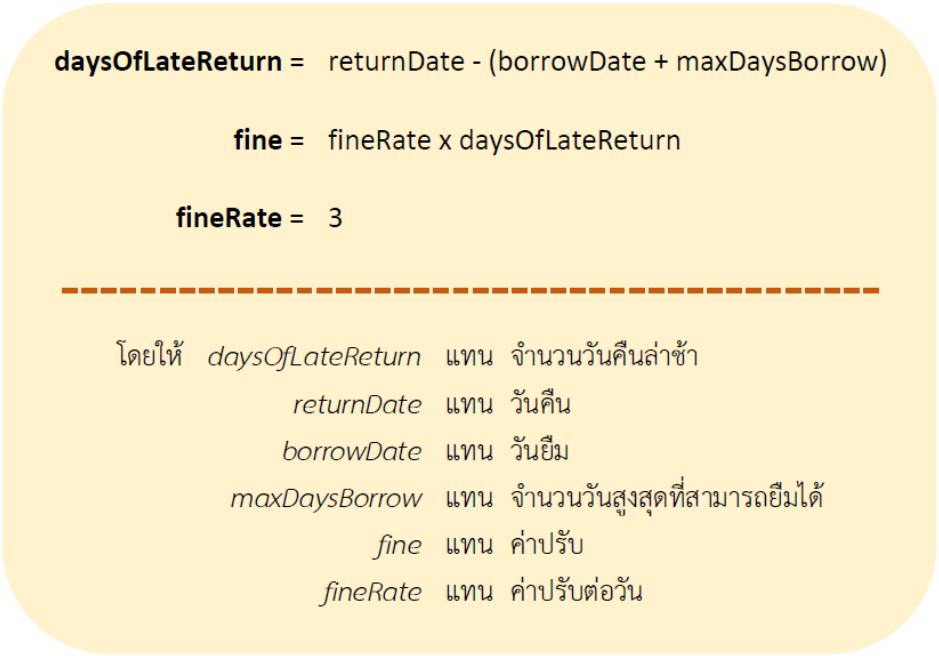
6) ตัวแปร daysOfLateReturn แบ่งชั้นสมมูลได้ 3 ชั้น เป็นชั้นสมมูลที่ถูกต้อง 2 ชั้น คือ “No Late” โดยชั้นนี้จะสอดคล้องกับตัวแปร returnDate ในช่วงของวันคืนที่ไม่เกินกำหนดซึ่งจำนวนวันล่าช้าจะเท่ากับ 0 หรือ ไม่มีวันคืนล่าช้า และค่า (1-Late Return) จะเป็นช่วงของจำนวนวันที่มีการคืนล่าช้า โดยจะมีค่าตั้งแต่ 1 วันเป็นต้นไป และมีค่าข้อมูลที่ไม่ถูกต้อง คือ (Smallest Number-1) โดยจะเป็นค่าที่น้อยกว่า 1 ลงไป

7) ตัวแปร fine แบ่งชั้นสมมูลได้ 3 ชั้น เป็นชั้นสมมูลที่ถูกต้อง 2 ชั้น คือ “No Fine” ซึ่งจะสอดคล้องกับ “No Late” ถ้าไม่มีวันคืนล่าช้าจะไม่มีการคิดค่าปรับ และ (fineRate-(fineRate x daysOfLateRetur)) จะเป็นช่วงที่มีการคิดค่าปรับกรณีมีการคืนล่าช้า โดยช่วงของข้อมูลจะเริ่มตั้งแต่จำนวนค่าปรับต่อวันคูณกับวันที่มีการคืนล่าช้า และมีค่าข้อมูลที่ไม่ถูกต้อง คือ (Smallest Number-fineRate) ซึ่งเป็นค่าที่น้อยกว่าจำนวนค่าปรับลงไปจนถึงค่าติดลบ

4.1.5.3 กรณีทดสอบระบบจัดการห้องสมุด

จากการแบ่งชั้นสมมูลของค่าข้อมูลข้างต้นสามารถนำมาสร้างกรณีทดสอบโดยคำนวณจากผลคูณคาร์ทีเซียนของทุก ๆ ชั้นสมมูลที่เป็นตัวแปรแบบ input มาคูณเข้าด้วยกัน

และเนื่องจากระบบจัดการห้องสมุดต้องมีการคำนวณในส่วนค่าของตัวแปร output ซึ่งจะเป็นการคำนวน 2 ส่วน ได้แก่ การคำนวณจำนวนวันคืนล่าช้าและการคำนวณค่าปรับ โดยใช้สูตรการคำนวณดังรูปที่ 4.6



รูปที่ 4.6 สูตรการคำนวณค่าปรับระบบจัดการห้องสมุด

จากรูปที่ 4.6 สามารถอธิบายได้ว่าการคำนวนวันคืนล่าช้าจะคำนวณจากวันคืน–(วันยืม+จำนวนวันสูงสุดที่สามารถยืมได้) โดยวันคืนจะต้องอยู่ในช่วงที่มีการคืนเกินกำหนด เมื่อได้จำนวนวันคืนล่าช้าแล้วสามารถนำมาคำนวณค่าปรับได้จาก (3*จำนวนวันคืนล่าช้า) โดยที่ 3 เป็นจำนวนค่าปรับต่อวันของระบบจัดการห้องสมุด กล่าวคือ ถ้ามีการคืนช้ากว่ากำหนดจะถูกคิดค่าปรับจำนวน 3 บาทต่อวัน โดยสูตรการคำนวณนี้จะเป็นการคิดค่าปรับของตัวแปร *daysOfLateReturn* และ *fine* ซึ่งเป็นตัวแปร output ของระบบจัดการห้องสมุด

ในการสร้างกรณีทดสอบจะใช้ผลคูณคาร์ทีเซียนของทุกชั้นสมมูลที่เป็นตัวแปร Input ทั้ง 4 ตัวแปรที่เกี่ยวข้องของระบบจัดการห้องสมุด ซึ่งสามารถสร้างกรณีทดสอบได้ทั้งหมด 160 (5*4*2*4) กรณีทดสอบ โดยกรณีทดสอบทั้งหมดจะแสดงไว้ในภาคผนวก ค. (กรณีทดสอบของทั้ง 2 กรณีศึกษา ในส่วน ค1) และสามารถแสดงตัวอย่างกรณีทดสอบ ดังตารางที่ 4.8

ตารางที่ 4.8 ตัวอย่างกรณีทดสอบระบบจัดการห้องสมุด

Test Case	Member	Item	borrow Date	return Date	maxDays Borrow	daysOfLate Return	fine	Comments
1	AdminStaff	Book	1/7/2019	2/7/2019	7	0	0	Valid (no late return)
2	AdminStaff	Book	5/7/2019	15/7/2019	7	3	9	Valid (late return)
...	...	...	...	...	...	...	...	...
67	Lecturer	Book	10/7/2019	9/6/2019	30	-1	-3	Invalid (return date)
68	Lecturer	Book	10/7/2019	None	30	None	None	Invalid (none return date)
...	...	...	...	...	...	...	...	...
159	None	None	None	10/8/2019	None	None	None	Invalid
160	None	None	None	None	None	None	None	Invalid

จากตารางที่ 4.8 เป็นผลลัพธ์ที่ได้จากการสร้างกรณีทดสอบของระบบจัดการห้องสมุด สามารถอธิบายตัวอย่างผลลัพธ์ของกรณีทดสอบที่ 1 ได้ว่า สมาชิกประเภท AdminStaff ทำการยืมหนังสือเมื่อวันที่ 1/7/2019 และได้มีการคืนในวันที่ 2/7/2019 จากระยะเวลาการยืมกับเงื่อนไขวันยืมสูงสุดของสมาชิก คือ 7 วัน ซึ่งไม่เกินกำหนดของการคืนทำให้จำนวนวันคืนล่าช้าเป็น 0 และไม่ต้องชำระค่าปรับ แตกต่างจากกรณีทดสอบที่ 2 ซึ่งมีการคืนเกินกำหนดจำนวน 3 วัน ทำให้ต้องชำระค่าปรับ 9 (3*3) บาท และทั้งสองกรณีนี้เป็นกรณีทดสอบที่ถูกต้อง นอกเหนือจากนั้นในกรณีทดสอบที่ไม่ถูกต้อง ตัวอย่างเช่น กรณีทดสอบที่ 67 เป็นกรณีทดสอบที่มีค่าข้อมูลไม่ถูกต้อง คือ ตัวแปร returnDate ซึ่งมีวันคืนก่อนหน้าวันยืมทำให้กรณีนี้ไม่ถูกต้อง เนื่องจากถ้ายังไม่มีการยืมจะไม่สามารถทำการคืนได้ เป็นต้น

4.2 กรณีศึกษาที่ 2 ระบบวิเคราะห์คำนวณค่าไต (GFR module)

ระบบวิเคราะห์คำนวณค่าไต[2] เป็นระบบในการแนะนำแนวทางการรักษาคนไข้ที่มีภาวะไตไม่ปกติให้กับแพทย์เพื่อช่วยให้สามารถรักษาคนไข้ได้อย่างถูกต้องและเหมาะสม โดยมีการคำนวณจากผลอัตราการกรองของไต (Glomerular Filtration Rate: GFR) ได้แก่ เพศ อายุ และผลครีเอตินีน (SCr) มาทำการจับคู่กับผลยูรีนครีเอตีนิน (UO) ของคนไข้ เพื่อแสดงการแปลผลระยะ

[2] https://kb.psu.ac.th/psukb/bitstream/2016/13552/1/434689.pdf

ความล้มเหลวของไตให้แพทย์ทราบว่าอยู่ในระยะใด ซึ่งสามารถช่วยวิเคราะห์การรักษาให้มีความเหมาะสมกับความเสี่ยงของคนไข้แต่ละระยะความล้มเหลวของไตที่มีภาวะผิดปกติ โดยกรณีศึกษานี้ได้มีการอ้างอิงมาจากวิทยานิพนธ์ของทิวาทิพย์ (ทิวาทิพย์ ศรีรักษา, 2561) เป็นระบบโอเพนซอร์ส (opensource) และเป็นส่วนหนึ่งของระบบสารสนเทศโรงพยาบาล (Hospital Information System: HIS) ที่มีชื่อเรียกระบบว่า “HospitalOS” เป็นระบบที่มีการติดตั้งและใช้งานจริงในโรงพยาบาลชุมชนและคลินิกมากกว่า 100 แห่งทั่วประเทศไทย โดยผู้วิจัยได้นำกรณีศึกษานี้มาใช้สำหรับทดสอบการทำงานของต้นแบบว่าสามารถสร้างกรณีทดสอบได้สอดคล้องกับกรณีศึกษาของวิทยานิพนธ์ดังกล่าวหรือไม่ เพื่อเป็นการยืนยันความถูกต้องในการทำงานของกรอบแนวคิดที่ผู้วิจัยได้มีการนำเสนอเอาไว้

4.2.1 ความต้องการของระบบ

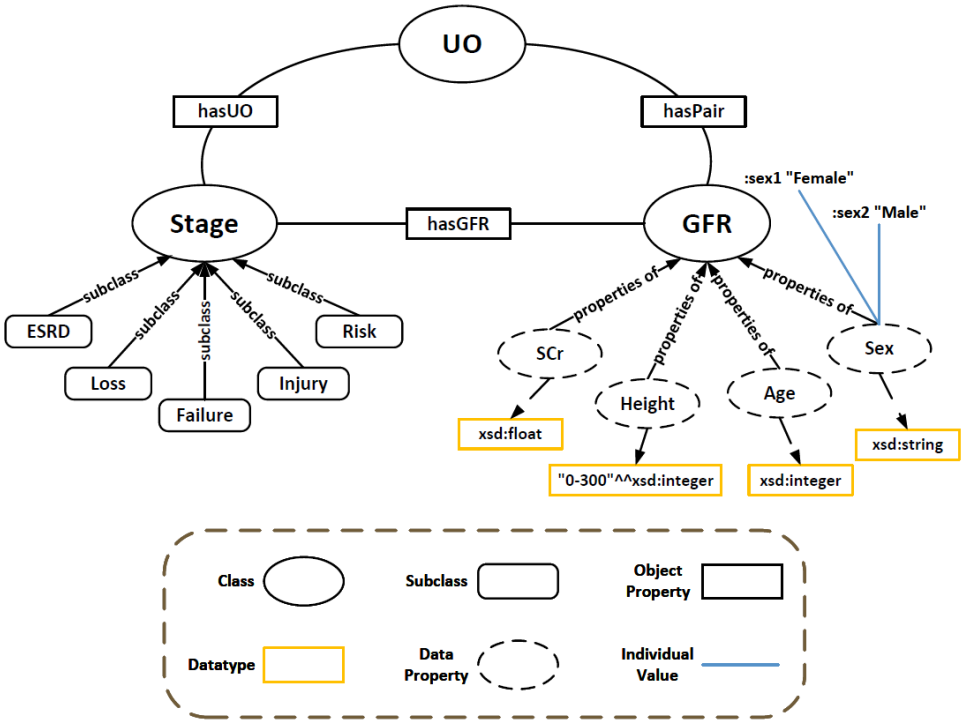
ความต้องการของระบบวิเคราะห์คำนวณค่าไตที่นำมาพิจารณาในการออกแบบและพัฒนาออนโทโลยีประกอบไปด้วยความต้องการทั้งหมด 4 ความต้องการ ดังตารางที่ 4.9

ตารางที่ 4.9 ความต้องการระบบวิเคราะห์คำนวณค่าไต

Requirement No.	Requirements
Requirement 01:	GFR is calculated with sex, age, height and SCr.
Requirement 02:	Stage of GFR includes ESRD, Loss, Failure, Injury and Risk.
Requirement 03:	ESRD, Loss, Failure, Injury, Risk must be disjointed.
Requirement 04:	Stage is paired with GFR and UO.

4.2.2 การออกแบบและพัฒนาออนโทโลยี

จากตารางที่ 4.9 ความต้องการระบบวิเคราะห์คำนวณค่าไต สามารถนำมาออกแบบและพัฒนาออนโทโลยีของระบบ ซึ่งประกอบไปด้วยคลาส ความสัมพันธ์ และคุณสมบัติของคลาส สามารถแสดงโครงสร้างออนโทโลยี ดังรูปที่ 4.7



รูปที่ 4.7 ออนโทโลยีระบบวิเคราะห์คำนวณค่าไต

จากรูปที่ 4.7 ออนโทโลยีความต้องการของระบบวิเคราะห์คำนวณค่าไต ประกอบไปด้วยคลาสทั้งหมด 3 คลาสและมีความสัมพันธ์กัน โดยคลาส *Stage* จะเป็นระยะความล้มเหลวของไต ซึ่งประกอบไปด้วย *ESRD, Loss, Failure, Injury* และ *Risk* ส่วนคลาส *GFR* เป็นอัตราการกรองของไต และคลาส *UO* เป็นผลยูรีนคลีเอตินีน

นอกเหนือจากนั้นยังมีความสัมพันธ์ *hasPair* ซึ่งเป็นความสัมพันธ์แบบ ObjectProperty ระหว่างคลาส *GFR* และ *UO* เพื่อใช้แสดงความสัมพันธ์ที่มีการจับคู่ค่าของทั้ง 2 คลาสนี้เพื่อแปรผลระยะความล้มเหลวของไต และความสัมพันธ์ *hasGFR* เป็นความสัมพันธ์ระหว่างคลาส *Stage* และ *GFR* และความสัมพันธ์ *hasUO* เป็นความสัมพันธ์ระหว่างคลาส *Stage* และ *UO* ซึ่งความสัมพันธ์ *hasGFR* และ *hasUO* จะสัมพันธ์กับคลาส *Stage* เพื่อให้สามารถแปรผลระยะความล้มเหลวของไตได้ สามารถแสดงความสัมพันธ์ระหว่างคลาสของระบบวิเคราะห์คำนวณค่าไตดังตารางที่ 4.10

ตารางที่ 4.10 ความสัมพันธ์ระหว่างคลาสของระบบวิเคราะห์คำนวณค่าไต

Object Property	Domain	Range
hasPair	GFR	UO
hasGFR	Stage	GFR
hasUO	Stage	UO

นอกจากนี้ยังมีคุณสมบัติของข้อมูล หรือ DataProperty ของระบบวิเคราะห์คำนวณค่าไต โดยมีโดเมน (Domain) คือ คลาส *GFR* และมีประเภทข้อมูล (data type) เป็น Rane ของคุณสมบัติข้อมูล รวมถึงมีข้อจำกัดของคุณสมบัติข้อมูล *Height* ตั้งแต่ 0-300 ในรูปที่ 4.7 และมีรายละเอียดของคุณสมบัติข้อมูล ดังตารางที่ 4.11

ตารางที่ 4.11 คุณสมบัติข้อมูลของระบบวิเคราะห์คำนวณค่าไต

Data Property	Domain	Range
Sex	GFR	xsd:string
Age	GFR	xsd:integer
Height	GFR	xsd:integer
SCr	GFR	xsd:float

รวมถึงค่าข้อมูล หรือ Individual ของระบบวิเคราะห์คำนวณค่าไตจำนวน 2 ค่า ได้แก่ *Female* และ *Male* ซึ่งเป็นเพศของคนไข้ สามารถแสดงรายละเอียดค่าข้อมูล ดังตารางที่ 4.12

ตารางที่ 4.12 ค่าข้อมูลของระบบวิเคราะห์คำนวณค่าไต

Individual/ Instance	Value
Sex1	Female
Sex2	Male

4.2.3 การแปลงออนโทโลยีจาก OWL เป็น XML

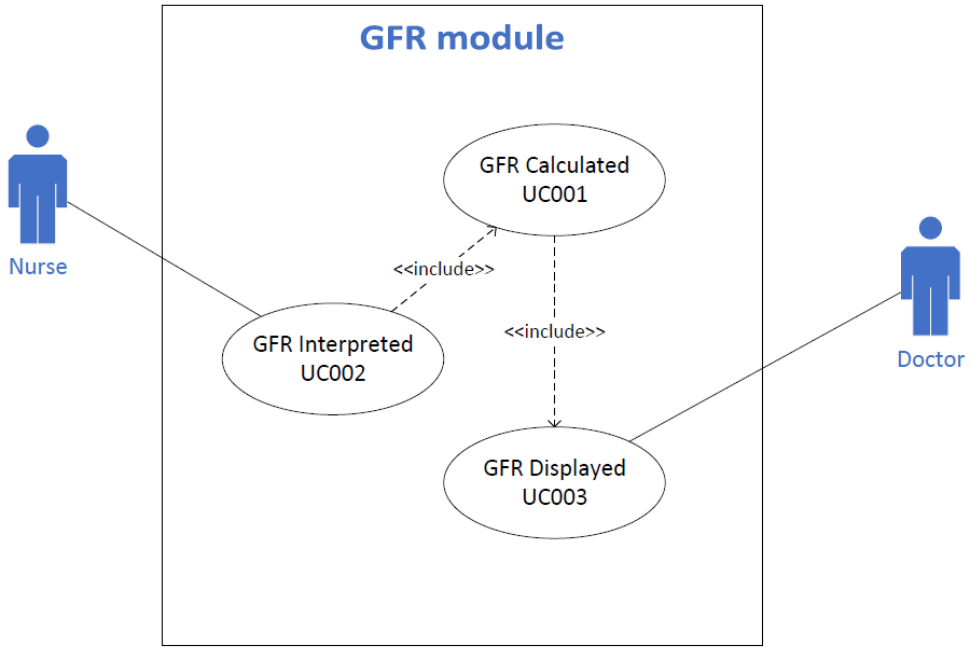
ในขั้นตอนการแปลงออนโทโลยีจาก OWL เป็น XML ผู้วิจัยได้นำออนโทโลยีจากขั้นตอนก่อนหน้าซึ่งมีการส่งออกไฟล์ในรูปแบบของไฟล์ .owl โดยใช้กฎการแปลงที่ได้มีการออกแบบไว้ สำหรับรายละเอียดไฟล์ .owl ของออนโทโลยีระบบวิเคราะห์คำนวณค่าไตจะแสดงไว้ในภาคผนวก ข. ((รูปแบบไฟล์นำเข้าของต้นแบบ ในส่วน ข3) และสามารถแสดงโครงสร้างของ XML หลังจากได้ทำการแปลงเรียบร้อยแล้ว ดังรูปที่ 4.8

```
▼<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="ESRD" type="Stage"/>
   <xs:element name="Failure" type="Stage"/>
  ▼<xs:element name="GFR">
   ▼<xs:complexType>
    ▼<xs:sequence>
       <xs:element name="hasPair" ref="UO"/>
       <xs:element name="Age" type="xs:integer"/>
      ▼<xs:element name="Height">
       ▼<xs:simpleType>
        ▼<xs:restiction base="xs:integer">
           <xs:minInclusive value="0"/>
           <xs:maxInclusive value="300"/>
         </xs:restiction>
        </xs:simpleType>
       </xs:element>
       <xs:element name="SCr" type="xs:float"/>
      ▼<xs:element name="Sex">
       ▼<xs:complexType>
        ▼<xs:choice>
           <xs:element name="Female" type="xs:string"/>
           <xs:element name="Male" type="xs:string"/>
         </xs:choice>
        </xs:complexType>
       </xs:element>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
   <xs:element name="Injury" type="Stage"/>
   <xs:element name="Loss" type="Stage"/>
   <xs:element name="Risk" type="Stage"/>
  ▼<xs:element name="Stage">
   ▼<xs:complexType>
    ▼<xs:sequence>
       <xs:element name="hasGFR" ref="GFR"/>
       <xs:element name="hasUO" ref="UO"/>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
   <xs:element name="UO"/>
 </xs:schema>
```

รูปที่ 4.8 โครงสร้าง XML ระบบวิเคราะห์คำนวณค่าไต

4.2.4 การจัดการตัวแปรและต้นไม้การตัดสินใจ

ในขั้นตอนการจัดการตัวแปรและต้นไม้การตัดสินใจจะพิจารณาไฟล์ยูสเคสและไฟล์พจนานุกรมข้อมูลที่ได้จากการแปลงออนโทโลยีความต้องการซอฟต์แวร์ที่เป็นโครงสร้าง OWL ให้เป็นโครงสร้าง XML ซึ่งไฟล์ยูสเคสที่นำเข้ามาจะสอดคล้องกับฟังก์ชันที่ต้องการสร้างกรณีทดสอบ โดยรายละเอียดไฟล์ยูสเคสของระบบวิเคราะห์คำนวณค่าไตที่นำเข้ามาจะแสดงไว้ในภาคผนวก ข. (รูปแบบไฟล์นำเข้าของต้นแบบ ในส่วน ข4) และสามารถแสดงแผนภาพยูสเคส ดังรูปที่ 4.9



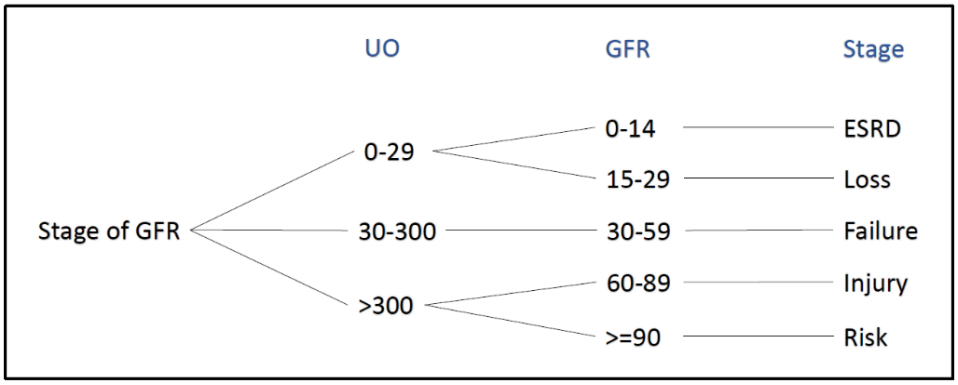
รูปที่ 4.9 แผนภาพยูสเคสระบบวิเคราะห์คำนวณค่าไต

จากรูปที่ 4.9 ประกอบไปด้วยยูสเคสฟังก์ชันการทำงานของระบบวิเคราะห์คำนวณค่าไตทั้งหมด 3 ยูสเคส ได้แก่ ฟังก์ชันการคำนวณค่า GFR (UC001: GFR Calculated) ฟังก์ชันการแปลผลระดับความล้มเหลวของไต (UC002: GFR Interpreted) และฟังก์ชันแสดงระยะความล้มเหลวของไต (UC003: GFR Displayed) ซึ่งฟังก์ชัน UC002 มีการเรียกใช้ฟังก์ชัน UC001 และฟังก์ชัน UC001 มีการเรียกใช้ฟังก์ชัน UC003 โดยในการสร้างกรณีทดสอบจะพิจารณาฟังก์ชัน UC002 ซึ่งมีการเรียกใช้ฟังก์ชันการคำนวณค่า GFR เพื่อแปลผลระยะความล้มเหลวของไต จากพจนานุกรมข้อมูลและยูสเคสของระบบวิเคราะห์คำนวณค่าไต สามารถกำหนดตัวแปรที่เกี่ยวข้องกับฟังก์ชัน UC002 ดังตารางที่ 4.13

ตารางที่ 4.13 ตัวแปรระบบวิเคราะห์คำนวณค่าไต

ชื่อตัวแปร	ชนิดตัวแปร	ความหมายตัวแปร	ช่วงข้อมูลที่เป็นไปได้
Sex	string	เพศ	Female, Male
Age	integer	อายุ	0-120
Height	integer	ส่วนสูง	0-300
SCr	float	ผลแลปครีเอตินีน	0.0-10.0
UO	integer	ผลยูรีนครีเอตินีน	0-2147483647
GFR	integer	ค่า GFR	3-2147483647
Stage	string	ระยะความล้มเหลวของไต	ESRD, Loss, Failure, Injury, Risk

ในการจัดการตัวแปรจะพิจารณาถึงเงื่อนไขการตัดสินใจของระบบด้วย ซึ่งระบบวิเคราะห์คำนวณค่าไตมีเงื่อนไขในการแปลผลระยะความล้มเหลวของไตในแต่ละสถานะที่แตกต่างกัน โดยมีตัวแปรที่เกี่ยวข้องทั้งหมด 3 ตัวแปร ได้แก่ UO, GFR และ Stage สามารถแสดงแผนภาพต้นไม้การตัดสินใจ ดังรูปที่ 4.10



รูปที่ 4.10 แผนภาพต้นไม้การตัดสินใจระบบวิเคราะห์คำนวณค่าไต

จากรูปที่ 4.10 สามารถอธิบายรายละเอียดเงื่อนไขการตัดสินใจของระบบวิเคราะห์คำนวณค่าไตในการแปลผลระยะความล้มเหลวของไต ดังตารางที่ 4.14

ตารางที่ 4.14 เงื่อนไขการแปลผลระยะความล้มเหลวของไต

ค่า GFR	ค่ายูรีนครีเอตินีน (UO)	ระยะความล้มเหลวของไต (Stage)
0-14	0-29	ไตวายระยะสุดท้าย (ESRD)
15-29	0-29	สูญเสียการทำงานของไต (Loss)
30-59	30-300	ไตล้มเหลว (Failure)
60-89	มากกว่า 300	ไตเริ่มเสียหาย (Injury)
มากกว่าเท่ากับ 90	มากกว่า 300	เสี่ยงไตวาย (Risk)

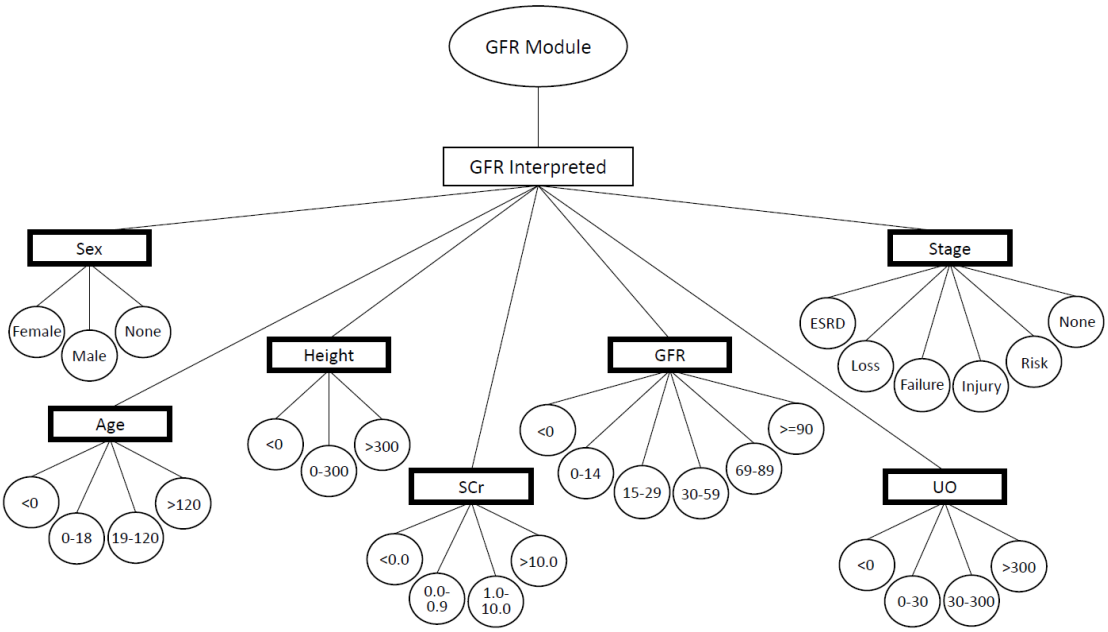
จากตารางที่ 4.14 สามารถอธิบายการแปลผลระยะความล้มเหลวของไตได้จากการจับคู่ค่า GFR และค่า UO ซึ่งมีทั้งหมด 5 ระดับของความล้มเหลว ตัวอย่างเช่น ถ้าคนไข้มีค่า GFR อยู่ในช่วง 30-59 และมีค่า UO 30-300 สามารถแปลผลได้ว่า คนไข้อยู่ในระยะ Failure ซึ่งเป็นระยะของไตล้มเหลว เป็นต้น หลังจากได้ทำการจัดการตัวแปรและต้นไม้การตัดสินใจที่เกี่ยวข้องกับการสร้างกรณีทดสอบเรียบร้อยแล้ว ในลำดับถัดไปจะเป็นกระบวนการสร้างกรณีทดสอบโดยจะนำตัวแปรที่จัดการเรียบร้อยแล้วมาจำแนกโดยใช้ต้นไม้การจำแนกและการแบ่งชั้นสมมูลของตัวแปรตามลำดับ โดยจะอธิบายกระบวนการดำเนินงานและผลลัพธ์ของขั้นตอนการทำงานในหัวข้อถัดไป

4.2.5 การสร้างกรณีทดสอบ

ในกระบวนการสร้างกรณีทดสอบผู้วิจัยได้นำเทคนิคการรวมกันของชั้นสมมูลและต้นไม้การจำแนกมาใช้ในการสร้างกรณีทดสอบเพื่อให้มีความครอบคลุมในทุก ๆ เงื่อนไขและความต้องการของการทดสอบซอฟต์แวร์ โดยมีรายละเอียดแต่ละกระบวนการสร้างกรณีทดสอบ ดังต่อไปนี้

4.2.5.1 สร้างกรณีทดสอบโดยต้นไม้การจำแนก

สำหรับขั้นตอนการสร้างกรณีทดสอบด้วยต้นไม้การจำแนกจะนำตัวแปรที่มีการจัดการค่าของตัวแปรและต้นไม้การตัดสินใจเรียบร้อยแล้วมาทำการจำแนกในแต่ละโหนดของต้นไม้การจำแนก โดยจะเริ่มสร้างตั้งแต่โหนดรากซึ่งจะเป็นชื่อของระบบที่ต้องการสร้างกรณีทดสอบ โหนดรองลงมาจะเป็นโหนดในระดับของฟังก์ชัน และถัดมาเป็นโหนดตัวแปรและค่าของตัวแปรซึ่งเป็นโหนดในลำดับสุดท้ายที่จะนำไปใช้เป็นค่าข้อมูลสำหรับสร้างกรณีทดสอบของเทคนิคการแบ่งชั้นสมมูล สามารถแสดงต้นไม้การจำแนกของระบบวิเคราะห์คำนวณค่าไต ดังรูปที่ 4.11



รูปที่ 4.11 ต้นไม้การจำแนกระบบวิเคราะห์คำนวณค่าไต

จากรูปที่ 4.11 สามารถจำแนกตัวแปรและค่าของตัวแปรได้ทั้งหมด 7 ตัวแปร ที่เกี่ยวข้องกับระบบวิเคราะห์คำนวณค่าไต ได้แก่ Sex, Age, Height, SCr, GFR, UO และ Stage โดยในแต่ละตัวแปรจะมีค่าของตัวแปรซึ่งเป็นค่าข้อมูลที่จะนำไปใช้ในการแบ่งชั้นสมมูลซึ่งจะพิจารณาทั้งค่าที่ถูกต้องและไม่ถูกต้องเพื่อให้ครอบคลุมกระบวนการสร้างกรณีทดสอบ โดยทางผู้วิจัยได้มี

การพิจารณาข้อมูลที่เป็นช่วงเพิ่มเติม คือ ช่วงของค่าที่ไม่สิ้นสุด (Infinity) ตัวอย่างเช่น ตัวแปร Age จะมีช่วงของค่าข้อมูลที่น้อยกว่า 0 และมากกว่า 120 ซึ่งเป็นช่วงข้อมูลที่ไม่สิ้นสุด ผู้วิจัยได้มีการกำหนดช่วงข้อมูลนี้เป็น 2147483647 ในกรณีที่เป็นช่วงของค่าจำนวนเต็มบวก และ -2147483647 ในกรณีที่เป็นช่วงของค่าติดลบตามลำดับ (โดยตัวเลขนี้เป็นขนาดของข้อมูลแบบ integer ซึ่งเป็นค่าที่เพียงพอต่อการสร้างข้อมูลทดสอบและเพื่อให้สามารถใช้งานต้นแบบบนอุปกรณ์ที่เป็นระบบปฏิบัติการแบบ 32-bit และ 64-bit ได้) ในส่วนของค่าข้อมูลแต่ละค่าที่เป็นไปได้จะถูกกล่าวถึงในการอธิบายการสร้างกรณีทดสอบด้วยเทคนิคการแบ่งชั้นสมมูลในขั้นตอนถัดไป

4.2.5.2 สร้างกรณีทดสอบจากการแบ่งชั้นสมมูล

จากการจำแนกตัวแปรด้วยเทคนิคต้นไม้การจำแนกข้างต้น สามารถนำค่าข้อมูลของแต่ละตัวแปรมาทำการแบ่งชั้นสมมูลเพื่อสร้างกรณีทดสอบของระบบวิเคราะห์คำนวณค่าไต แสดงรายละเอียดการแบ่งช่วงของค่าข้อมูล ดังตารางที่ 4.15

ตารางที่ 4.15 การแบ่งชั้นสมมูลระบบวิเคราะห์คำนวณค่าไต

Partition	Num of Partition	Valid Data					Invalid Data	
Sex	3	Female (1)		Male (2)			None (3)	
Age	4	0–18 (4)		19–120 (5)			<0 (6)	>120 (7)
Height	3	0–300 (8)					<0 (9)	>300 (10)
SCr	4	0.0–0.9 (11)		1.0–10.0 (12)			<0.0 (13)	>10.0 (14)
GFR (Fix Rate)	6	0–14 (15)	15–29 (16)	30–59 (17)	60–89 (18)	>=90 (19)	<0 (20)	
UO (Fix Rate)	4	0–29 (21)		30–300 (22)		>300 (23)	<0 (24)	
Stage (Output)	6	ESRD (25)	Loss (26)	Failure (27)	Injury (28)	Risk (29)	None (30)	

จากตารางที่ 4.15 สามารถอธิบายรายละเอียดของตัวแปรและค่าข้อมูลจากการแบ่งชั้นสมมูลโดยพิจารณาทั้งค่าข้อมูลที่ถูกต้อง (Valid Data) และไม่ถูกต้อง (Invalid Data) ซึ่งจำนวนชั้นสมมูลที่เกี่ยวข้องทั้งหมดมี 30 ชั้นสมมูล และตัวแปรทั้งหมด 7 ตัวแปร โดยมีตัวแปรที่เป็น input จำนวน 4 ตัวแปร และตัวแปร fix rate (ค่าคงที่ที่สามารถเป็นได้แค่ค่าใดค่าหนึ่งเท่านั้น) จำนวน 2 ตัวแปร และตัวแปร output จำนวน 1 ตัวแปร ดังนี้

<u>ตัวแปร Input ระบบวิเคราะห์คำนวณค่าไต</u>

1) ตัวแปร Sex แบ่งชั้นสมมูลได้ 3 ชั้น คือ Female และ Male ซึ่งเป็นค่าข้อมูลที่ถูกต้อง และ None เป็นค่าข้อมูลที่ไม่ถูกต้อง

2) ตัวแปร Age แบ่งชั้นสมมูลได้ 4 ชั้น คือ ช่วงอายุตั้งแต่ 0-18 และ 19-120 ซึ่งเป็นช่วงของค่าข้อมูลที่ถูกต้อง และค่าข้อมูลที่ไม่ถูกต้อง คือ ช่วงอายุน้อยกว่า 0 และมากกว่า 120

3) ตัวแปร Height แบ่งชั้นสมมูลได้ 3 ชั้น คือ 0-300 ซึ่งเป็นช่วงความสูงที่ถูกต้อง ส่วนช่วงน้อยกว่า 0 และมากกว่า 300 เป็นช่วงความสูงที่ไม่ถูกต้อง

4) ตัวแปร SCr แบ่งชั้นสมมูลได้ 4 ชั้น คือ 0.0-0.9 และ 1.0-10.0 เป็นช่วงของค่าข้อมูลที่ถูกต้อง และช่วงน้อยกว่า 0.0 และมากกว่า 10.0 เป็นช่วงค่าข้อมูลที่ไม่ถูกต้อง

ตัวแปร Fix Rate ระบบจัดการห้องสมุด

5) ตัวแปร GFR แบ่งชั้นสมมูลได้ 6 ชั้น คือ 0-14, 15-29, 30-59, 60-89 และมากกว่าเท่ากับ 90 ซึ่งเป็นช่วงของค่าข้อมูลที่ถูกต้อง และช่วงที่ไม่ถูกต้อง คือ ช่วงที่น้อยกว่า 0

6) ตัวแปร UO แบ่งชั้นสมมูลได้ 4 ชั้น คือ 0-29, 30-300 และมากกว่า 300 เป็นช่วงของค่าข้มูลที่ถูกต้อง และช่วงที่ไม่ถูกต้อง คือ ช่วงที่น้อยกว่า 0

ตัวแปร Output ระบบวิเคราะห์คำนวณค่าไต

7) ตัวแปร Stage แบ่งชั้นสมมูลได้ 6 ชั้น คือ ESRD, Loss, Failure, Injury และ Risk ซึ่งเป็นค่าข้อมูลระยะความล้มเหลวของไตที่ถูกต้อง และค่า None เป็นค่าข้อมูลที่ไม่ถูกต้อง

4.2.5.3 กรณีทดสอบระบบวิเคราะห์คำนวณค่าไต

จากการแบ่งชั้นสมมูลของค่าข้อมูลข้างต้นสามารถนำมาสร้างกรณีทดสอบโดยคำนวณจากผลคูณคาร์ทีเซียนของทุก ๆ ชั้นสมมูลที่เป็นตัวแปรแบบ Input มาคูณเข้าด้วยกัน และเนื่องจากระบบวิเคราะห์คำนวณค่าไตมีการคำนวณค่า GFR ของคนไข้เพื่อนำไปจับคู่กับค่า UO ในการแปลผลระยะความล้มเหลวของไต โดยใช้สูตรการคำนวณ ดังตารางที่ 4.16

ตารางที่ 4.16 สูตรการคำนวณค่า GFR

Age	Sex	Serum creatinine result	Formula
>18	Female	SCr <= 0.7	$GFR = 144 \times (SCr/0.7)^{-0.329} (0.993)^{Age}$
		SCr > 0.7	$GFR = 144 \times (SCr/0.7)^{-1.209} (0.993)^{Age}$
	Male	SCr <= 0.9	$GFR = 144 \times (SCr/0.9)^{-0.411} (0.993)^{Age}$
		SCr > 0.9	$GFR = 144 \times (SCr/0.9)^{-1.209} (0.993)^{Age}$
<=18	Female/Male	GFR = (0.413 x (Height(cm)) / SCr(mg/dL)	

จากสูตรการคำนวณหาค่า GFR มีสูตรการคำนวณที่แตกต่างกันตามช่วงอายุ โดยสามารถจำแนกสูตรการคำนวณได้ตาม 2 ช่วงอายุ คือ ช่วงเด็กอายุตั้งแต่ 0-18 ปี และผู้ใหญ่ช่วงอายุมากกว่า 18 ปีขึ้นไป ซึ่งมีสูตรการคำนวณตามที่แสดงในตารางที่ 4.16 สูตรการคำนวณค่า GFR

ในการสร้างกรณีทดสอบจะใช้ผลคูณคาร์ทีเซียนของทุกชั้นสมมูลที่เป็นตัวแปร Input ทั้ง 4 ตัวแปรที่เกี่ยวข้องของระบบวิเคราะห์คำนวณค่าไต ซึ่งสามารถสร้างกรณีทดสอบได้ทั้งหมด 144 (3*4*3*4) กรณีทดสอบ ซึ่งจำนวนกรณีทดสอบที่ได้สอดคล้องกับจำนวนกรณีทดสอบในงานวิจัยของทิวาทิพย์ (ทิวาทิพย์ ศรีรักษา, 2561) โดยกรณีทดสอบทั้งหมดจะแสดงไว้ในภาคผนวก ค. (กรณีทดสอบของทั้ง 2 กรณีศึกษา ในส่วน ค2) และสามารถแสดงตัวอย่างกรณีทดสอบ ดังตารางที่ 4.17

ตารางที่ 4.17 ตัวอย่างกรณีทดสอบระบบวิเคราะห์คำนวณค่าไต

Test Case	Sex	Age	Height	SCr	GFR	UO	Stage	Comments
1	Female	10	142	0.8	73	450	Injury	Valid
2	Female	18	165	4.5	15	27	Loss	Valid
...	...	...	...	...	...	...	...	...
61	Male	29	177	0.3	356	555	Risk	Valid
62	Male	65	104	7.2	7	10	ESRD	Valid
...	...	...	...	...	...	...	...	...
143	None	200	784	-7.2	-426	-487	None	Invalid
144	None	200	1140	110.2	11558	65487	None	Invalid

จากตารางที่ 4.17 เป็นผลลัพธ์ที่ได้จากการสร้างกรณีทดสอบของระบบวิเคราะห์คำนวณค่าไต สามารถอธิบายตัวอย่างผลลัพธ์กรณีทดสอบที่ถูกต้องของกรณีทดสอบที่ 1 ได้ว่า การคำนวณค่า GFR ของคนไข้ที่เป็นเพศหญิงอายุน้อยกว่า 18 ปี มีส่วนสูง 142 ซม. และมีค่า SCr เท่ากับ 0.8 คำนวณแล้วค่า GFR อยู่ที่ 73 เมื่อนำมาจับคู่กับค่า UO ซึ่งอยู่ที่ 450 สามารถแปลผลระยะความล้มเหลวของไตได้ว่าอยู่ในระยะ Injury หรือระยะไตเริ่มเสียหาย ซึ่งสอดคล้องกับเงื่อนไขการแปลผลระยะความล้มเหลวของไต และตัวอย่างกรณีทดสอบที่ 144 เป็นกรณีทดสอบที่ไม่ถูกต้องเนื่องจากมีค่าข้อมูลซึ่งอยู่นอกช่วงที่สนใจทั้งหมด เป็นต้น

บทที่ 5

กระบวนการพัฒนาต้นแบบ

ในบทนี้จะเป็นการอธิบายกระบวนการในการพัฒนาต้นแบบตามกรอบแนวคิดของงานวิจัยที่ได้ทำการออกแบบไว้ โดยจะกล่าวถึงเครื่องมือต่าง ๆ ที่นำมาใช้ในการพัฒนาต้นแบบ การวิเคราะห์และออกแบบต้นแบบ รวมถึงรายละเอียดหน้าจอการทำงานและส่วนต่อประสานของต้นแบบที่จะทำให้เห็นรูปแบบการทำงานและภาพรวมการทำงานของต้นแบบได้ชัดเจนขึ้น โดยวัตถุประสงค์ในการพัฒนาต้นแบบเพื่อเป็นเครื่องมือที่ช่วยสนับสนุนและยืนยันความถูกต้องในการทำงานของกรอบแนวคิดงานวิจัยที่ได้นำเสนอไว้ในบทที่ 3 หัวข้อที่ 3.2 กรอบแนวคิดของงานวิจัย ในการสร้างกรณีทดสอบด้วยออนโทโลยีความต้องการซอฟต์แวร์ รวมถึงการนำเทคนิคการรวมกันของชั้นสมมูลและต้นไม้การจำแนกมาใช้สามารถช่วยเพิ่มขีดความสามารถในการสร้างกรณีทดสอบให้มีความครอบคลุมในทุก ๆ กรณีของการทดสอบซอฟต์แวร์ด้วยเช่นกัน (คำถามงานวิจัย ข้อ 3 และสอดคล้องกับวัตถุประสงค์งานวิจัย ข้อ 3) โดยมีรายละเอียดในการพัฒนาต้นแบบแต่ละส่วนดังต่อไปนี้

5.1 เครื่องมือที่ใช้ในการพัฒนาต้นแบบ

ในการพัฒนาต้นแบบผู้วิจัยได้ศึกษาเทคโนโลยีและเครื่องมือที่ได้รับความนิยมเพื่อนำมาใช้ในการพัฒนาต้นแบบของงานวิจัย โดยได้มีการจำแนกประเภทของเครื่องมือในการพัฒนาออกเป็น 2 ส่วน ได้แก่ 1) เครื่องมือที่เป็นฮาร์ดแวร์ (Hardware) เป็นอุปกรณ์ที่ใช้สำหรับพัฒนาต้นแบบ และ 2) เครื่องมือที่เป็นซอฟต์แวร์ (Software) ซึ่งจะประกอบไปด้วยระบบปฏิบัติการและโปรแกรมต่าง ๆ โดยมีรายละเอียดแต่ละส่วน ดังต่อไปนี้

5.1.1 ฮาร์ดแวร์ (Hardware)

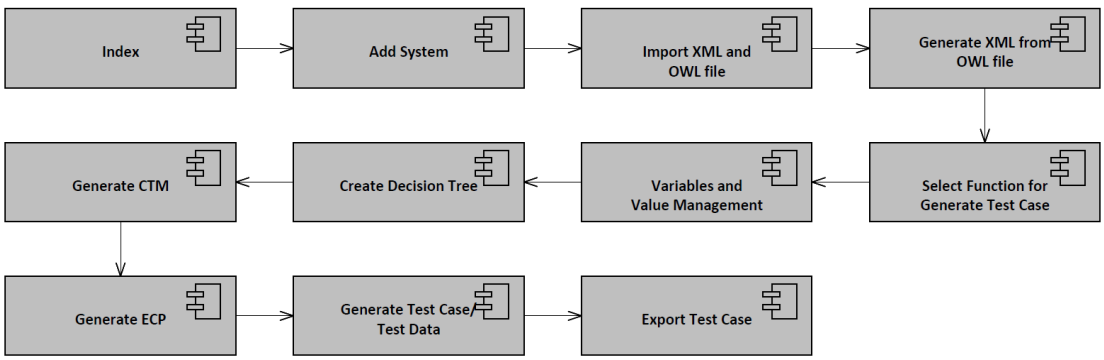
1) คอมพิวเตอร์โน้ตบุ๊ก
   ก. หน่วยประมวลผล Intel Corei7 8th Gen 2.2 GHz
   ข. หน่วยความจำหลัก 16 GB
   ค. ฮาร์ดดิสก์ 1TB+SSD 128GB

5.1.2 ซอฟต์แวร์ (Software)

1) ระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 10 (Microsoft Windows 10)
2) โปรแกรมวิชวล สตูดิโอ โค้ด เวอร์ชัน 1.65.2 (Visual Studio Code 1.65.2)
3) โปรแกรมอโดบี อิลลัสเตรเตอร์ ซีซี 2019 (Adobe Illustrator CC 2019)
4) โปรแกรมโหนด เจเอส เวอร์ชัน 16.14.0 (Node.js 16.14.0)
5) เครื่องมือแองกูลาร์ ซี แอล ไอ เวอร์ชัน 11.2.4 (Angular CLI 11.2.4)
6) โปรแกรมเว็บเบราว์เซอร์ กูเกิล โครม เวอร์ชัน 99.0.4844.82 (Google chrome 99.0.4844.82)

5.2 สถาปัตยกรรมของต้นแบบ

ในกระบวนการพัฒนาต้นแบบผู้วิจัยได้มีการออกแบบขั้นตอนและฟังก์ชันการทำงานต่าง ๆ ซึ่งสอดคล้องกับกรอบแนวคิดของงานวิจัยในการสร้างกรณีทดสอบด้วยออนโทโลยีความต้องการซอฟต์แวร์ที่ได้ออกแบบไว้ สามารถแสดงสถาปัตยกรรมของต้นแบบโดยใช้แผนภาพส่วนประกอบในการอธิบาย ดังรูปที่ 5.1



รูปที่ 5.1 แผนภาพส่วนประกอบของต้นแบบ

จากรูปที่ 5.1 แผนภาพส่วนประกอบของต้นแบบประกอบไปด้วยส่วนการทำงานทั้งหมด 11 ส่วนที่มีการทำงานร่วมกัน ดังนี้

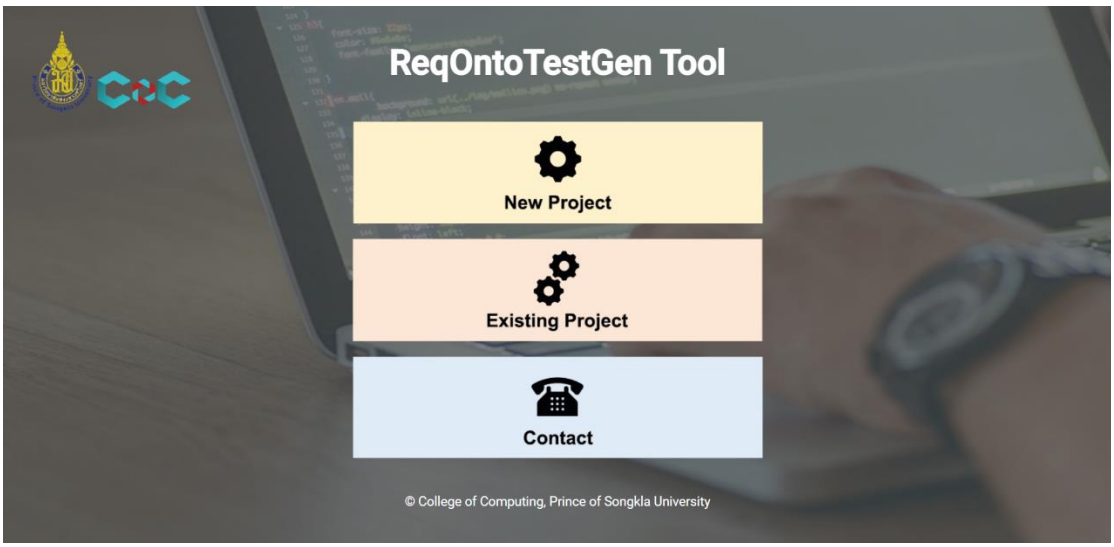
1) Index: ส่วนหน้าจอเริ่มต้นการทำงานของต้นแบบ
2) Add System: ส่วนฟังก์ชันการเพิ่มระบบสำหรับสร้างกรณีทดสอบ
3) Import XML and OWL file: ส่วนฟังก์ชันการนำเข้าไฟล์ XML และ OWL
4) Generate XML from OWL file: ส่วนฟังก์ชันการแปลงออนโทโลยีจาก OWL เป็น XML

5) Select Function for Generate Test Case: ส่วนฟังก์ชันการเลือกฟังก์ชันของระบบที่ต้องการสร้างกรณีทดสอบ
6) Variable and Value Management: ส่วนฟังก์ชันการจัดการตัวแปร
7) Create Decision Tree: ส่วนฟังก์ชันการสร้างต้นไม้การตัดสินใจ
8) Generate CTM: ส่วนฟังก์ชันการสร้างกรณีทดสอบด้วยเทคนิคต้นไม้การจำแนก
9) Generate ECP: ส่วนฟังก์ชันการสร้างกรณีทดสอบด้วยเทคนิคการแบ่งชั้นสมมูล
10) Generate Test Case/Test Data: ส่วนฟังก์ชันการสร้างกรณีทดสอบและข้อมูลทดสอบ
11) Export Test Case: ส่วนฟังก์ชันการส่งออกกรณีทดสอบในรูปแบบไฟล์ excel

5.3 หน้าจอการทำงานและส่วนต่อประสานกับผู้ใช้ (User Interface)

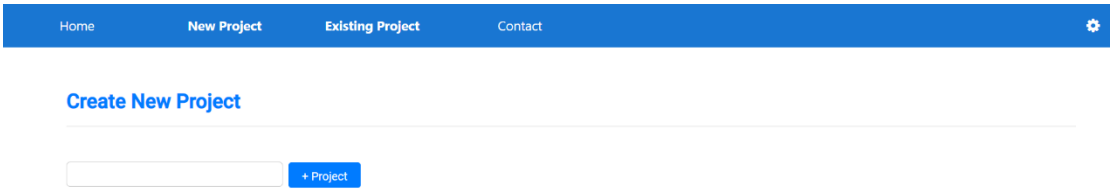
จากรูปแผนภาพส่วนประกอบของต้นแบบ (รูปที่ 5.1) ประกอบด้วยฟังก์ชันต่าง ๆ ที่มีการทำงานร่วมกัน สามารถอธิบายด้วยหน้าจอการทำงานและส่วนต่อประสานกับผู้ใช้ ดังต่อไปนี้

1) หน้าจอเริ่มต้นการทำงานของต้นแบบประกอบไปด้วยเมนูการทำงานทั้งหมด 3 หน้าจอหลัก ๆ ได้แก่ เมนู (New Project) เป็นการสร้างโปรเจคใหม่สำหรับสร้างกรณีทดสอบ เมนู (Existing Project) เชื่อมไปยังหน้าโปรเจคที่เคยมีการสร้างกรณีทดสอบไว้ซึ่งสามารถกลับมาเรียกดู แก้ไข หรือดาวน์โหลดไปใช้งานต่อได้ และสุดท้ายเมนู (Contact) เชื่อมไปยังหน้าการติดต่อสอบถาม ดังรูปที่ 5.2



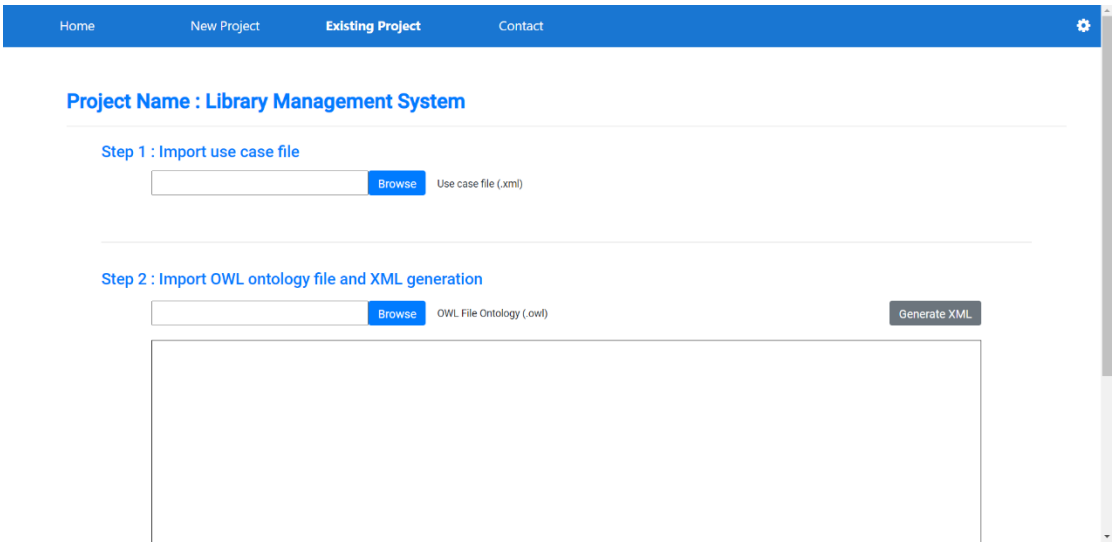
รูปที่ 5.2 หน้าจอเริ่มต้นการทำงานของต้นแบบ

2) หน้าจอสร้างโปรเจคใหม่ (New Project) เป็นหน้าจอเริ่มต้นสำหรับการสร้างกรณีทดสอบโดยให้ผู้ใช้ทำการสร้างชื่อโปรเจคของระบบที่จะทำการสร้างกรณีทดสอบ ดังรูปที่ 5.3



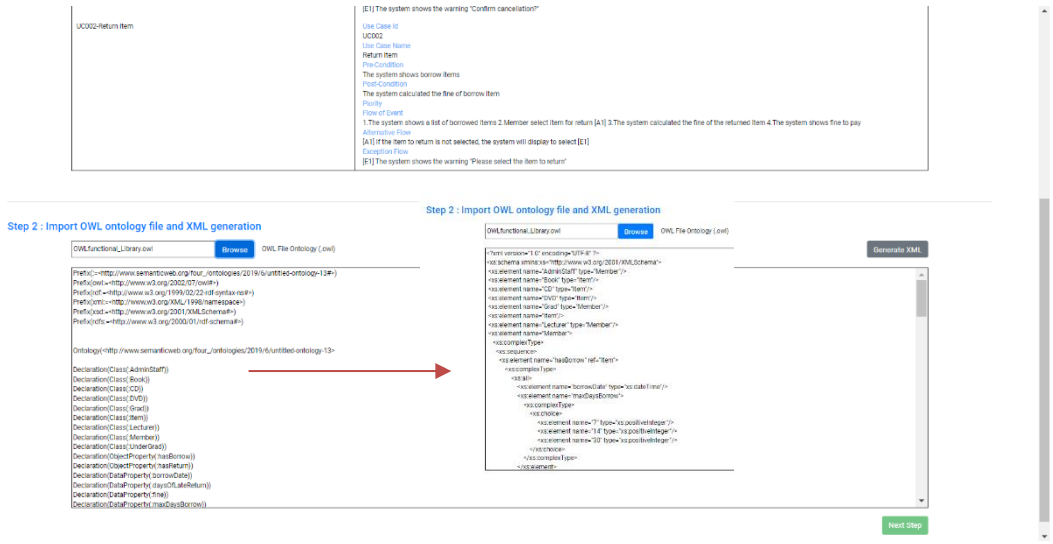
รูปที่ 5.3 หน้าจอการสร้างโปรเจคใหม่

3) หน้าจอนำเข้าไฟล์ของต้นแบบ หลังจากทำการสร้างโปรเจคใหม่เรียบร้อยแล้วจะปรากฏหน้าจอของการนำเข้าไฟล์จำนวน 2 ไฟล์ คือ ไฟล์ .xml ของยูสเคส และไฟล์ .owl ของออนโทโลยีความต้องการซอฟต์แวร์ ดังรูปที่ 5.4



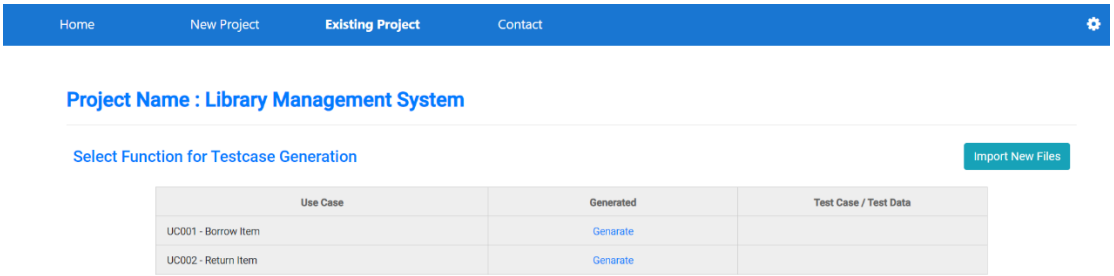
รูปที่ 5.4 หน้าจอการนำเข้าไฟล์ของต้นแบบ

จากรูปที่ 5.4 (Step: 1) ทำการเลือก (Browse) ไฟล์ .xml ของยูสเคสหลังจากเลือกแล้วต้นแบบจะแสดงรายละเอียดต่าง ๆ ของยูสเคส และ (Step: 2) เลือกไฟล์ .owl ซึ่งจะแสดงโครงสร้างของไฟล์ในกล่องข้อความ หลังจากนั้นให้ทำการกดปุ่ม “Generate XML” เพื่อทำการแปลงออนโทโลยีจาก OWL เป็น XML ก่อนนำไปใช้งาน หลังจากกดปุ่มต้นแบบจะแสดงโครงสร้าง XML ที่แปลงแล้วแทนที่โครงสร้าง OWL หลังจากนั้นทำการกดปุ่ม “Next” ที่มุมล่างขวาเพื่อไปยังขั้นตอนการจัดการตัวแปรและค่าของตัวแปรที่จะนำไปใช้ในการสร้างกรณีทดสอบ ดังรูปที่ 5.5



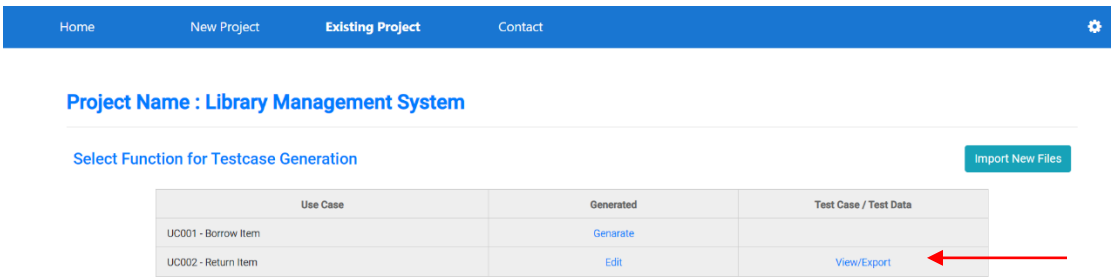
รูปที่ 5.5 แสดงโครงสร้างไฟล์หลังจากนำเข้า

4) หน้าจอการเลือกฟังก์ชันของระบบสำหรับสร้างกรณีทดสอบ หลังจากกดปุ่ม “Next Step” จะแสดงหน้าจอเพื่อให้ทำการเลือกฟังก์ชันที่ต้องการจัดการตัวแปรสำหรับการสร้างกรณีทดสอบ โดยฟังก์ชันทั้งหมดซึ่งสอดคล้องกับไฟล์ยูสเคสที่ได้ทำการนำเข้าในขั้นตอนก่อนหน้าจะถูกแสดงไว้ทั้งหมด ดังรูปที่ 5.6



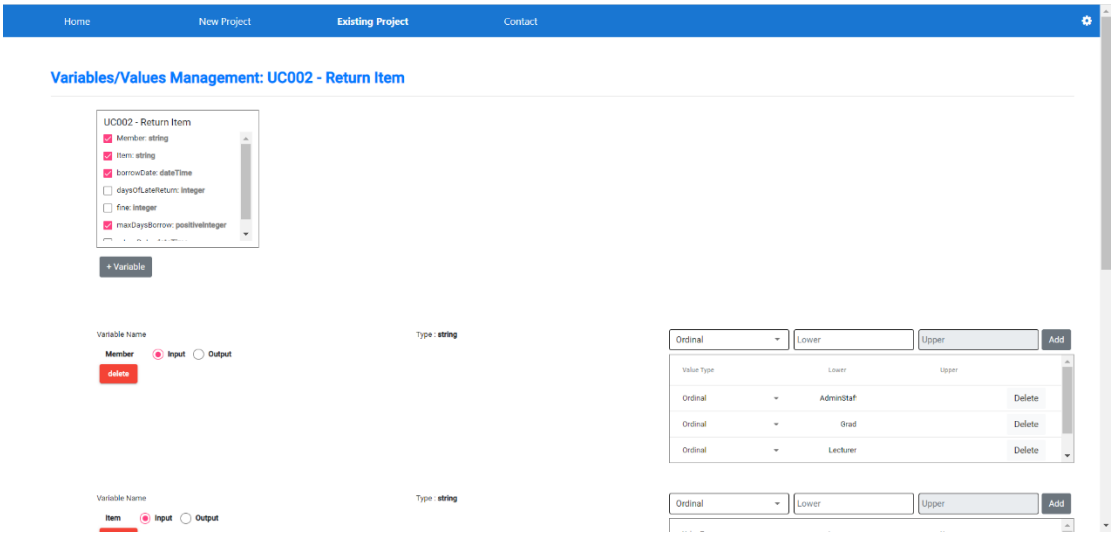
รูปที่ 5.6 หน้าจอเลือกฟังก์ชันระบบสำหรับสร้างกรณีทดสอบ

จากรูปที่ 5.6 ในคอลัมน์แรก (Use Case) จะแสดงรหัสและชื่อของยูสเคสซึ่งเป็นฟังก์ชันสำหรับสร้างกรณีทดสอบ และคอลัมน์ที่สอง (Generated) แสดงสถานะการทำงานของการสร้างกรณีทดสอบ โดยถ้าหากยังไม่ได้มีการสร้างกรณีทดสอบต้นแบบจะแสดง “Generate” เพื่อให้กดเข้าไปสร้างกรณีทดสอบได้ และเมื่อได้ทำการสร้างกรณีทดสอบแล้วต้นแบบจะเปลี่ยนเป็น “Edit” เพื่อให้สามารถกลับมาแก้ไขได้ และในคอลัมน์สุดท้าย (Test Case/ Test Data) ต้นแบบจะแสดง “View/Export” เมื่อมีการสร้างกรณีทดสอบแล้วเพื่อให้สามารถเข้าไปดูและนำออกไฟล์กรณีทดสอบไปใช้งานต่อได้ แต่ถ้ายังเป็น “Generate” คอลัมน์สุดท้ายจะปล่อยว่าง และถ้าหากไฟล์นำเข้ามีการเปลี่ยนแปลงสามารถกดปุ่ม “Import New Files” เพื่ออัพโหลดไฟล์ใหม่ได้ ดังรูปที่ 5.7



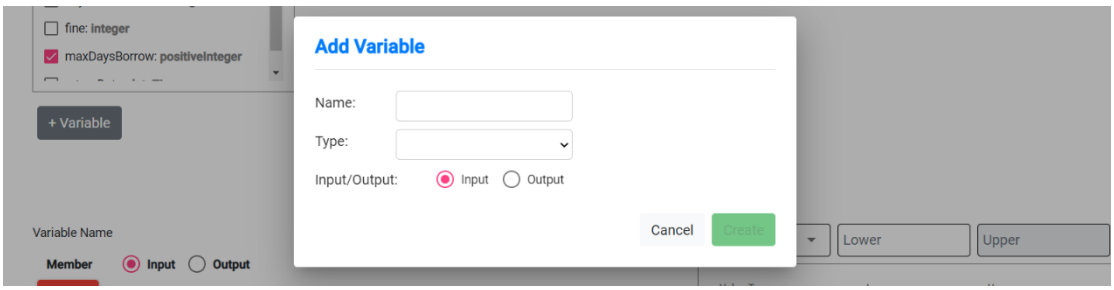
รูปที่ 5.7 แสดงหน้าจอเลือกฟังก์ชันเมื่อมีการสร้างกรณีทดสอบ

5) หน้าจอการจัดการตัวแปรและค่าของตัวแปร เมื่อกดเลือก “Generate” ของฟังก์ชันที่ต้องการสร้างกรณีทดสอบแล้วระบบจะแสดงหน้าจอสำหรับจัดการตัวแปรและค่าของตัวแปรที่เกี่ยวข้องภายในฟังก์ชัน โดยต้นแบบจะทำการดึงตัวแปรจากไฟล์ของขั้นตอนก่อนหน้าซึ่งเป็นพจนานุกรมข้อมูลสำหรับสร้างกรณีทดสอบให้อัตโนมัติ โดยสามารถทำการเลือกตัวแปรที่เกี่ยวข้องด้วยการติ๊กเครื่องหมายถูกหน้า checkbox ของตัวแปรที่ต้องการสร้างกรณีทดสอบ ดังรูปที่ 5.8



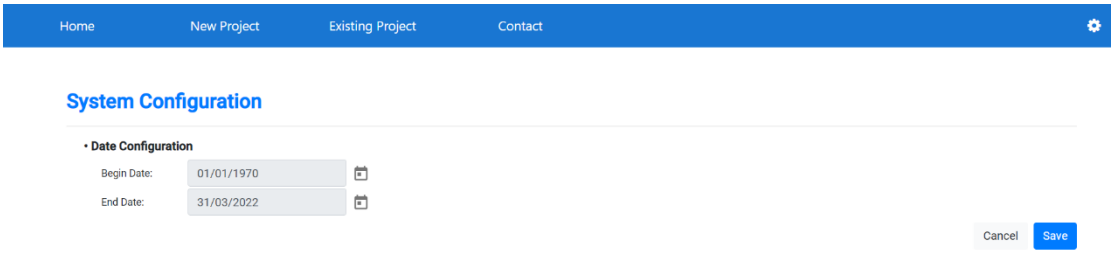
รูปที่ 5.8 หน้าจอการจัดการตัวแปรของต้นแบบ

จากรูปที่ 5.8 หน้าจอการจัดการตัวแปรสามารถเพิ่ม/ลบ/แก้ไขค่าของตัวแปรได้ และสามารถกำหนดรูปแบบของตัวแปรได้ว่าเป็นตัวแปรประเภท input หรือ output โดยการติ๊กเลือกในช่องวงกลมหลังชื่อตัวแปร และในตารางด้านขวาถัดจากชื่อตัวแปรจะเป็นตารางในการจัดการค่าของตัวแปรโดยจะพิจารณาเฉพาะค่าของตัวแปรที่เป็น valid ส่วนค่า invalid ต้นแบบจะทำการสร้างให้โดยอัตโนมัติ และถ้าหากต้องการเพิ่มตัวแปรใหม่สามารถกดปุ่ม “+Variable” เพื่อเพิ่มตัวแปรได้ โดยป๊อปอัพการเพิ่มตัวแปร ดังรูปที่ 5.9



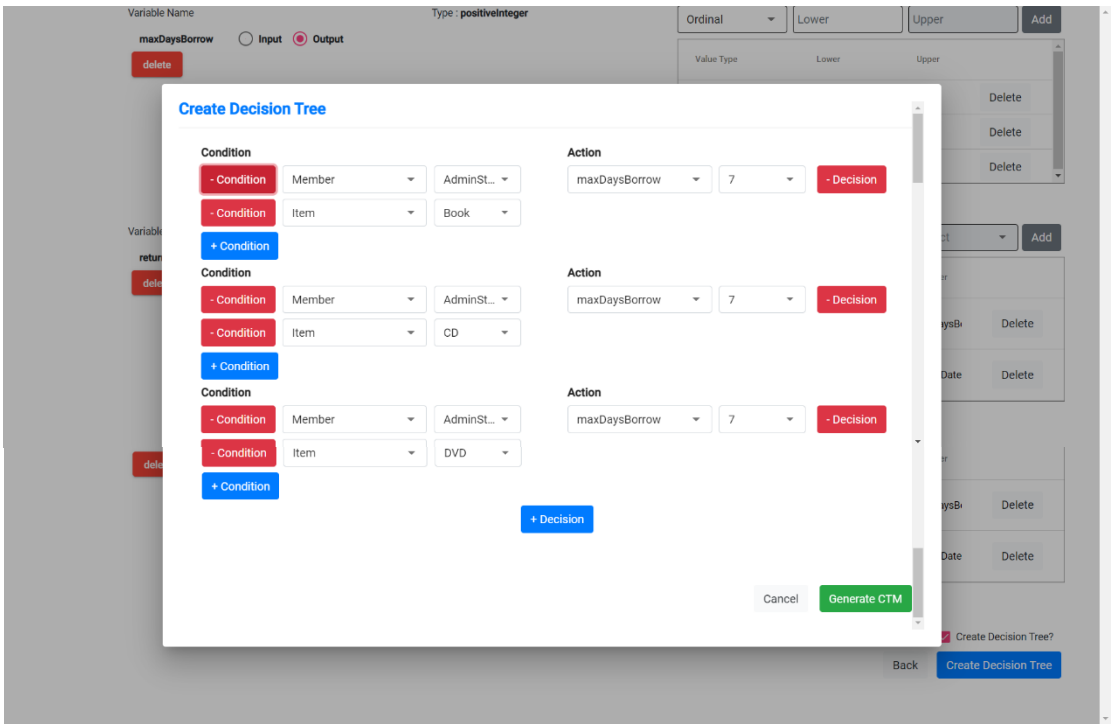
รูปที่ 5.9 ป็อปอัพการเพิ่มตัวแปร

และถ้าหากระบบที่จะสร้างกรณีทดสอบมีการกำหนดช่วงเริ่มต้นและสิ้นสุดของช่วงเวลาสามารถกดปุ่มเฟืองมุมขวามบนของหน้าจอเพื่อทำการตั้งค่าช่วงของวันที่ที่ต้องการในการสร้างกรณีทดสอบได้ ดังรูปที่ 5.10



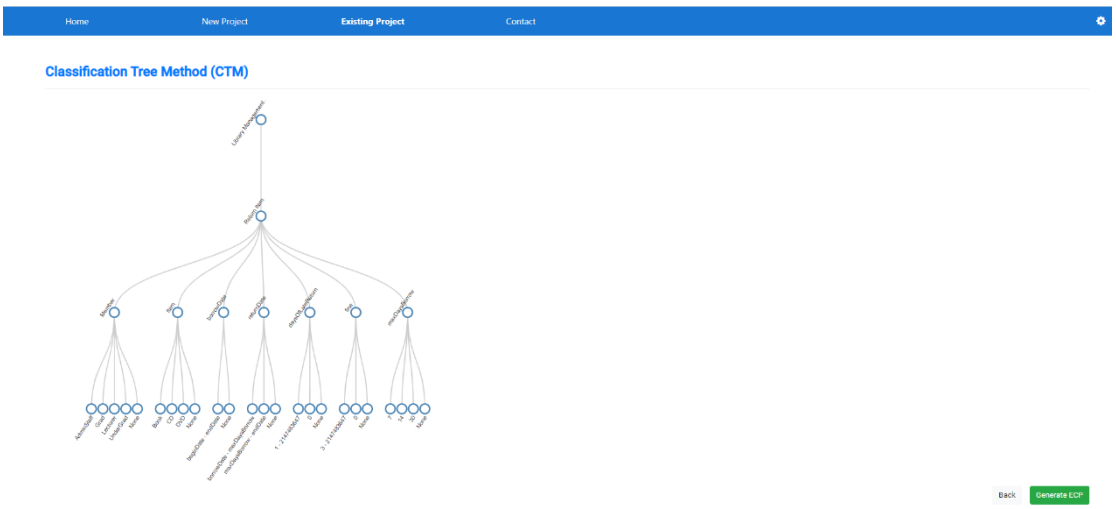
รูปที่ 5.10 หน้าจอการตั้งค่าระยะเวลา

6) หน้าจอการสร้างเงื่อนไขการตัดสินใจ ถ้าหากในกรณีที่ระบบมีเงื่อนไขการตัดสินใจสามารออกแบบต้นไม้การตัดสินใจได้ โดยหลังจากจัดการตัวแปรที่ต้องการทั้งหมดแล้ว มุมล่างขวาของหน้าจอจะมีช่องสี่เหลี่ยมและปุ่ม “Generate CTM” ซึ่งถ้าระบบไม่มีเงื่อนไขการตัดสินใจสามารถกดปุ่ม “Generate CTM” เพื่อเข้าสู่กระบวนการสร้างกรณีทดสอบได้ แต่ถ้ามีเงื่อนไขการตัดสินใจให้ทำการติ๊กถูกในกล่องข้อความแล้วปุ่มจะเปลี่ยนเป็น “Create Decision Tree” เพื่อให้สามารถออกแบบต้นไม้การตัดสินใจได้ โดยจะทำการเลือกตัวแปรที่เป็นเงื่อนไข (Condition) ซึ่งเป็นตัวแปร input และ (Action) ที่เป็นผลของการตัดสินใจซึ่งเป็นตัวแปร output และเลือกค่าตัวแปรของแต่ละเงื่อนไข เมื่อออกแบบต้นไม้การตัดสินใจครอบคลุมแล้วให้ทำการกดปุ่ม “Generate CTM” ภายในหน้าจอเพื่อเข้าสู่กระบวนการสร้างกรณีทดสอบ ดังรูปที่ 5.11



รูปที่ 5.11 หน้าจอการออกแบบต้นไม้การตัดสินใจ

7) หน้าจอการสร้างต้นไม้การจำแนก (CTM) หลังจากจัดการตัวแปรและค่าของตัวแปรรวมถึงเงื่อนไขการตัดสินใจเรียบร้อยแล้ว เมื่อทำการกดปุ่ม “Generate CTM” ในหน้าจอก่อนหน้า ระบบจะทำการสร้างต้นไม้การจำแนกโดยจะแสดงเป็นแผนภาพให้เห็นรายละเอียดแต่ละกิ่งก้านของระบบ โดยจะพิจารณาในส่วนของเทอมินัลซึ่งจะเป็นค่าของตัวแปรในการสร้างช่วงของชั้นสมมูลสำหรับการสร้างกรณีทดสอบ ดังรูปที่ 5.12

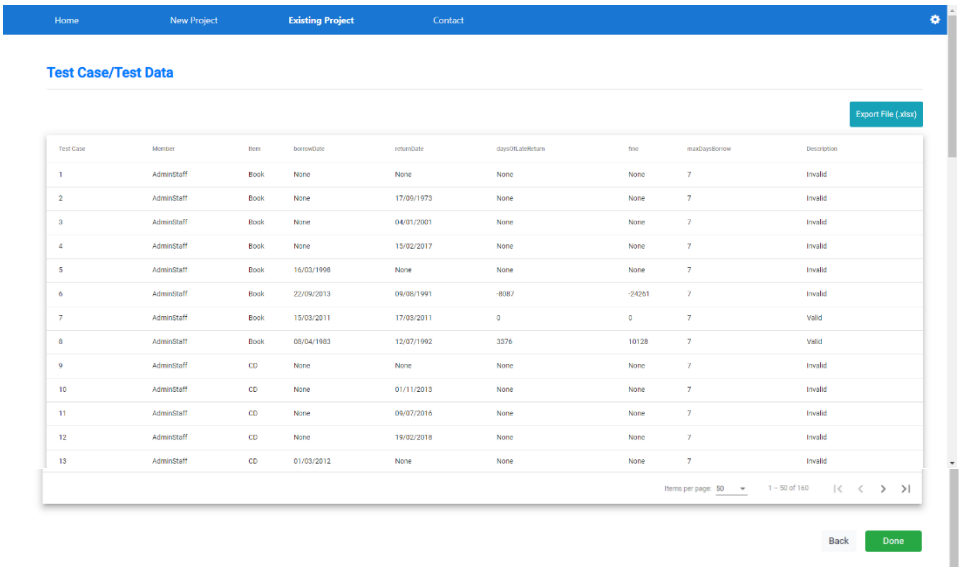


รูปที่ 5.12 หน้าจอการสร้างต้นไม้การจำแนก

8) หน้าจอการแบ่งชั้นสมมูล (ECP) หลังจากทำการกดปุ่ม “Generate ECP” ในหน้าจอก่อนหน้าระบบจะทำการสร้างชั้นสมมูลของตัวแปรแต่ละตัวโดยชั้นสมมูลที่ถูกต้อง (valid) จะสร้างเป็นสีดำ ส่วนชั้นสมมูลที่ไม่ถูกต้อง (invalid) จะสร้างเป็นสีแดง ดังรูปที่ 5.13

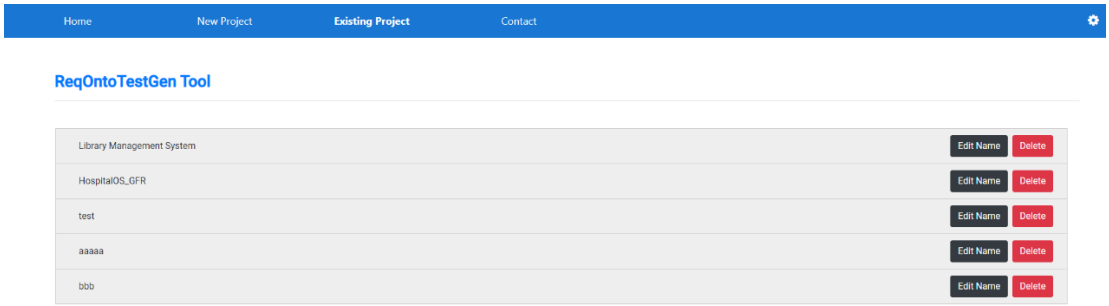
รูปที่ 5.13 หน้าจอการแบ่งชั้นสมมูล

9) หน้าจอแสดงกรณีทดสอบที่ได้จากต้นแบบ โดยเมื่อทำการกดปุ่ม “Generate Test Case” ในหน้าจอก่อนหน้า ระบบจะทำการสร้างกรณีทดสอบโดยใช้ผลคูณคาร์ทีเซียนของชั้นสมมูลในทุก ๆ ชั้น โดยค่าของตัวแปรที่เป็นช่วงในแต่ละช่วง ต้นแบบจะทำการสุ่มค่ามาให้ ซึ่งจะสร้างกรณีทดสอบทั้งที่ถูกต้องและไม่ถูกต้องเพื่อให้ครอบคลุมที่สุด โดยกรณีทดสอบและข้อมูลทดสอบที่ได้จะถูกแสดงอยู่ในรูปแบบของตาราง และสามารถส่งออกไฟล์ในรูปแบบไฟล์ excel เพื่อนำไปใช้ในการทดสอบต่อไปได้ และเมื่อกดปุ่ม “Done” จะเสร็จสิ้นกระบวนการสร้างกรณีทดสอบและจะกลับไปยังหน้าจอการเลือกฟังก์ชันเพื่อสร้างกรณีทดสอบฟังก์ชันอื่นที่ต้องการได้ ดังรูปที่ 5.14



รูปที่ 5.14 ตัวอย่างหน้าจอกรณีทดสอบที่ได้จากต้นแบบ

10) หน้าจอ Existing Project เมื่อทำการสร้างกรณีทดสอบของระบบเรียบร้อยแล้ว ระบบที่ถูกสร้างมาทั้งหมดจะถูกเก็บไว้ในหน้าจอนี้เพื่อให้สามารถกลับมาปรับปรุงแก้ไขได้ โดยสามารถแก้ไข “Edit Name” ชื่อโปรเจคได้และสามารถลบ “Delete” โปรเจคที่ไม่ต้องการแล้วออกไปได้ ดังรูปที่ 5.15



รูปที่ 5.15 หน้าจอสำหรับจัดเก็บโปรเจคที่เคยสร้างกรณีทดสอบ

บทที่ 6

กระบวนการทดสอบและประเมินผลต้นแบบ

ในกระบวนการทดสอบและประเมินผลต้นแบบมีวัตถุประสงค์เพื่อยืนยันความถูกต้องและประสิทธิภาพในการทำงานของต้นแบบที่พัฒนาขึ้นว่ามีความสอดคล้องกับกรอบแนวคิดของงานวิจัยที่ได้นำเสนอไว้หรือไม่ โดยผู้วิจัยได้ทำการประเมินผลจากกรณีศึกษาทั้ง 2 กรณีศึกษาซึ่งได้นำเสนอไว้ในบทที่ 4 โดยทำการเปรียบเทียบผลลัพธ์ในการสร้างกรณีทดสอบที่ได้ดำเนินการด้วยวิธีการมือและดำเนินการจากต้นแบบ รวมถึงการประเมินผลต้นแบบสำหรับการนำไปใช้งานในอนาคตโดยผู้เชี่ยวชาญทางด้านวิศวกรรมซอฟต์แวร์ที่เกี่ยวข้อง (คำถามงานวิจัย ข้อ 4 และสอดคล้องกับวัตถุประสงค์งานวิจัย ข้อ 4) ซึ่งรายละเอียดและเนื้อหาของบทนี้ประกอบด้วย 1) สภาพแวดล้อมที่ใช้ในการทดสอบและประเมินผลต้นแบบ 2) การทดสอบต้นแบบ 3) การประเมินผลต้นแบบ และ 4) สรุปการประเมินผลต้นแบบ

6.1 สภาพแวดล้อมที่ใช้ในการทดสอบและประเมินผลต้นแบบ

ในการทดสอบและประเมินผลต้นแบบจะดำเนินการบนสภาพแวดล้อมซึ่งแบ่งออกเป็น 2 ส่วนด้วยกัน ได้แก่ 1) ฮาร์ดแวร์ (Hardware) เป็นอุปกรณ์ที่นำมาใช้ทดสอบและประเมินผลต้นแบบ 2) ซอฟต์แวร์ (Software) ประกอบไปด้วยระบบปฏิบัติการและโปรแกรมต่าง ๆ ในการดำเนินการ โดยมีรายละเอียดในแต่ละส่วน ดังต่อไปนี้

6.1.1 ฮาร์ดแวร์ (Hardware)

1) คอมพิวเตอร์โน้ตบุ๊ก
   ก. หน่วยประมวลผล Intel Corei7 8th Gen 2.2 GHz
   ข. หน่วยความจำหลัก 16 GB
   ค. ฮาร์ดดิสก์ 1TB+SSD 128GB

6.1.2 ซอฟต์แวร์ (Software)

1) ระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 10 (Microsoft Windows 10)
2) โปรแกรมวิชวล สตูดิโอ โค้ด เวอร์ชัน 1.65.2 (Visual Studio Code 1.65.2)
3) โปรแกรมโหนด เจเอส เวอร์ชัน 16.14.0 (Node.js 16.14.0)
4) เครื่องมือแองกูลาร์ ซี แอล ไอ เวอร์ชัน 11.2.4 (Angular CLI 11.2.4)
5) โปรแกรมเว็บเบราว์เซอร์ กูเกิล โครม เวอร์ชัน 99.0.4844.82 (Google Chrome 99.0.4844.82)

6.2 การทดสอบต้นแบบ

ในการทดสอบต้นแบบผู้วิจัยได้ทำการทดสอบฟังก์ชันการทำงานต่าง ๆ ตามที่ได้นำเสนอการทำงานไว้ในบทที่ 5 ซึ่งผู้วิจัยได้พิจารณาทดสอบฟังก์ชันการทำงานของต้นแบบหลัก ๆ 3 ฟังก์ชันด้วยกัน ได้แก่ 1) ฟังก์ชันการแปลงออนโทโลยีจาก OWL เป็น XML 2) ฟังก์ชันการจัดการตัวแปรและต้นไม้การตัดสินใจ 3) ฟังก์ชันการสร้างกรณีทดสอบ สามารถแสดงรายละเอียดของภาพรวมระดับบนสุดที่เป็นบทภาพเหตุการณ์การทดสอบในแต่ละฟังก์ชัน ดังตารางที่ 6.1

ตารางที่ 6.1 บทภาพเหตุการณ์การทดสอบฟังก์ชันของต้นแบบ

TS-ID	TS description	Expected Result
TS-01	ทดสอบฟังก์ชันการแปลงออนโทโลยีจาก OWL เป็น XML	ต้นแบบสามารถแปลงออนโทโลยีจาก OWL เป็น XML ได้สอดคล้องกับกฎทั้ง 13 กฎที่ได้ออกแบบไว้ รายละเอียดในหัวข้อ 6.2.1 ทดสอบฟังก์ชันการแปลงออนโทโลยีจาก OWL เป็น XML
TS-02	ทดสอบฟังก์ชันการจัดการตัวแปรและต้นไม้การตัดสินใจ	ต้นแบบสามารถจัดการตัวแปรและต้นไม้การตัดสินใจที่ใช้สำหรับการสร้างกรณีทดสอบ รายละเอียดในหัวข้อ 6.2.2 ทดสอบฟังก์ชันการจัดการตัวแปรและต้นไม้การตัดสินใจ
TS-03	ทดสอบฟังก์ชันการการสร้างกรณีทดสอบ	ต้นแบบสามารถสร้างกรณีทดสอบได้ครอบคลุมและสอดคล้องกับเงื่อนไขการตัดสินใจที่ได้ออกแบบไว้ รายละเอียดในหัวข้อ 6.2.3 ทดสอบฟังก์ชันการสร้างกรณีทดสอบ

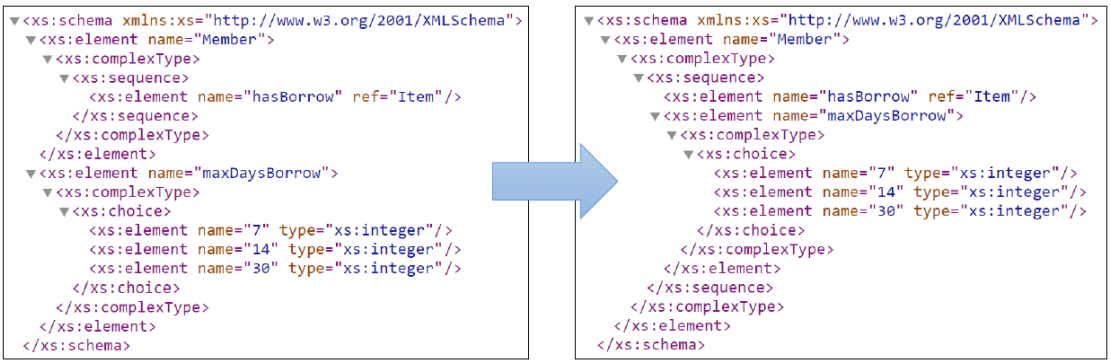
6.2.1 ทดสอบฟังก์ชันการแปลงออนโทโลยีจาก OWL เป็น XML

การทดสอบฟังก์ชันการแปลงออนโทโลยีจาก OWL เป็น XML ตามกฎที่ได้ออกแบบไว้ทั้งหมด 13 กฎ เพื่อทดสอบว่าต้นแบบที่พัฒนาสามารถแปลงกฎได้สอดคล้องตามที่ได้ออกแบบไว้หรือไม่ โดยผู้วิจัยได้ทำการออกแบบออนโทโลยีทั้งหมด 4 ชุดที่สอดคล้องกับกฎทั้ง 13 กฎ เพื่อให้ครอบคลุมการแปลงของทุกกฎตามที่ได้ออกแบบ สามารถดูตัวอย่างการแปลงของกฎทั้งหมดได้ในภาคผนวก ก. โดยในการทดสอบสามารถแสดงตารางการทดสอบ ดังตารางที่ 6.2

ตารางที่ 6.2 การทดสอบฟังก์ชันการแปลงออนโทโลยีจาก OWL เป็น XML

TC-ID	Objective	Pass/Fail	Revision
The structure of Classes and Relations			
TC-01	ตรวจสอบกฎการแปลง “Class”	**Pass**	-
TC-02	ตรวจสอบกฎการแปลง “Sub Class”	**Pass**	-
TC-03	ตรวจสอบกฎการแปลง “Object Property”	**Pass**	-
TC-04	ตรวจสอบกฎการแปลง “Data Property”	**Pass**	-
Object Property Restrictions			
TC-05	ตรวจสอบกฎการแปลง “Existential”	**Pass**	-
TC-06	ตรวจสอบกฎการแปลง “Universal”	**Pass**	-
TC-07	ตรวจสอบกฎการแปลง “Minimum cardinality”	**Pass**	-
TC-08	ตรวจสอบกฎการแปลง “Maximum cardinality”	**Pass**	-
Data Property Restrictions			
TC-09	ตรวจสอบกฎการแปลง “Existential”	**Pass**	-
TC-010	ตรวจสอบกฎการแปลง “Universal”	**Pass**	-
TC-011	ตรวจสอบกฎการแปลง “Individual value”	Fail	**Pass**
TC-012	ตรวจสอบกฎการแปลง “Minimum cardinality”	**Pass**	-
TC-013	ตรวจสอบกฎการแปลง “Maximum cardinality”	**Pass**	-

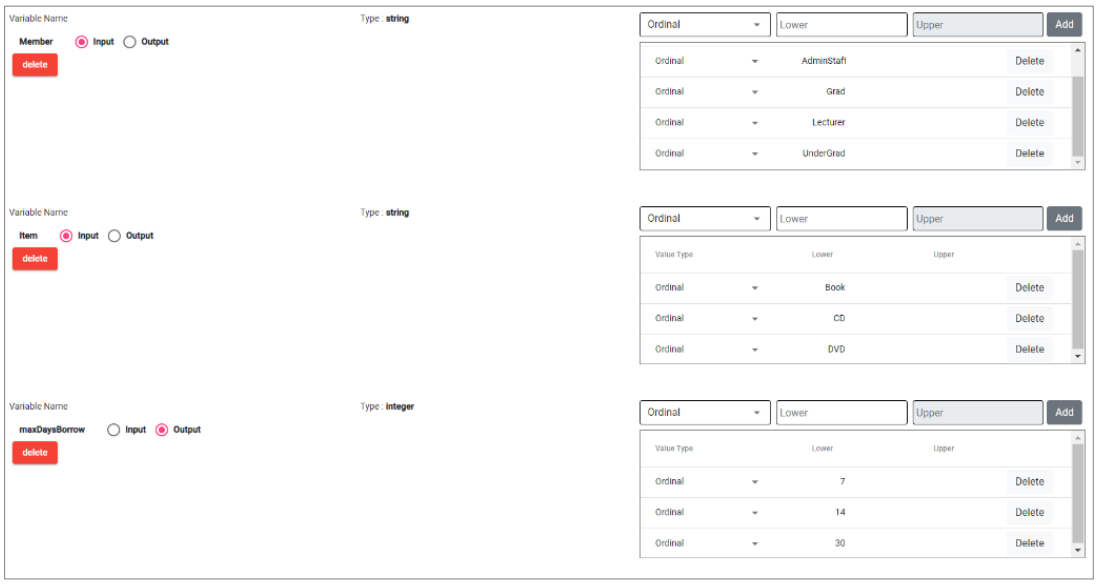
จากตารางที่ 6.2 ในการทดสอบฟังก์ชันการแปลงออนโทโลยีจาก OWL เป็น XML ผู้วิจัยได้ออกแบบไฟล์ .owl ที่สอดคล้องกับกฎทั้ง 13 กฎ เพื่อทดสอบความถูกต้องของต้นแบบในการแปลงตามกฎที่ได้ออกแบบ พบว่าในทุก ๆ กฎสามารถแปลงออกมาตรงกับที่ผู้วิจัยได้นำเสนอกฎการแปลงเอาไว้ มีเพียงกฎที่ 11 ในการทดสอบฟังก์ชันการแปลง ต้นแบบได้สร้างอิลีเมนต์แยกออกมาอยู่ภายนอกอิลีเมนต์ของคลาส โดยกฎที่ 11 จะเป็นค่าข้อมูลของคุณสมบัติของคลาส ดังนั้นจึงต้องสร้างให้อยู่ภายใต้อิลีเมนต์ของคลาสที่เกี่ยวข้อง ซึ่งผู้วิจัยได้ทำการปรับโครงสร้างของการเกิดในอิลีเมนต์ให้ถูกต้องตามกฎที่ได้ออกแบบ ดังรูปที่ 6.1



รูปที่ 6.1 การปรับกฎการแปลงของกฎที่ 11

6.2.2 ทดสอบฟังก์ชันการจัดการตัวแปรและต้นไม้การตัดสินใจ

การทดสอบฟังก์ชันการจัดการตัวแปรและต้นไม้การตัดสินใจ โดยผู้วิจัยได้ทำการเพิ่มตัวแปรและค่าของตัวแปรเพื่อใช้ในการทดสอบฟังก์ชันการจัดการตัวแปรว่าสามารถนำตัวแปรที่จัดการเรียบร้อยแล้วไปใช้ในเทคนิคการสร้างกรณีทดสอบด้วยต้นไม้การจำแนกและการแบ่งชั้นสมมูลได้สอดคล้องกับที่ระบุไปหรือไม่ รวมถึงการออกแบบเงื่อนไขการตัดสินใจในกรณีที่ระบบมีเงื่อนไขการตัดสินใจ เพื่อให้การสร้างกรณีทดสอบมีความครอบคลุมตามเงื่อนไขของการทดสอบ โดยหน้าจอการจัดการตัวแปรและค่าของตัวแปรจะอิงจากระบบจัดการห้องสมุด โดยมีตัวแปร input จำนวน 2 ตัวแปร คือ ตัวแปร Member และ Item ส่วนตัวแปร maxDaysBorrow เป็นตัวแปร output ดังรูปที่ 6.2



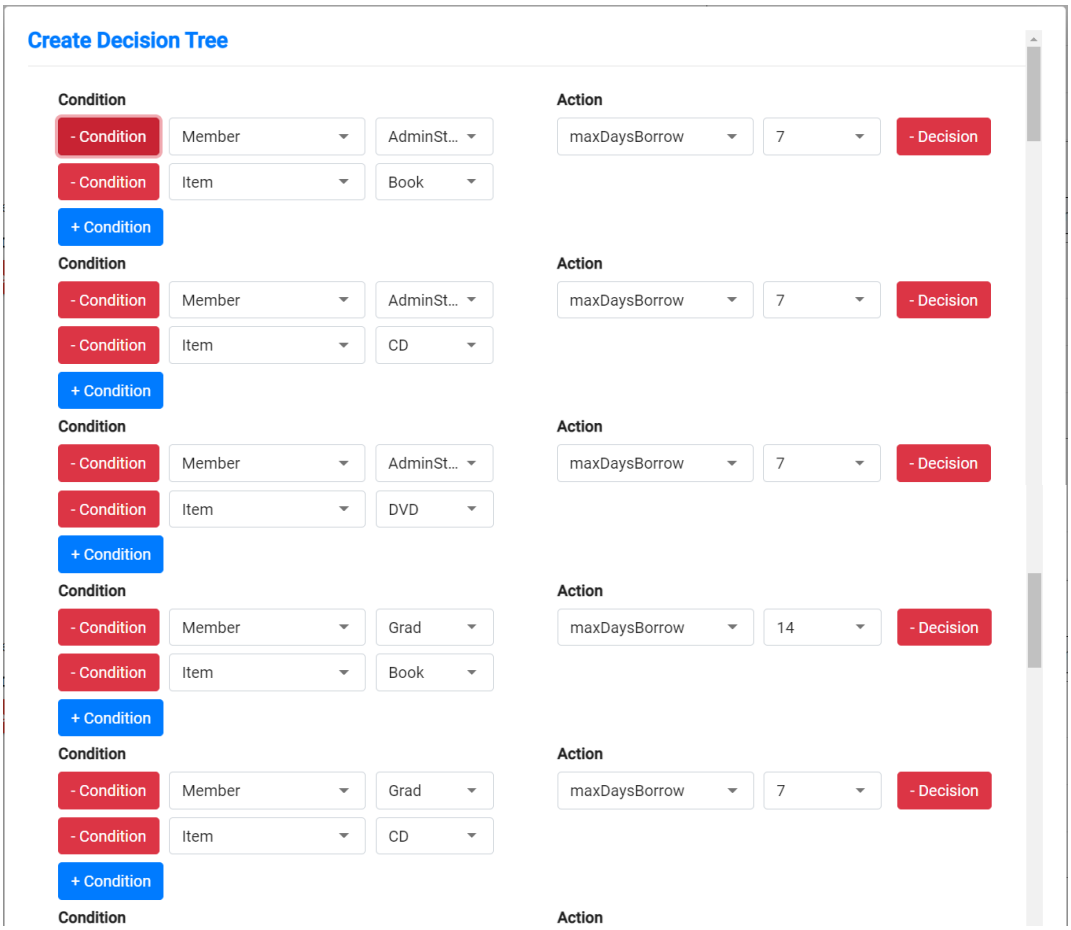
รูปที่ 6.2 การจัดการตัวแปรของต้นแบบ

จากรูปที่ 6.2 ตัวแปรที่นำมาใช้ในการทดสอบฟังก์ชันการจัดการตัวแปรของต้นแบบประกอบไปด้วยตัวแปรที่เกี่ยวข้องในการสร้างกรณีทดสอบจำนวน 3 ตัวแปร ดังแสดงรายละเอียดตัวแปรและค่าของตัวแปร ดังตารางที่ 6.3

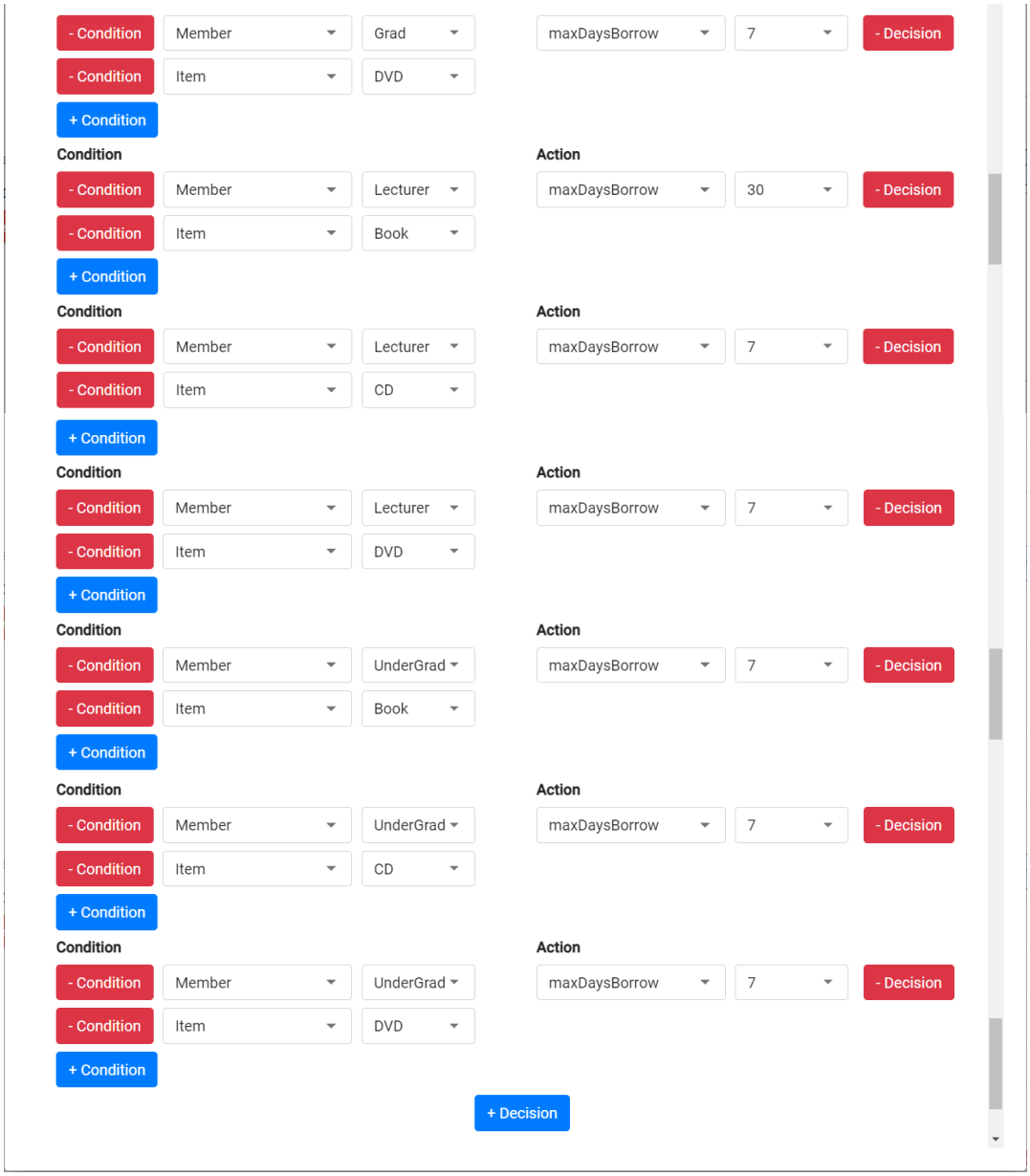
ตารางที่ 6.3 ตัวอย่างตัวแปรระบบจัดการห้องสมุด

ชื่อตัวแปร	ชนิดตัวแปร	ความหมายตัวแปร	ช่วงข้อมูลที่เป็นไปได้
Member	string	สมาชิกห้องสมุด	AdminStaff, Grad, Lecturer, UnderGrad
Item	string	ทรัพยากรห้องสมุด	Book, CD, DVD
maxDaysBorrow	integer	จำนวนวันสูงสุดที่สามารถยืมได้	7, 14, 30

จากตารางที่ 6.3 สามารถนำตัวแปรทั้ง 3 ตัวมาออกแบบเงื่อนไขการตัดสินใจโดยอ้างอิงจากเงื่อนไขการยืมทรัพยากรห้องสมุดของสมาชิกแต่ละประเภทว่าสามารถยืมได้สูงสุดจำนวนกี่วัน โดยมีเงื่อนไขการตัดสินใจทั้งหมด 12 เงื่อนไข ดังรูปที่ 6.3



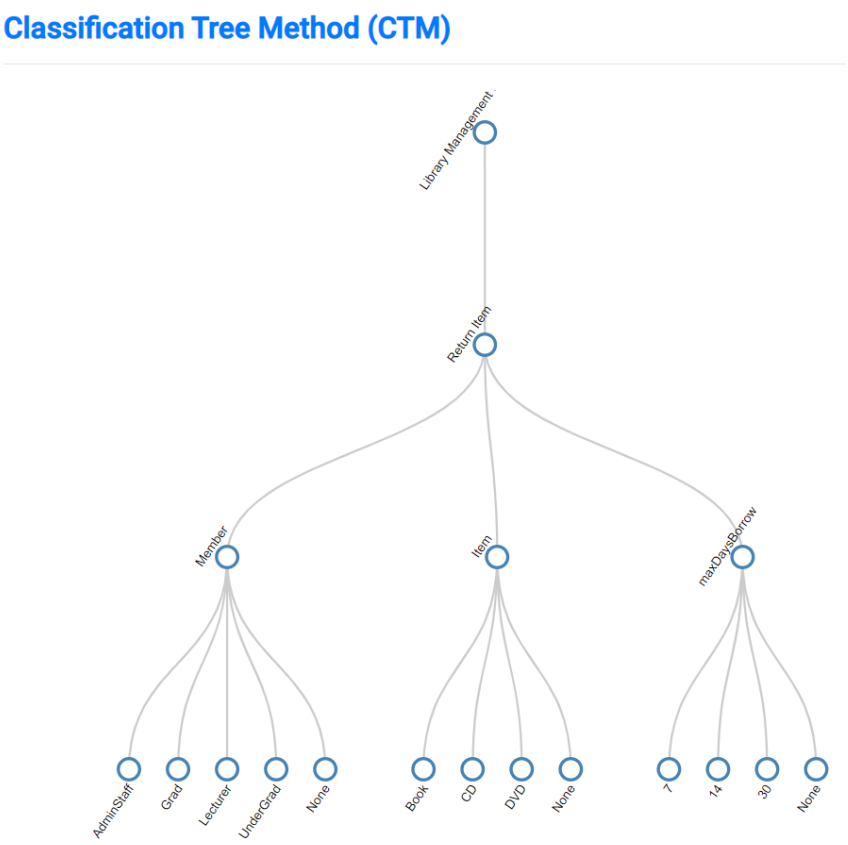
รูปที่ 6.3 การออกแบบต้นไม้การตัดสินใจของต้นแบบ



รูปที่ 6.3 การออกแบบต้นไม้การตัดสินใจของต้นแบบ (ต่อ)

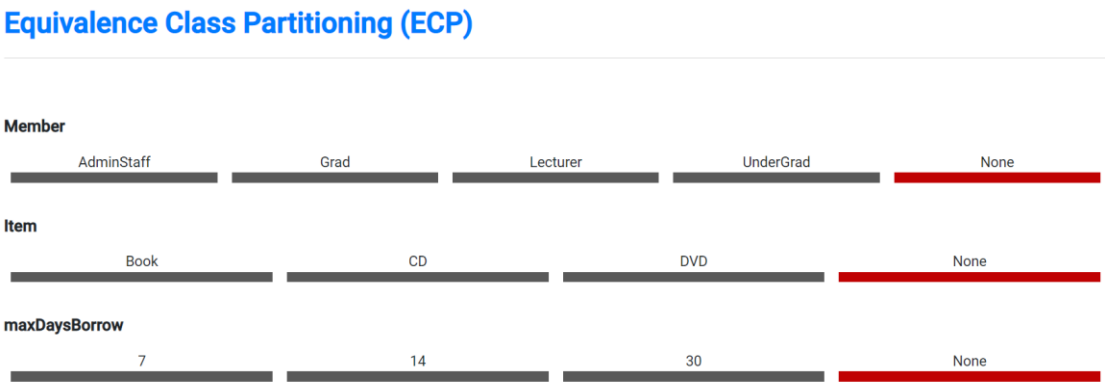
6.2.3 ทดสอบฟังก์ชันการสร้างกรณีทดสอบ

การทดสอบฟังก์ชันการสร้างกรณีทดสอบ ผู้วิจัยได้มีการพิจารณาเทคนิคการทดสอบแบบกล่องดำจำนวน 2 เทคนิคมาใช้ในการสร้างกรณีทดสอบ โดยจะนำตัวแปรที่ได้จากฟังก์ชันการจัดการตัวแปรและต้นไม้การตัดสินใจมาทำการจำแนกด้วยเทคนิคต้นไม้การจำแนกเป็นเทคนิคแรก สามารถแสดงต้นไม้การจำแนกจากตัวแปรข้างต้น ดังรูปที่ 6.4



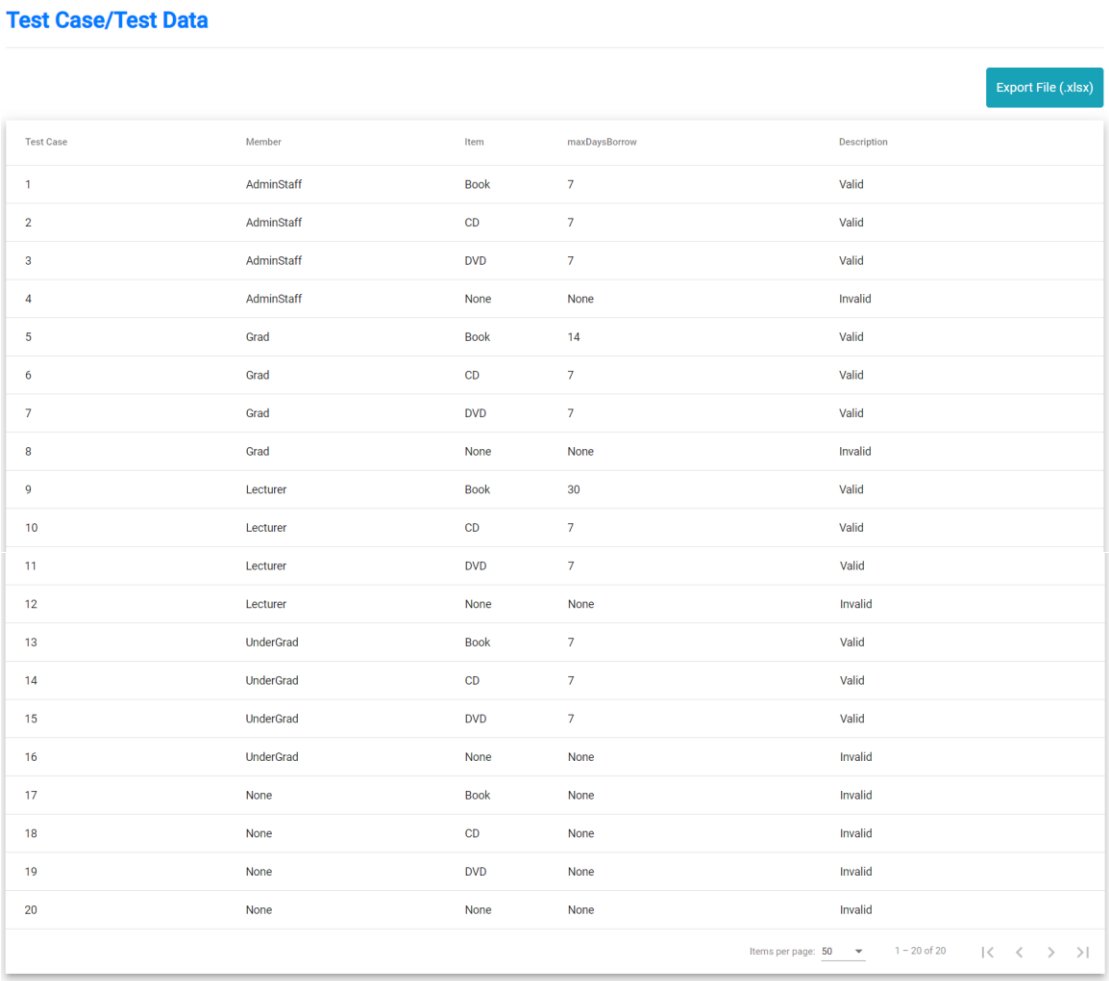
รูปที่ 6.4 ต้นไม้การจำแนกของต้นแบบ

จากรูปที่ 6.4 สามารถแสดงต้นไม้การจำแนกของตัวแปรซึ่งจะเห็นได้ว่าตัวแปรและค่าของตัวแปรทั้งหมดสามารถนำมาจำแนกโดยต้นไม้การจำแนก รวมถึงการพิจารณาค่าข้อมูลที่ไม่ถูกต้องในแต่ละตัวแปรเพื่อเพิ่มความครอบคลุมในการสร้างกรณีทดสอบ ต้นแบบสามารถสร้างค่าข้อมูลเหล่านี้มาให้โดยอัตโนมัติ ซึ่งค่าที่ไม่ถูกต้องของทั้ง 3 ตัวแปร คือ “None” หลังจากนั้นจะนำค่าของตัวแปรในโหนดสุดท้ายของต้นไม้การจำแนกมาทำการแบ่งชั้นสมมูลในเทคนิคที่ 2 คือ เทคนิคการแบ่งชั้นสมมูลสำหรับการสร้างกรณีทดสอบ ดังรูปที่ 6.5



รูปที่ 6.5 การแบ่งชั้นสมมูลของต้นแบบ

จากรูปที่ 6.5 ประกอบไปด้วยชั้นสมมูลที่เกี่ยวข้องในการสร้างกรณีทดสอบ โดยชั้นสมมูลที่ถูกต้องจะสร้างเป็นชั้นสีดำ และชั้นสมมูลที่ไม่ถูกต้องจะสร้างเป็นชั้นสีแดง สำหรับการสร้างกรณีทดสอบจะนำชั้นสมมูลทั้งหมดของตัวแปร input มาคูณกันโดยใช้ผลคูณคาร์ทีเซียนของทุก ๆ ชั้นสมมูลทั้งชั้นสมมูลที่ถูกต้องและไม่ถูกต้องเพื่อให้ครอบคลุมการทดสอบ จากชั้นสมมูลของทั้ง 3 ตัวแปรสามารถสร้างกรณีทดสอบได้ทั้งหมด 20 (5*4) กรณีทดสอบ โดยผลลัพธ์ของกรณีทดสอบที่ได้จะมีความครอบคลุมตามเงื่อนไขของการทดสอบ และสอดคล้องกับกรณีทดสอบที่ผู้วิจัยทดลองสร้างด้วยวิธีการมือ สามารถแสดงกรณีทดสอบที่ได้จากต้นแบบ ดังรูปที่ 6.6



รูปที่ 6.6 กรณีทดสอบของต้นแบบ

6.3 การประเมินผลต้นแบบ

เพื่อเป็นการยืนยันความถูกต้องของต้นแบบที่พัฒนาสำหรับการสร้างกรณีทดสอบด้วยออนโทโลยีความต้องการซอฟต์แวร์ตามกรอบแนวคิดที่ได้นำเสนอไว้นั้น ผู้วิจัยได้มีการประเมินผล

โดยแบ่งออกเป็น 2 ส่วนหลัก ๆ ได้แก่ 1) การประเมินผลความถูกต้องของต้นแบบโดยใช้กรณีศึกษาทั้ง 2 กรณีศึกษา และ 2) การประเมินผลความพึงพอใจของต้นแบบโดยผู้เชี่ยวชาญในการนำไปใช้งาน สำหรับรายละเอียดของการประเมินในแต่ละส่วนมี ดังนี้

6.3.1 การประเมินผลต้นแบบโดยกรณีศึกษาทั้ง 2 กรณีศึกษา

การประเมินผลต้นแบบโดยใช้กรณีศึกษาทั้ง 2 กรณีศึกษาที่ได้นำเสนอในบทที่ 4 กรณีศึกษาของต้นแบบ โดยผู้วิจัยได้ทำการเปรียบเทียบระหว่างการสร้างกรณีทดสอบด้วยวิธีการมือกับกรณีทดสอบที่ได้จากต้นแบบว่ามีความสอดคล้องตามที่ได้ออกแบบไว้หรือไม่ โดยกรณีศึกษาที่ 1 คือ ระบบจัดการห้องสมุด และกรณีศึกษาที่ 2 คือ ระบบวิเคราะห์คำนวณค่าไต แสดงรายละเอียดของการประเมินในแต่ละกรณีศึกษา ดังตารางที่ 6.4

ตารางที่ 6.4 การประเมินผลต้นแบบโดยกรณีศึกษาทั้ง 2 กรณีศึกษา

การสร้างกรณีทดสอบ	**ด้วยวิธีการมือ**	**ด้วยต้นแบบ**
	จำนวนกรณีทดสอบ	จำนวนกรณีทดสอบ
กรณีศึกษาที่ 1: ระบบจัดการห้องสมุด	300	160
กรณีศึกษาที่ 2: ระบบวิเคราะห์คำนวณค่าไต	144	144

จากการประเมินต้นแบบโดยใช้กรณีศึกษาทั้ง 2 กรณีศึกษา สามารถอธิบายรายละเอียดของการประเมินผลต้นแบบในการสร้างกรณีทดสอบด้วยวิธีการมือเปรียบเทียบกับผลลัพธ์ที่ได้จากต้นแบบว่ามีความสอดคล้องและได้จำนวนกรณีทดสอบเท่ากันหรือไม่ โดยรายละเอียดการประเมินผลของทั้ง 2 กรณีศึกษาสามารถอธิบายผลการประเมินได้ ดังนี้

6.3.1.1 กรณีศึกษาที่ 1 ระบบจัดการห้องสมุด

จากตารางการประเมินผลต้นแบบโดยกรณีศึกษาทั้ง 2 กรณีศึกษา (ตารางที่ 6.4) แสดงให้เห็นว่าการเปรียบเทียบการสร้างกรณีทดสอบด้วยวิธีการมือกับผลลัพธ์ที่ได้จากต้นแบบมีความแตกต่างกัน เนื่องจากการสร้างกรณีทดสอบด้วยต้นแบบมีค่าข้อมูลของตัวแปรเปลี่ยนแปลงไปจำนวน 2 ค่า ทำให้กรณีทดสอบที่ได้มีจำนวนลดลง โดยตัวแปรที่มีค่าข้อมูลเปลี่ยนแปลงไป คือ ตัวแปร *borrowDate* และ *returnDate* ซึ่งเป็นตัวแปรชนิด dateTime ที่ได้มีการตัดค่าข้อมูลที่เป็นช่วง

“Out of Range” ของทั้ง 2 ตัวแปรนี้ตามที่ได้ออกแบบด้วยวิธีการมือในชั้นสมมูลที่ 12 และ 17 ออกไป เนื่องจากชั้นสมมูลนี้เป็นช่วงของวันที่ซึ่งอยู่นอกเหนือจากช่วงที่เป็นไปได้ โดยในชั้นสมมูลที่ 12 ของตัวแปร *borrowDate* เป็นช่วงของ “Out of Range” ซึ่งเป็นช่วงที่ไม่มีความเป็นไปได้ที่จะเกิดขึ้น เนื่องจากการกำหนดวันยืมสามารถเป็นได้ทุกช่วงตั้งแต่วันที่เริ่มต้น (Begin of Date) จนถึงวันที่สิ้นสุด (End of Date) เช่นเดียวกันกับชั้นสมมูลที่ 17 ของตัวแปร *returnDate* ในช่วงของ “Out of Range” จะครอบคลุมชั้นสมมูลที่ 13 ซึ่งเป็นช่วงที่เป็นไปไม่ได้ เนื่องจากถ้ายังไม่มีวันที่ยืมเกิดขึ้นจะไม่สามารถทำการคืนทรัพยากรที่ยืมมาได้ จึงกลายเป็นช่วงของชั้นสมมูลที่อยู่นอกเหนือจากช่วงที่เป็นไปได้ นอกเหนือจากนั้นได้มีการกำหนดค่า “None” ในชั้นสมมูลของทั้ง 2 ตัวแปร ซึ่งสามารถครอบคลุมค่าอื่น ๆ ที่นอกเหนือจากค่าที่เป็นไปได้และไม่ถูกต้องอยู่แล้ว ดังนั้นผู้วิจัยจึงได้มีการตัดชั้นสมมูลของทั้ง 2 ตัวแปรนี้ออกไป ส่งผลให้การสร้างกรณีทดสอบมีจำนวนลดลงจากที่ได้ด้วยวิธีการมือ 300 (5*4*3*5) กรณีทดสอบ เหลือแค่ 160 (5*4*2*4) กรณีทดสอบที่ได้จากต้นแบบแต่ยังคงครอบคลุมในทุก ๆ เงื่อนไขของการทดสอบซอฟต์แวร์ ดังตารางที่ 6.5

ตารางที่ 6.5 ค่าข้อมูลที่มีการเปลี่ยนแปลง

Partition	Num of Partition	Valid Data				Invalid Data		
Member (input)	5	AdminStaff (1)	Grad(2)	Lecturer(3)	UnderGrad(4)	None (5)		
Item (input)	4	Book (6)	CD (7)	DVD (8)		None (9)		
borrowDate (input)	3	Begin of Date – End of Date (10)				None (11)		Out of Range (12)
returnDate (input)	5	hasBorrowDate – (13) (hasBorrowDate + hasMaxDaysBook)		(hasBorrowDate + (14) hasMaxDaysBook) – End of Date		Begin of Date – (15) hasBorrowDate	None (16)	Out of Range (17)
maxDaysBorrow (fix rate)	4	7 (18)	14 (19)	30 (20)		None (21)		
daysOfLateReturn (output)	3	1 – Late Return (22)		None (23)		Smallest Number – 1 (24)		
fine (output)	3	1 x Fine Rate – (25) daysOfLateReturn x Fine Rate		None (26)		Smallest number – Fine Rate(27)		

6.3.1.2 กรณีศึกษาที่ 2 ระบบวิเคราะห์คำนวณค่าไต

กรณีศึกษาระบบวิเคราะห์คำนวณค่าไต เป็นกรณีศึกษาที่ได้นำเสนอไว้ในวิทยานิพนธ์ของทิวาทิพย์ (ทิวาทิพย์ ศรีรักษา, 2561) ซึ่งผู้วิจัยได้นำกรณีศึกษานี้มาเพื่อประเมินผลต้นแบบที่พัฒนาว่าสามารถสร้างกรณีทดสอบได้ถูกต้องและสอดคล้องกับที่ได้นำเสนอไว้หรือไม่ โดยผู้วิจัยได้ทำการสร้างกรณีทดสอบด้วยวิธีการมือ และสร้างกรณีทดสอบโดยต้นแบบที่พัฒนาขึ้น ตามความต้องการ เงื่อนไขการตัดสินใจ และข้อจำกัดต่าง ๆ ของระบบ ผลการประเมินหลังจากได้ทำการสร้างกรณีทดสอบด้วยวิธีการมือและจากต้นแบบที่ได้พบว่า ผลลัพธ์ของกรณีทดสอบมีความสอดคล้องกับกรณีทดสอบที่นำเสนอในวิทยานิพนธ์ข้างต้น โดยตัวแปรหรือค่าข้อมูลของตัวแปรรวมถึงจำนวนกรณีทดสอบที่สามารถสร้างได้ไม่มีการเปลี่ยนแปลง โดยกรณีทดสอบที่ได้จากตัวแปรและค่าข้อมูล

รวมถึงเงื่อนไขการตัดสินใจที่เกี่ยวข้องมีจำนวนทั้งหมด 144 กรณีทดสอบที่สอดคล้องกัน ดังนั้นจึงสรุปได้ว่าต้นแบบที่พัฒนาสามารถสร้างกรณีทดสอบได้ถูกต้องและครอบคลุมตามเงื่อนไขความต้องการของระบบ และสามารถนำไปใช้กับกรณีศึกษาอื่น ๆ ที่ไม่มีความซับซ้อน และมีรูปแบบความต้องการที่คล้ายคลึงกับกรณีศึกษาทั้ง 2 กรณีศึกษาที่นำมาใช้กับงานวิจัยได้

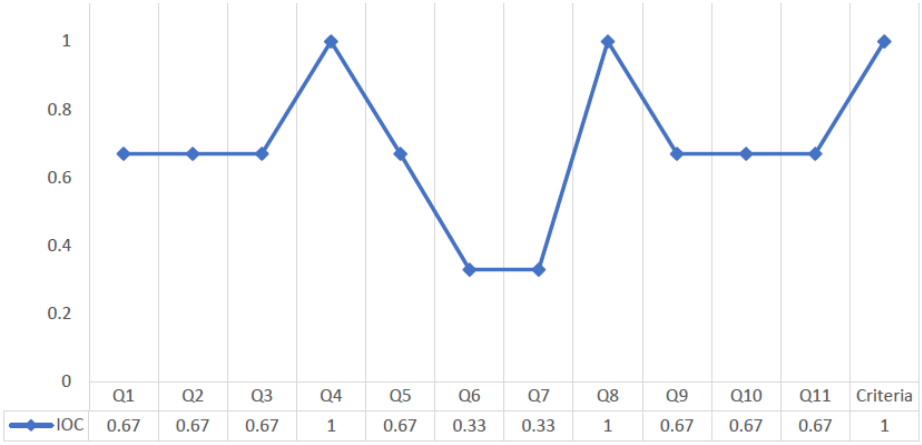
6.3.2 การประเมินผลต้นแบบโดยผู้เชี่ยวชาญ

ในการประเมินผลต้นแบบโดยผู้เชี่ยวชาญ ผู้วิจัยได้มีการออกแบบข้อคำถามสำหรับการประเมินต้นแบบที่พัฒนาขึ้น ซึ่งประกอบไปด้วยด้านต่าง ๆ ของการประเมิน 4 ด้านด้วยกัน ได้แก่ 1) ด้านฟังก์ชันการทำงาน 2) ด้านประสิทธิภาพและความน่าเชื่อถือ 3) ด้านการใช้งาน และ 4) ด้านความสามารถในการประยุกต์ใช้ โดยมีข้อคำถามทั้งหมดของด้านต่าง ๆ 11 ข้อด้วยกัน ดังตารางที่ 6.6

ตารางที่ 6.6 คำถามการประเมินต้นแบบ

ข้อคำถาม	คำถามการประเมิน
ก. ด้านฟังก์ชันการทำงาน	
Q1	ฟังก์ชันการทำงานมีความถูกต้องและเหมาะสม
Q2	ฟังก์ชันการทำงานสามารถทำงานร่วมกันได้
Q3	ฟังก์ชันการทำงานสามารถทำงานได้ตรงตามข้อกำหนดความต้องการ
ข. ด้านประสิทธิภาพและความน่าเชื่อถือ	
Q4	ต้นแบบสามารถประมวลผลกรณีทดสอบได้อย่างถูกต้อง
Q5	ต้นแบบสามารถเพิ่มโครงสร้างให้กับข้อกำหนดความต้องการได้
Q6	ต้นแบบสามารถจัดการข้อผิดพลาดได้
Q7	ต้นแบบสามารถทำงานได้อย่างสมบูรณ์
ค. ด้านการใช้งาน	
Q8	ต้นแบบสามารถเรียนรู้และศึกษาทำความเข้าใจได้ง่าย
Q9	ต้นแบบใช้งานได้ง่ายและฟังก์ชันการทำงานไม่ซับซ้อน
ง. ด้านความสามารถในการประยุกต์ใช้	
Q10	ต้นแบบสามารถนำไปประยุกต์ใช้ในระบบงานหรือกรณีศึกษาอื่น ๆ ได้
Q11	ต้นแบบสามารถนำไปติดตั้งใช้งานได้ง่าย

จากตารางที่ 6.6 เป็นข้อคำถามที่ใช้ในการประเมินผลการทำงานของต้นแบบ ซึ่งผู้วิจัยได้มีการตรวจสอบคุณภาพของเครื่องมือโดยผู้เชี่ยวชาญทางด้านวิศวกรรมซอฟต์แวร์และด้านสถิติการคำนวณจำนวน 3 ท่าน โดยทำการวัดความสอดคล้องระหว่างข้อคำถามกับวัตถุประสงค์ (IOC) รวมถึงเกณฑ์ที่ใช้ในการประเมินซึ่งพิจารณาตามมาตรวัดของลิเคิร์ท (Likert, 1932) เพื่อให้สอดคล้องกับวัตถุประส่งของงานวิจัยสำหรับประเมินต้นแบบในการสร้างกรณีทดสอบแบบอัตโนมัติด้วยออนโทโลยีความต้องการซอฟต์แวร์ จากการประเมินผลการตรวจสอบเครื่องมือโดยผู้เชี่ยวขาญมีค่าเฉลี่ย IOC ของข้อคำถามทั้งหมดอยู่ที่ 0.70 ซึ่งสามารถสรุปได้ว่าข้อคำถามสำหรับการประเมินสอดคล้องกับวัตถุประสงค์ของงานวิจัย และสามารถนำไปใช้ในการประเมินต้นแบบได้ แสดงแผนภาพกราฟเส้นสถิติ (Stacked Line) ของข้อคำถามการประเมิน ดังรูปที่ 6.7



รูปที่ 6.7 กราฟเส้นสถิติแสดงผลการประเมินคุณภาพของเครื่องมือสำหรับประเมินต้นแบบ

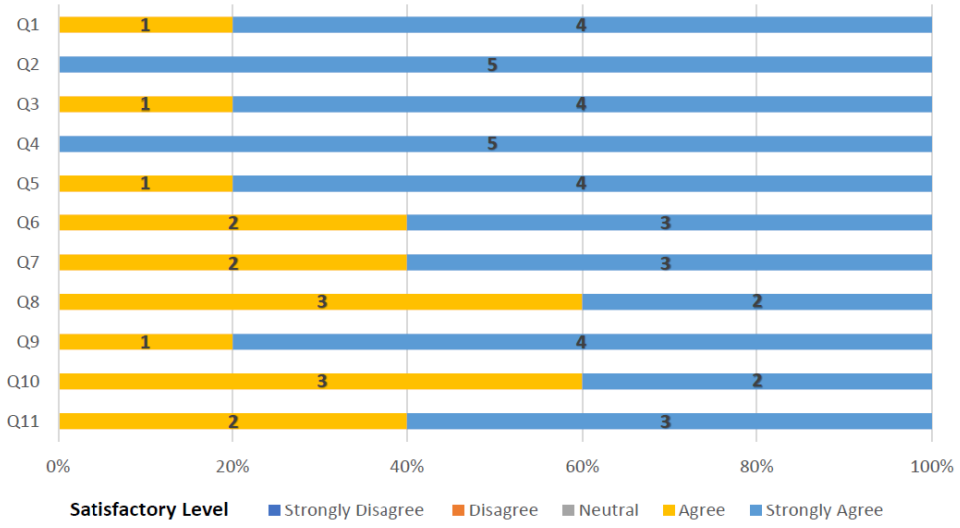
จากที่ผู้วิจัยได้นำเสนอเครื่องมือสำหรับการประเมินต้นแบบข้างต้น โดยให้ผู้เชี่ยวชาญทางด้านวิศวกรรมซอฟต์แวร์และการทดสอบซอฟต์แวร์ที่มีประสบการณ์โดยตรงไม่ต่ำกว่า 5 ปี จำนวน 5 ท่าน เป็นผู้ประเมิน สามารถแสดงรายละเอียดผู้เชี่ยวชาญ ดังตารางที่ 6.7

ตารางที่ 6.7 รายละเอียดจำนวนผู้เชี่ยวชาญในการประเมินต้นแบบ

ผู้เชี่ยวชาญ	จำนวน (ท่าน)
นักวิชาการ	2
นักทดสอบระบบ	3

ในการประเมินผู้เชี่ยวชาญจะประเมินด้วยข้อคำถามทั้งหมดจำนวน 11 ข้อ และใช้เกณฑ์การประเมินของลิเคิร์ท (Likert Scale) สำหรับประเมินข้อคำถาม 5 ระดับด้วยกัน ได้แก่ ระดับ 5 หมายถึง เห็นด้วยมากที่สุด (Strongly Agree), ระดับ 4 หมายถึงเห็นด้วย (Agree), ระดับ 3 หมายถึง เห็นด้วยปานกลาง (Neutral), ระดับ 2 หมายถึง เห็นด้วยน้อย (Disagree), และ ระดับ 1

หมายถึง เห็นด้วยน้อยที่สุด (Strongly Disagree) ซึ่งการประเมินผลความพึงพอใจของผู้เชี่ยวชาญทั้ง 5 ท่าน สามารถสรุปผลการประเมินด้วยแผนภาพกราฟแท่งสถิติ (Stacked Bar) ดังรูปที่ 6.8



รูปที่ 6.8 กราฟแท่งสถิติแสดงการกระจายระดับความพึงพอใจของผู้เชี่ยวชาญ

จากรูปที่ 6.8 เป็นกราฟแท่งสถิติแสดงการกระจายระดับความพึงพอใจของผู้เชี่ยวชาญในการประเมินต้นแบบงานวิจัย โดยใช้ข้อคำถาม 11 ข้อในการประเมินผลการทำงานของต้นแบบที่พัฒนาขึ้น จากผลการประเมินพึงพอใจของผู้เชี่ยวชาญทั้ง 5 ท่าน สามารถแสดงผลการประเมิน ดังตารางที่ 6.8

ตารางที่ 6.8 ผลการประเมินต้นแบบโดยผู้เชี่ยวชาญ

ข้อคำถาม	ผู้เชี่ยวชาญ					ค่าเฉลี่ย	ความหมายของความพึงพอใจ
	คนที่ 1	คนที่ 2	คนที่ 3	คนที่ 4	คนที่ 5		
Q1	5	5	5	4	5	4.8	มากที่สุด
Q2	5	5	5	5	5	5	มากที่สุด
Q3	5	5	5	4	5	4.8	มากที่สุด
Q4	5	5	5	5	5	5	มากที่สุด
Q5	5	5	5	5	4	4.8	มากที่สุด
Q6	4	5	5	5	4	4.6	มากที่สุด
Q7	5	5	5	4	4	4.6	มากที่สุด
Q8	5	5	4	4	4	4.4	มาก

ตารางที่ 6.8 ผลการประเมินต้นแบบโดยผู้เชี่ยวชาญ (ต่อ)

ข้อคำถาม	ผู้เชี่ยวชาญ					ค่าเฉลี่ย	ความหมายของความพึงพอใจ
	คนที่ 1	คนที่ 2	คนที่ 3	คนที่ 4	คนที่ 5		
Q9	5	5	5	4	5	4.8	มากที่สุด
Q10	5	4	4	4	5	4.4	มาก
Q11	5	5	4	5	4	4.6	มากที่สุด
ภาพรวม	**-**	**-**	**-**	**-**	**-**	**4.71**	**มากที่สุด**

จากตารางที่ 6.8 เป็นผลการประเมินต้นแบบโดยผู้เชี่ยวชาญ สำหรับการแปลผลความหมายของแต่ละระดับโดยใช้ค่าเฉลี่ยเป็นตัวชี้วัดตามแนวคิดของ Best and Kahn (Best and Kahn, 1977) มีรายละเอียดในการวิเคราะห์เกณฑ์การประเมิน ดังนี้

ค่าเฉลี่ย 4.50 – 5.00 หมายถึง ระดับความพึงพอใจมากที่สุด
ค่าเฉลี่ย 3.50 – 4.49 หมายถึง ระดับความพึงพอใจมาก
ค่าเฉลี่ย 2.50 – 3.49 หมายถึง ระดับความพึงพอใจปานกลาง
ค่าเฉลี่ย 1.50 – 2.49 หมายถึง ระดับความพึงพอใจน้อย
ค่าเฉลี่ย 1.00 – 1.49 หมายถึง ระดับความพึงพอใจน้อยที่สุด

จากผลรวมของค่าเฉลี่ยที่ได้ทำการประเมินความพึงพอใจของผู้เชี่ยวชาญตามแนวคิดข้างต้น พบว่าผู้เชี่ยวชาญมีความพึงพอใจคิดเป็นค่าเฉลี่ยรวมของข้อคำถามที่ใช้ในการประเมินทั้ง 11 ข้อ มีผลรวมค่าเฉลี่ยอยู่ที่ 4.71 ซึ่งเป็นระดับความพึงพอใจมากที่สุด และมีความสอดคล้องกับการกระจายของระดับความพึงพอใจที่ถูกประเมินโดยผู้เชี่ยวชาญ ซึ่งโดยส่วนมากจะมีความพึงพอใจอยู่ในระดับความพึงพอใจมากที่สุดเช่นเดียวกัน ดังนั้นจึงสรุปได้ว่าต้นแบบที่พัฒนาขึ้นตามกรอบแนวคิดของงานวิจัยสามารถสร้างกรณีทดสอบได้สอดคล้องกับวัตถุประสงค์ของงานวิจัยที่ได้นำเสนอไว้ อีกทั้งผู้เชี่ยวชาญยังให้คำแนะนำและข้อเสนอแนะเพื่อเป็นแนวทางในการปรับปรุงต้นแบบของงานวิจัยต่อไปในอนาคต ดังตารางที่ 6.9

ตารางที่ 6.9 คำแนะนำและข้อเสนอแนะของผู้เชี่ยวชาญ

ข้อ	คำแนะนะและข้อเสนอแนะของผู้เชี่ยวชาญ
ก. ปัญหาทางเทคนิค/การใช้งาน	
1.	จำเป็นต้องสร้างไฟล์ XML และ OWL ของข้อกำหนดความต้องการก่อน
2.	ควรแสดงขั้นตอนการทำงานแบบ progress bar เพื่อให้เห็นขั้นตอนได้ชัดเจนขึ้น

ตารางที่ 6.9 คำแนะนำและข้อเสนอแนะของผู้เชี่ยวชาญ (ต่อ)

ข้อ	คำแนะนะและข้อเสนอแนะของผู้เชี่ยวชาญ
ข. ข้อเสนอแนะเพิ่มเติม/แนวทางการปรับปรุง/สิ่งที่คาดหวัง	
1.	ควรมีการเพิ่มฟังก์ชันในการ generate ไฟล์ OWL ภายในต้นแบบเพื่อนำไปใช้งานได้เลย
2.	คาดหวังให้มีการระบุหรือขยายความคำศัพท์เฉพาะที่ใช้งานภายในต้นแบบ

6.4 สรุปการประเมินผลต้นแบบ

สรุปการประเมินผลต้นแบบสำหรับการสร้างกรณีทดสอบแบบอัตโนมัติด้วยออนโทโลยีความต้องการซอฟต์แวร์ โดยได้มีการจัดการข้อกำหนดความต้องการซึ่งเขียนด้วยภาษาธรรมชาติให้อยู่ในรูปแบบเชิงโครงสร้างและมีความชัดเจนก่อนจะนำไปสร้างกรณีทดสอบ โดยการเพิ่มโครงสร้างให้กับข้อกำหนดความต้องการจะใช้เครื่องมือ ROO ซึ่งเป็นเครื่องมือในการออกแบบและพัฒนาออนโทโลยี โดยเครื่องมือนี้สามารถช่วยขจัดความคลุมเครือของความต้องการที่ไม่ชัดเจนด้วยภาษาธรรมชาติควบคุม (CNL) ซึ่งมีการกำหนดไวยกรณ์ที่มีรูปแบบเฉพาะด้วยภาษาเชิงโครงสร้าง ทำให้การสร้างกรณีทดสอบด้วยออนโทโลยีความต้องการซอฟต์แวร์ตรงตามความต้องการมากยิ่งขึ้น รวมถึงการพิจารณาเทคนิคในการสร้างกรณีทดสอบด้วยการรวมเทคนิคการทดสอบแบบกล่องดำจำนวน 2 เทคนิค คือ เทคนิคการรวมกันของชั้นสมมูลและต้นไม้การจำแนกเพื่อให้การสร้างกรณีทดสอบมีความครอบคลุมในทุก ๆ กรณีของการทดสอบซอฟต์แวร์

จากการประเมินความถูกต้องในการสร้างกรณีทดสอบของกรณีศึกษาทั้ง 2 กรณีศึกษาด้วยวิธีการมือเปรียบเทียบกับการสร้างกรณีทดสอบแบบอัตโนมัติด้วยต้นแบบ สามารถสร้างกรณีทดสอบของกรณีศึกษาที่ 1 หลังจากที่ได้มีการปรับปรุงค่าข้อมูลเรียบร้อยแล้วได้ทั้งหมดจำนวน 160 กรณีทดสอบ และกรณีศึกษาที่ 2 จำนวน 144 กรณีทดสอบ ซึ่งสอดคล้องและตรงกันกับการสร้างกรณีทดสอบด้วยวิธีการมือและต้นแบบที่พัฒนาขึ้น รวมถึงผู้วิจัยได้ดำเนินการประเมินต้นแบบสำหรับการนำไปใช้งานในอนาคตโดยผู้เชี่ยวชาญทางด้านวิศวกรรมซอฟต์แวร์และการทดสอบซอฟต์แวร์ซึ่งมีประสบการณ์การทำงานโดยตรงไม่ต่ำกว่า 5 ปีมาเป็นผู้ประเมินต้นแบบของงานวิจัย จากผลการประเมินของผู้เชี่ยวชาญมีความเห็นตรงกันว่าต้นแบบสามารถสร้างกรณีทดสอบได้ตามวัตถุประสงค์ของงานวิจัยโดยเฉลี่ยจากข้อคำถามที่ใช้ในการประเมิน 11 ข้อ จากภาพรวมในตารางที่ 6.8 ผลการประเมินต้นแบบโดยผู้เชี่ยวชาญ มีความพึงพอใจเฉลี่ยอยู่ที่ 4.71 ซึ่งอยู่ในระดับความพึงพอใจมากที่สุด ดังนั้นจึงสรุปได้ว่าต้นแบบที่ได้ทำการพัฒนาสามารถสร้างกรณีทดสอบด้วยออนโทโลยีความต้องการซอฟต์แวร์ที่ได้มีการจัดการความต้องการให้อยู่ในรูปแบบเชิงโครงสร้างและมีความชัดเจนมากยิ่งขึ้นก่อนจะนำไปสร้างกรณีทดสอบ และสามารถสร้างกรณีทดสอบได้ครอบคลุมในทุก ๆ กรณีของการทดสอบซอฟต์แวร์ตามความต้องการของผู้ใช้งาน

บทที่ 7

อภิปรายและสรุปผลการวิจัย

ในการอภิปรายและสรุปผลการวิจัยจะกล่าวถึงบทสรุปของงานวิจัยที่ได้ดำเนินการในครั้งนี้ รวมถึงข้อจำกัดและข้อเสนอแนะต่าง ๆ เพื่อเป็นแนวทางและเป็นประโยชน์สำหรับการนำไปใช้งานต่อไปในอนาคต โดยมีรายละเอียดผลสรุปของงานวิจัยในแต่ละหัวข้อ ดังต่อไปนี้

7.1 อภิปรายผลการวิจัย

การอภิปรายผลการวิจัยของกรอบแนวคิดในการสร้างกรณีทดสอบด้วยออนโทโลยีความต้องการซอฟต์แวร์ที่ได้นำเสนอในงานวิจัยนี้ โดยได้มีการนำออนโทโลยีมาใช้ในการแทนความหมายข้อกำหนดความต้องการในเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ที่เขียนด้วยภาษาอังกฤษซึ่งไม่มีโครงสร้างให้อยู่ในรูปแบบเชิงโครงสร้างและมีความชัดเจนก่อนจะนำไปสร้างกรณีทดสอบ (คำถามงานวิจัย ข้อ 1) โดยได้มีการพิจารณานำเครื่องมือ ROO มาใช้ในการจัดการออนโทโลยีความต้องการซอฟต์แวร์ให้มีโครงสร้างที่ชัดเจนมากขึ้น ทำให้การสร้างกรณีทดสอบมีความถูกต้องและลดข้อผิดพลาดที่เกิดจากความไม่ชัดเจนของข้อกำหนดความต้องการซอฟต์แวร์ซึ่งเป็นสาเหตุหลักของความผิดพลาดที่เกิดขึ้นในกระบวนการทดสอบซอฟต์แวร์ โดยเครื่องมือ ROO จะมีโครงสร้างในการออกแบบและกำหนดออนโทโลยีด้วยภาษาธรรมชาติควบคุม (CNL) หรือที่เรียกว่า Rabbit Language ซึ่งทำให้ผู้เชี่ยวชาญด้านโดเมนสามารถออกแบบและพัฒนาออนโทโลยีได้ง่ายและไม่ซับซ้อนด้วยไวยกรณ์ที่มีรูปแบบเฉพาะของเครื่องมือในการขจัดความคลุมเครือของคำศัพท์ที่ใช้ในออนโทโลยี ทำให้สามารถมั่นใจได้ว่าความต้องการที่ไม่ชัดเจนของข้อกำหนดความต้องการซอฟต์แวร์ที่นำมาใช้ในการสร้างกรณีทดสอบได้ถูกขจัดออกไปแล้ว นอกจากนี้การรวมเทคนิคการทดสอบแบบกล่องดำจำนวน 2 เทคนิค คือ เทคนิคการรวมกันของชั้นสมมูลและต้นไม้การจำแนกมาใช้ในการสร้างกรณีทดสอบนั้น สามารถช่วยเพิ่มประสิทธิภาพให้กรณีทดสอบมีความครอบคลุมในทุก ๆ เงื่อนไขของการทดสอบและสามารถลดความซ้ำซ้อนของกรณีทดสอบในการทดสอบซอฟต์แวร์ได้ (คำถามงานวิจัย ข้อ 2)

โดยทางผู้วิจัยได้ทำการพัฒนาต้นแบบสำหรับการสร้างกรณีทดสอบแบบอัตโนมัติด้วยออนโทโลยีความต้องการซอฟต์แวร์ที่มีความครอบคลุมในทุก ๆ กรณีของการทดสอบและตรงตามความต้องการ เพื่อเป็นการสนับสนุนและยืนยันความถูกต้องในการทำงานของกรอบแนวคิดที่นำเสนอ (คำถามงานวิจัย ข้อ 3) สำหรับการทดสอบและประเมินผลการทำงานของต้นแบบผู้วิจัยได้

นำกรณีศึกษาจริง 2 กรณีศึกษา ได้แก่ กรณีศึกษาระบบจัดการห้องสมุด และกรณีศึกษาระบบวิเคราะห์คำนวณค่าไต รวมถึงการประเมินผลโดยผู้เชี่ยวชาญในการประเมินความถูกต้องและประสิทธิภาพการทำงานของต้นแบบตามกรอบแนวคิดของงานวิจัยที่ได้มีการออกแบบไว้ (คำถามงานวิจัย ข้อ 4) โดยเริ่มต้นผู้วิจัยได้ทำการสร้างกรณีทดสอบด้วยวิธีการมือเพื่อเปรียบเทียบกับผลลัพธ์ที่ได้จากการสร้างกรณีทดสอบแบบอัตโนมัติด้วยต้นแบบร่วมกับการประเมินโดยผู้เชี่ยวชาญในการตรวจสอบความถูกต้องของต้นแบบที่พัฒนาขึ้นและการประเมินความพึงพอใจสำหรับการนำไปใช้งานต่อไปในอนาคต พบว่าผู้เชี่ยวชาญมีความคิดเห็นสอดคล้องกับผลลัพธ์ในการประเมินผลตามวัตถุประสงค์ของงานวิจัยสำหรับการสร้างกรณีทดสอบด้วยออนโทโลยีความต้องการซอฟต์แวร์ที่ผู้วิจัยได้นำเสนอ จากผลลัพธ์ในการสร้างกรณีทดสอบด้วยวิธีการมือและการสร้างกรณีทดสอบแบบอัตโนมัติจากต้นแบบเป็นการยืนยันได้ว่ากรอบแนวคิดที่ผู้วิจัยนำเสนอสามารถสร้างกรณีทดสอบด้วยออนโทโลยีความต้องการซอฟต์แวร์ที่ได้มีการจัดการข้อกำหนดความต้องการให้มีโครงสร้างและมีความชัดเจนก่อนนำไปสร้างกรณีทดสอบ และสามารถสร้างกรณีทดสอบได้ครอบคลุมทุกกรณีของการทดสอบซอฟต์แวร์ได้

7.2 ข้อจำกัดและข้อเสนอแนะของการวิจัย

1) ต้นแบบของงานวิจัยที่พัฒนาขึ้นสามารถรองรับออนโทโลยีความต้องการซอฟต์แวร์ในรูปแบบไฟล์ที่เป็นไวยกรณ์แบบ OWL และไฟล์เอ็กซ์เอ็มแอล (XML) ในรูปแบบที่กำหนดไว้ในงานวิจัยนี้เท่านั้น

2) ข้อกำหนดความต้องการซอฟต์แวร์ที่นำมาใช้กับต้นแบบจำเป็นต้องมีลักษณะคล้ายคลึงกับความต้องการของกรณีศึกษาที่นำเสนอในงานวิจัย และสามารถนำมาสร้างกรณีทดสอบด้วยเทคนิคการรวมกันของชั้นสมมูลและต้นไม้การจำแนกได้

3) ต้นแบบสามารถสร้างกรณีทดสอบได้เฉพาะความต้องการที่ไม่ซับซ้อน เนื่องจากเงื่อนไขความต้องการที่มีการคำนวณข้อมูลทดสอบสามารถทำงานได้กับกรณีศึกษาที่นำเสนอเท่านั้น ในการคำนวณจะทำงานโดยฟังก์ชันภายในต้นแบบจึงไม่สามารถคำนวณข้อมูลทดสอบที่แตกต่างจากที่นำเสนอ แต่สามารถสร้างกรณีทดสอบให้ครอบคลุมเงื่อนไขของการทดสอบด้วยต้นไม้การตัดสินใจซึ่งมีการออกแบบฟังก์ชันการทำงานที่ผู้ใช้สามารถกำหนดเองได้ ในอนาคตอาจจะมีการพัฒนาฟังก์ชันเพิ่มเติมสำหรับการคำนวณข้อมูลทดสอบของระบบที่แตกต่างกันออกไปด้วยการดำเนินการผ่านหน้าจอส่วนต่อประสานกับผู้ใช้ หรือการเพิ่มสูตรการคำนวณลงไปในต้นแบบ เป็นต้น

4) ความเร็วในการทำงานของต้นแบบจะขึ้นอยู่กับจำนวนกรณีทดสอบและข้อมูลทดสอบของแต่ละระบบที่นำมาประมวลผล

7.3 สรุปผลการวิจัย

งานวิจัยนี้นำเสนอกรอบแนวคิดสำหรับการสร้างกรณีทดสอบด้วยออนโทโลยีความต้องการซอฟต์แวร์ โดยได้มีการนำออนโทโลยีมาใช้ในการแทนความหมายข้อกำหนดความต้องการในเอกสารข้อกำหนดคุณลักษณะความต้องการซอฟต์แวร์ (SRS) ซึ่งเขียนด้วยภาษาธรรมชาติให้อยู่ในรูปแบบที่มีโครงสร้างและมีความชัดเจนมากยิ่งขึ้นก่อนจะนำไปสร้างกรณีทดสอบ (วัตถุประสงค์งานวิจัย ข้อ 1) โดยได้พิจารณานำเครื่องมือ ROO มาช่วยในการจัดการออนโทโลยีความต้องการซอฟต์แวร์ด้วยภาษาธรรมชาติควบคุม (CNL) หรือที่เรียกว่า Rabbit Language ทำให้ข้อกำหนดความต้องการมีโครงสร้างและชัดเจนมากขึ้น เพื่อให้การสร้างกรณีทดสอบซอฟต์แวร์มีความถูกต้องและตรงตามความต้องการของผู้ใช้งาน รวมถึงการนำเทคนิคการรวมกันของชั้นสมมูลและต้นไม้การจำแนกมาใช้ในการสร้างกรณีทดสอบสามารถช่วยลดความซ้ำซ้อนของกรณีทดสอบ และสามารถทำให้มีความครอบคลุมเงื่อนไขการทดสอบในทุก ๆ กรณีของการทดสอบซอฟต์แวร์ (วัตถุประสงค์งานวิจัย ข้อ 2) โดยผู้วิจัยได้ทำการพัฒนาต้นแบบในการสร้างกรณีทดสอบแบบอัตโนมัติเพื่อสนับสนุนและยืนยันการทำงานของกรอบแนวคิดที่ได้มีการออกแบบไว้ ซึ่งสามารถทำงานได้สอดคล้องตามวัตถุประสงค์ของงานวิจัย (วัตถุประสงค์งานวิจัย ข้อ 3) และได้มีการนำกรณีศึกษาที่มีอยู่จริงมาใช้ในการทดสอบความถูกต้องและประเมินผลการทำงานของต้นแบบ รวมถึงการประเมินผลโดยผู้เชี่ยวชาญในการประเมินความพึงพอใจและการนำต้นแบบที่พัฒนาขึ้นไปใช้งานต่อไปในอนาคต (วัตถุประสงค์งานวิจัย ข้อ 4) เพื่อให้การสร้างกรณีทดสอบมีความถูกต้อง และครอบคลุมในทุก ๆ กรณีของการทดสอบซอฟต์แวร์ตามความต้องการของผู้ใช้งาน

ผลลัพธ์ที่ได้จากงานวิจัยนี้สามารถช่วยเพิ่มประสิทธิภาพให้กับการทดสอบแบบดั้งเดิมที่จะสร้างกรณีทดสอบจากความต้องการที่เป็นภาษาธรรมชาติซึ่งไม่มีโครงสร้างและอาจนำไปสู่การสร้างกรณีทดสอบที่ผิดพลาดและไม่ตรงตามความนต้องการได้ การนำออนโทโลยีมาใช้ในการแทนความหมายของข้อกำหนดความต้องการให้อยู่ในรูปแบบเชิงโครงสร้างและมีความชัดเจนก่อนจะนำไปสร้างกรณีทดสอบ รวมถึงการรวมกันของเทคนิคการทดสอบแบบกล่องดำทั้ง 2 เทคนิคมาใช้ในงานวิจัยนี้สามารถช่วยเพิ่มความครอบคลุมให้กับกระบวนการสร้างกรณีทดสอบในทุก ๆ เงื่อนไขของการทดสอบที่เป็นไปได้เพื่อให้นักทดสอบสามารถค้นหาข้อผิดพลาดและทำการแก้ไขได้ตรงตามความต้องการ และผู้วิจัยหวังเป็นอย่างยิ่งว่างานวิจัยนี้จะเป็นประโยชน์ไม่มากก็น้อยในการเพิ่มประสิทธิภาพของกระบวนการทดสอบซอฟต์แวร์เพื่อให้มีความถูกต้องและตรงตามความต้องการของผู้ใช้มากที่สุด รวมถึงสามารถช่วยลดภาระงานและค่าใช้จ่ายในการแก้ไขข้อผิดพลาดสำหรับการทดสอบ ส่งผลให้กระบวนการวิศวกรรมซอฟต์แวร์และการทดสอบซอฟต์แวร์มีประสิทธิภาพและน่าเชื่อถือมากยิ่งขึ้น

เอกสารอ้างอิง

ทิวาทิพย์ ศรีรักษา. (2561). กรอบแนวคิดสำหรับการวิเคราะห์ผลกระทบต่อกรณีทดสอบจากการเปลี่ยนแปลงคุณลักษณะความต้องการแบบยูสเคสด้วยแนวคิดการรวมกันระหว่างการสร้างข้อมูลทดสอบแบบการแบ่งชั้นสมมูลและต้นไม้การจำแนก. วิทยานิพนธ์วิทยาศาสตร์มหาบัณฑิต, สาขาเทคโนโลยีสารสนเทศ, คณะเทคโนโลยีและสิ่งแวดล้อม, มหาวิทยาลัยสงขลานครินทร์.

อิสรา ชื่นตา, จารี ทองคำ, และจิรัฏฐา ภูบุญอบ. (2015). การพัฒนาออนโทโลยีสำหรับระบบให้คำแนะนำข้อมูลการท่องเที่ยวในภาคตะวันออกเฉียงเหนือของประเทศไทย. วารสารวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยมหาสารคาม, *34*(1), 32–44.

Antoniou, G., & Van Harmelen, F. (2004). *A Semantic Web Primer* (2nd ed.). Cambridge, MA, USA: MIT press.

Barbosa, E., Nakagawa, E., & Maldonado, J. (2006). Towards the Establishment of an Ontology of Software Testing. *Proceedings of The 18th International Conference on Software Engineering & Knowledge Engineering (SEKE)*. San Francisco, CA, USA: 5-7 July 2006.

Berry, D. M., Kamsties, E., & Krieger, M. M. (2003). From contract drafting to software specification: Linguistic sources of ambiguity. *Perspectives on Software Requirements, Series: The Springer International Series in Engineering and Computer Science*, *753*, 7–44.

Best, J. W., & Kahn, J. V. (1977). *Research in Education* (3rd ed.). Englewood Cliffs, New Jersey: Prentice Hall, Inc.

Brady, F. (2013). *Cambridge university study states software bugs cost economy 312 billion per year*. PRWeb: Press Release Servises.

Castañeda, V., Ballejos, L., Caliusco, M., & Galli, M. (2010). The use of ontologies in requirements engineering. *Global Journal of Researches in Engineering*, *10*(6), 2–8.

Cockburn, A. (2000). *Writing Effective Use Cases*. Addison-Wesley: U.S.A.

Denaux, R. (2013). *Intuitive Ontology Authoring Using Controlled Natural Language*. University of Leeds: U.K.

Denaux, R., Dimitrova, V., Cohn, A., Dolbear, C., & Hart, G. (2009). Rabbit to OWL: Ontology Authoring with a CNL-based Tool. *Proceedings of The 2009 conference on Controlled natural language*. Marettimo Island, Italy: 8-10 June 2009.

Denaux, R., Holt, I., Dimitrova, V., Dolbear, C., & Cohn, A. G. (2008). Supporting the Construction of Conceptual Ontologies with the ROO Tool. *Proceedings of*

The 4th OWL Experiences and Directions Workshop (OWLED). Washington, DC, USA: 1-2 April 2008.

Grochtmann, M., Grimm, K., Wegener, J., & Grochtmann, M. (1993). Tool-Supported Test Case Design for Black-Box Testing by Means of the Classification-Tree Editor. *Proceedings of EuroSTAR*. Edwardian Hotel, London, UK: 25-28 October 1993.

Hajjamy, O., Alaoui, L., & Bahaj, M. (2017). XSD2OWL2: Automatic mapping from xml schema into owl2 ontology. *Journal of Theoretical and Applied Information Technology*, *95*(8), 17–30.

Hakimpour, F., & Geppert, A. (2001). Resolving Semantic Heterogeneity in Schema Integration: an Ontology Based Approach. *Proceedings of The 2nd International Conference on Formal Ontology in Information Systems*. Ogunquit Maine, USA: 17-19 October 2001.

Hakimpour, F., & Geppert, A. (2005). Resolution of semantic heterogeneity in database schema integration using formal ontologies. *Information Technology and Management*, *6*(1), 97–122.

IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications. (1998). IEEE Computer Society, Software Engineering Technical Committee. Retrieved from http://www.cse.msu.edu/ cse870/IEEEXplore-SRS-template.pdf

Jorgensen, P. C. (2003). *Software Testing: A Craftsman's Approach* (2nd ed.). Boca Raton: CRC press.

Kabmala, M., Manmart, L., & Malaivongs, K. (2006). Ontology approach for building. *Journal of Information Science*, *24*(1-3), 24–49.

Keet, M. C. (2020). *An Introduction to Ontology Engineering*. University of Cape Town: South Africa.

Khan, M. E. (2011). Different approaches to black box testing technique for finding errors. *International Journal of Software Engineering and Applications (IJSEA)*, *2*(4), 31–40. doi:10.5121/ijsea.2011.2404

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*, *140*, 5–55.

Luger, G. F. (2009). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving* (6th ed.). University of New Mexico: Pearson Education.

Minsky, M. (1974). *A Framework for Representing Knowledge*. MIT, Cambridge: U.S.A.

Montes, M., Gelbukh, A., Lopez-Lopez, A., & Baeza-Yates, R. (2001). Flexible Comparison of Conceptual Graphs. In H. C. Mayr, J. Lazansky, G. Quirchmayr, & P. Vogel (Eds.), *The International Conference on Database and Expert*

Systems Applications (pp. 102–111). Berlin, Heidelberg: Springer. doi:10.1007/3-540-44759-8_12

Noy, N. F., & Mcguinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology. *Knowledge Systems Laboratory*, *32*, 1–25.

Patton, R. (2005). *Software Testing* (2nd ed.). Indianapolis, USA: Sams Publishing.

Pilone, D., & Pitman, N. (2005). *UML 2.0 in a Nutshell*. O'Reilly Media, Inc.

Ramadoss, B., Prema, P., & Balasundaram, S. R. (2011). Combined Classification Tree Method for Test Suite Reduction. *Proceedings of The 2nd International Conference and workshop on Emerging Trends in Technology (ICWET)*. Mumbai, Maharashtra, India: 25-26 February 2011.

Ramirez, C., & Valdes, B. (2012). A general knowledge representation model of concepts. *Advances in Knowledge Representation*, 272. doi:10.5772/37113

Russell, S., & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach* (2nd ed.). University of California, Berkeley: Pearson Education.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures*. Psychology Press: U.K.

Schneider, G., & Winters, J. P. (2001). *Applying Use Cases: A Practical Guide* (2nd ed.). Pearson Education.

Siegemund, K., Thomas, E. J., Zhao, Y., Pan, J., & Assmann, U. (2011). Towards Ontology-Driven Requirements Engineering. *Proceeding of The 10th International Semantic Web Conference (ISWC)*. Bonn: October 2011.

Sim, W. W., & Brouse, P. (2014). Towards an ontology-based persona-driven requirements and knowledge engineering. *Procedia Computer Science*, *36*, 314–321. doi:10.1016/j.procs.2014.09.099

Souza, E., Falbo, R., & Vijaykumar, N. L. (2013). Using Ontology Patterns for Building a Reference Sofware Testing Ontology. *Proceedings of IEEE 17th International Enterprise Distributed Object Computing Workshop (EDOC)*. Vancouver, BC, Canada: 9-13 September 2013.

Sowa, J. F. (2008). Conceptual graphs. *Foundations of Artificial Intelligence*, *3*, 213–237.

Sugumaran, V., & Storey, V. C. (2002). Ontologies for conceptual modeling: their creation, use, and management. *Data & Knowledge Engineering*, *42*(3), 251–271. doi:https://doi.org/10.1016/S0169-023X(02)00048-4

Tarasov, V., Tan, H., Ismail, M., Adlemo, A., & Johansson, M. (2016). Application of Inference Rules to a Software Requirements Ontology to Generate Software Test Cases. *Proceeding of The 13th International Workshop, OWLED, and The 5th International Workshop, ORE*. Bologna, Italy: 20 November 2016.

Thongglin, K., Cardey, S., & Greenfield, P. (2013). Thai Software Requirements Specification Pattern. *Proceeding of IEEE 12th International Conference on Intelligent Software Methodologies, Tools and Techniques (SoMeT)*. Budapest, Hungary: 22-24 September 2013.

Ul Haq, S., & Qamar, U. (2019). Ontology Based Test Case Generation for Black Box Testing. *Proceedings of The 8th International Conference on Educational and Information Technology (ICEIT)*. Cambridge, UK: 2-4 March 2019.

Wang, Y., Bai, X., Li, J., & Huang, R. (2007). Ontology-Based Test Case Generation for Testing Web Services. *Proceedings of IEEE 8th International Symposium on Autonomous Decentralized Systems (ISADS)*. Sedona, AZ, USA: 21-23 March 2007.

ภาคผนวก ก

กฎการแปลงออนโทโลยีจาก OWL เป็น XML

ภาคผนวก ก

กฎการแปลงออนโทโลยีจาก OWL เป็น XML

ก1 ตารางกฎการแปลงออนโทโลยีจาก OWL เป็น XML

Rules	**OWL2 Functional**	**XML Schema**
The structure of Classes and Relations		
1. Class Declaration	Class(*:cname*)	Local: <xs:element name="*cname*"> <xs:complexType> </xs:complexType> </xs:element> Global: <xs:element name="*cname*"/>
2. Subclass Declaration	SubClassOf(*:scname :cname*)	<xs:element name="*scname*" type="*cname*"/>
3. Object Property Declaration	ObjectProperty(*:hasClass*) ObjectPropertyDomain(*:hasClass :cdomain*) ObjectPropertyRange:(*:hasClass :crange*)	<xs:element name="*cdomain*"> <xs:complexType> <xs:sequence> <xs:element name="*hasClass*" ref="*crange*"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="*crange*"> <xs:complexType> </xs:complexType> </xs:element>
4. Data Property Declaration	DataProperty(*:dprop*) DataPropertyDomain(*:dprop :cname*) DataPropertyRange(*:dprop* xsd:*dtype*)	<xs:element name="*cname*"> <xs:complexType> <xs:sequence> <xs:element name="*dprop*" type="xs:*dtype*"/> </xs:sequence> </xs:complexType> </xs:element>

ก1 ตารางกฎการแปลงออนโทโลยีจาก OWL เป็น XML (ต่อ)

Rules	OWL2 Functional	XML Schema
Object Property Restrictions		
5. Existential	EquivalentClasses(*:cdomain* ObjectSomeValuesFrom(*:hasClass :cname*))	<xs:element name="*cdomain*"> <xs:complexType> <xs:sequence> <xs:element name="*hasClass*" ref="*crange*"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="*crange*"> <xs:complexType> <xs:choice> </xs:choice> </xs:complexType> </xs:element>
6. Universal	EquivalentClasses(*:cdomain* ObjectAllValuesFrom(*:hasClass :crange*))	<xs:element name="*cdomain*"> <xs:complexType> <xs:sequence> <xs:element name="*hasClass*" ref="*crange*"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="*crange*"> <xs:complexType> <xs:all> </xs:all> </xs:complexType> </xs:element>
7. Minimum cardinality	EquivalentClasses(*:cdomain* ObjectMinCardinality(*min :hasClass :crange*))	<xs:element name="*hasClass*" ref="*crange*" minOccurs="*min*"/>
8. Maximum cardinality	EquivalentClasses(*:cdomain* ObjectMaxCardinality(*max :hasClass :crange*))	<xs:element name="*hasClass*" ref="*crange*" maxOccurs="*max*"/>
Data Property Restrictions		
9. Existential	EquivalentClasses(*:cname* DataSomeValuesFrom(*:dprop* DataIntersectionOf(Datatype Restriction(*xsd:dtype xsd:minInclusive "min"*^^xsd:*dtype*) DatatypeRestriction(*xsd:dtype xsd:maxInclusive "max"*^^xsd:*dtype*))))	<xs:element name="*dprop*"> <xs:simpleType> <xs:restiction base="xs:*dtype*"> <*xs:minInclusive* value="*min*"/> <*xs:maxInclusive* value="*max*"/> </xs:restiction> </xs:simpleType> </xs:element>
10. Universal	EquivalentClasses(*:cname* DataAllValuesFrom(*:dprop xsd:dtype*))	<xs:element name="*dprop*" type="xs:*dtype*"/>

ก1 ตารางกฎการแปลงออนโทโลยีจาก OWL เป็น XML (ต่อ)

Rules	OWL2 Functional	XML Schema
11. Individual value	NameIndividual(*:individual1*) NameIndividual(*:individual2*) NameIndividual(*:individual3*) ClassAssertion(*:cname1 :individual1*) DataPropertyAssertion(*:dprop :individual1 "value1"*^^xsd:*dtype*) ClassAssertion(*:cname1 :individual2*) DataPropertyAssertion(*:dprop :individual2 "value2"*^^xsd:*dtype*) ClassAssertion(*:cname1 :individual3*) DataPropertyAssertion(*:dprop :individual3 "value3"*^^xsd:*dtype*	<xs:element name="*dprop*"> <xs:complexType> <xs:choice> <xs:element name="*value1*" type="xs:*dtype*"/> <xs:element name="*value2*" type="xs:*dtype*"/> <xs:element name="*value3*" type="xs:*dtype*"/> </xs:choice> </xs:complexType> </xs:element>
12. Minimum cardinality	EquivalentClasses(*:cname* DataMinCardinality(*min :dprop xsd:dtype*))	<xs:element name="*cname*"> <xs:complexType> <xs:sequence> <xs:element name="*dprop*" type="xs:*dtype*" minOccurs="*min*"/> </xs:sequence> </xs:complexType> </xs:element>
13. Maximum cardinality	EquivalentClasses(*:cname* DataMaxCardinality(*max :dprop xsd:dtype*))	<xs:element name="*cname*"> <xs:complexType> <xs:sequence> <xs:element name="*dprop*" type="xs:*dtype*" maxOccurs="*max*"/> </xs:sequence> </xs:complexType> </xs:element>

ก2 ตารางตัวอย่างการแปลงของกฎทั้ง 13 กฎ

Rules	OWL2 Functional	XML Schema
The structure of Classes and Relations		
1. Class Declaration	Class(*:Member*)	Local: <xs:element name="*Member*"> <xs:complexType> </xs:complexType> </xs:element> Global: <xs:element name="*Member*"/>
2. Subclass Declaration	SubClassOf(*:Book :Item*)	<xs:element name="*Book*" type="*Item*"/>
3. Object Property Declaration	ObjectProperty(*:hasBorrow*) ObjectPropertyDomain(*:hasBorrow :Member*) ObjectPropertyRange:(*:hasBorrow :Item*)	<xs:element name="*Member*"> <xs:complexType> <xs:sequence> <xs:element name="*hasBorrow*" ref="*Item*"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="*Item*"> <xs:complexType> </xs:complexType> </xs:element>
4. Data Property Declaration	DataProperty(*:address*) DataPropertyDomain(*:address :Member*) DataPropertyRange(*:address* xsd:*string*)	<xs:element name="*Member*"> <xs:complexType> <xs:sequence> <xs:element name="*address*" type="xs:*string*"/> </xs:sequence> </xs:complexType> </xs:element>
Object Property Restrictions		
5. Existential	EquivalentClasses(*:Member* ObjectSomeValuesFrom(*:hasBorrow :Item*))	<xs:element name="*Member*"> <xs:complexType> <xs:sequence> <xs:element name="*hasBorrow*" ref="*Item*"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="*Item*"> <xs:complexType> <xs:choice> <xs:element name="*Book*" type="xs:*string*"/> <xs:element name="*CD*" type="xs:*string*"/> <xs:element name="*DVD*" type="xs:*string*"/> </xs:choice> </xs:complexType> </xs:element>

ก2 ตารางตัวอย่างการแปลงของกฎทั้ง 13 กฎ (ต่อ)

Rules	OWL2 Functional	XML Schema
6. Universal	EquivalentClasses(*:Member* ObjectAllValuesFrom(*:hasInfo :Information*))	<xs:element name="*Member*"> <xs:complexType> <xs:sequence> <xs:element name="*hasInfo*" ref="*Information*"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="*Information*"> <xs:complexType> <xs:all> <xs:element name="*firstname*" type="*xs:string*"/> <xs:element name="*lastname*" type="*xs:string*"/> </xs:all> </xs:complexType> </xs:element>
7. Minimum cardinality	EquivalentClasses(*:Member* ObjectMinCardinality(*1 :hasBorrow :Item*))	<xs:element name="*hasBorrow*" ref="*Item*" minOccurs="*1*"/>
8. Maximum cardinality	EquivalentClasses(*:Member* ObjectMaxCardinality(*15 :hasBorrow :Item*))	<xs:element name="*hasBorrow*" ref="*Item*" maxOccurs="*15*"/>
Data Property Restrictions		
9. Existential	EquivalentClasses(*:Information* DataSomeValuesFrom(*:age* DataIntersectionOf(Datatype Restriction(*xsd:integer xsd:minInclusive "0"*^^xsd:*integer*) DatatypeRestriction(*xsd:integer xsd:maxInclusive "120"*^^xsd:*integer*))))	<xs:element name="*age*"> <xs:simpleType> <xs:restiction base="xs:*integer*"> <*xs:minInclusive* value="*0*"/> <*xs:maxInclusive* value="*120*"/> </xs:restiction> </xs:simpleType> </xs:element>
10. Universal	EquivalentClasses(*:Information* DataAllValuesFrom(*:passportNumber xsd:string*))	<xs:element name="*passportNumber*" type="xs:*string*"/>
11. Individual value	NameIndividual(*:individual1*) NameIndividual(*:individual2*) NameIndividual(*:individual3*) ClassAssertion(*:Book :individual1*) DataPropertyAssertion(*:maxDays Borrow :individual1 "7"*^^xsd:*integer*) ClassAssertion(*:Book :individual2*) DataPropertyAssertion(*:maxDays Borrow :individual2 "14"*^^xsd:*integer*) ClassAssertion(*:Book :individual3*) DataPropertyAssertion(*:maxDays Borrow :individual3 "30"*^^*xsd:integer*)	<xs:element name="*maxDaysBorrow*"> <xs:complexType> <xs:choice> <xs:element name="*7*" type="xs:*integer*"/> <xs:element name="*14*" type="xs:*integer*"/> <xs:element name="*30*" type="xs:*integer*"/> </xs:choice> </xs:complexType> </xs:element>

ก2 ตารางตัวอย่างการแปลงของกฎทั้ง 13 กฎ (ต่อ)

Rules	OWL2 Functional	XML Schema
12. Minimum cardinality	EquivalentClasses(*:Information* DataMinCardinality(*1 :phoneNumber xsd:integer*))	<xs:element name="*Information*"> <xs:complexType> <xs:sequence> <xs:element name="*phoneNumber*" type="xs:*integer*" minOccurs="*1*"/> </xs:sequence> </xs:complexType> </xs:element>
13. Maximum cardinality	EquivalentClasses(*:Information* DataMaxCardinality(*3 :phoneNumber xsd:integer*))	<xs:element name="*Information*"> <xs:complexType> <xs:sequence> <xs:element name="*phoneNumber*" type="xs:*integer*" maxOccurs="*3*"/> </xs:sequence> </xs:complexType> </xs:element>

ภาคผนวก ข

รูปแบบไฟล์นำเข้าของต้นแบบ

ภาคผนวก ข

รูปแบบไฟล์นำเข้าของต้นแบบ

ข1 ไฟล์นำเข้าออนโทโลยีความต้องการ .owl ระบบจัดการห้องสมุด

```
Prefix(:=<http://www.semanticweb.org/four_/ontologies/2019/6/untitled-ontology-13#>)
Prefix(owl:=<http://www.w3.org/2002/07/owl#>)
Prefix(rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Prefix(xml:=<http://www.w3.org/XML/1998/namespace>)
Prefix(xsd:=<http://www.w3.org/2001/XMLSchema#>)
Prefix(rdfs:=<http://www.w3.org/2000/01/rdf-schema#>)


Ontology(<http://www.semanticweb.org/four_/ontologies/2019/6/untitled-ontology-13>

Declaration(Class(:AdminStaff))
Declaration(Class(:Book))
Declaration(Class(:CD))
Declaration(Class(:DVD))
Declaration(Class(:Grad))
Declaration(Class(:Item))
Declaration(Class(:Lecturer))
Declaration(Class(:Member))
Declaration(Class(:UnderGrad))
Declaration(ObjectProperty(:hasBorrow))
Declaration(ObjectProperty(:hasReturn))
Declaration(DataProperty(:borrowDate))
Declaration(DataProperty(:daysOfLateReturn))
Declaration(DataProperty(:fine))
Declaration(DataProperty(:maxDaysBorrow))
Declaration(DataProperty(:returnDate))
Declaration(NamedIndividual(:borrowDaysR1))
Declaration(NamedIndividual(:borrowDaysR2))
Declaration(NamedIndividual(:borrowDaysR3))
```

ข1 ไฟล์นำเข้าออนโทโลยีความต้องการ .owl ระบบจัดการห้องสมุด (ต่อ)

```
##############################
#   Object Properties
##############################

# Object Property: :hasBorrow (:hasBorrow)

ObjectPropertyDomain(:hasBorrow :Member)
ObjectPropertyRange(:hasBorrow :Item)

# Object Property: :hasReturn (:hasReturn)

ObjectPropertyDomain(:hasReturn :Member)
ObjectPropertyRange(:hasReturn :Item)


##############################
#   Data Properties
##############################

# Data Property: :borrowDate (:borrowDate)

DataPropertyDomain(:borrowDate ObjectAllValuesFrom(:hasBorrow :Member))
DataPropertyRange(:borrowDate xsd:dateTime)

# Data Property: :daysOfLateReturn (:daysOfLateReturn)

DataPropertyDomain(:daysOfLateReturn ObjectAllValuesFrom(:hasReturn :Member))
DataPropertyRange(:daysOfLateReturn xsd:integer)

# Data Property: :fine (:fine)

DataPropertyDomain(:fine ObjectAllValuesFrom(:hasReturn :Member))
DataPropertyRange(:fine xsd:integer)

# Data Property: :maxDaysBorrow (:maxDaysBorrow)

DataPropertyDomain(:maxDaysBorrow ObjectAllValuesFrom(:hasBorrow :Member))
DataPropertyRange(:maxDaysBorrow xsd:integer)
```

ข1 ไฟล์นำเข้าออนโทโลยีความต้องการ .owl ระบบจัดการห้องสมุด (ต่อ)

```
# Data Property: :returnDate (:returnDate)

DataPropertyDomain(:returnDate ObjectAllValuesFrom(:hasReturn :Member))
DataPropertyRange(:returnDate xsd:dateTime)

#############################
#   Classes
#############################

# Class: :AdminStaff (:AdminStaff)

SubClassOf(:AdminStaff :Member)
DisjointClasses(:AdminStaff :Grad)
DisjointClasses(:AdminStaff :Lecturer)
DisjointClasses(:AdminStaff :UnderGrad)

# Class: :Book (:Book)

SubClassOf(:Book :Item)
DisjointClasses(:Book :CD)
DisjointClasses(:Book :DVD)

# Class: :CD (:CD)

SubClassOf(:CD :Item)
DisjointClasses(:CD :DVD)

# Class: :DVD (:DVD)

SubClassOf(:DVD :Item)

# Class: :Grad (:Grad)

SubClassOf(:Grad :Member)
DisjointClasses(:Grad :Lecturer)
DisjointClasses(:Grad :UnderGrad)
```

ข1 ไฟล์นำเข้าออนโทโลยีความต้องการ .owl ระบบจัดการห้องสมุด (ต่อ)

```
# Class: :Lecturer (:Lecturer)

SubClassOf(:Lecturer :Member)

DisjointClasses(:Lecturer :UnderGrad)

# Class: :UnderGrad (:UnderGrad)

SubClassOf(:UnderGrad :Member)

#############################
#    Named Individuals
#############################

# Individual: :borrowDaysR1 (:borrowDaysR1)

ClassAssertion(:AdminStaff :borrowDaysR1)
ClassAssertion(:Book :borrowDaysR1)
ClassAssertion(:CD :borrowDaysR1)
ClassAssertion(:DVD :borrowDaysR1)
ClassAssertion(:Grad :borrowDaysR1)
ClassAssertion(:Lecturer :borrowDaysR1)
ClassAssertion(:UnderGrad :borrowDaysR1)
DataPropertyAssertion(:maxDaysBorrow :borrowDaysR1 "7"^^xsd:integer)

# Individual: :borrowDaysR2 (:borrowDaysR2)

ClassAssertion(:Book :borrowDaysR2)
ClassAssertion(:Grad :borrowDaysR2)
DataPropertyAssertion(:maxDaysBorrow :borrowDaysR2 "14"^^xsd:integer)

# Individual: :borrowDaysR3 (:borrowDaysR3)

ClassAssertion(:Book :borrowDaysR3)
ClassAssertion(:Lecturer :borrowDaysR3)
DataPropertyAssertion(:maxDaysBorrow :borrowDaysR3 "30"^^xsd:integer)

)
```

ข2 ไฟล์นำเข้ายูสเคส .xml ระบบจัดการห้องสมุด

```
<?xml version="1.0" encoding="UTF-8"?>
<usecases>
   <usecase usecase_id="UC001">
      <usecase_name>Borrow Item</usecase_name>
      <actor_goal>Member</actor_goal>
      <pre_conditions>Checking members of the library</pre_conditions>
      <post_conditions>Borrow items successfully</post_conditions>
      <priority>High</priority>
      <flow_of_event>
         1.The system shows GUI for a list of items to borrow
         2.Member select items to borrow
         3.Member confirm borrow items [A1]
         4.The system records the borrowed items
         5.The system displays a list of borrowed items
      </flow_of_event>
      <alternative_flow>
         [A1] If click "Cancel" button, the system will not records [E1]
      </alternative_flow>
      <exception_flow>
         [E1] The system shows the warning "Confirm cancellation?"
      </exception_flow>
   </usecase>
   <usecase usecase_id="UC002">
      <usecase_name>Return Item</usecase_name>
      <actor_goal>Member</actor_goal>
      <pre_conditions>The system shows borrow items</pre_conditions>
      <post_conditions>The system calculated the fine of borrow item</post_conditions>
      <priority>High</priority>
      <flow_of_event>
         1.The system shows a list of borrowed items
         2.Member select item for return [A1]
         3.The system calculated the fine of the returned item
         4.The system shows fine to pay
      </flow_of_event>
      <alternative_flow>
         [A1] If the item to return is not selected, the system will display to select [E1]
      </alternative_flow>
```

ข2 ไฟล์นำเข้ายูสเคส .xml ระบบจัดการห้องสมุด (ต่อ)

```
      <exception_flow>
         [E1] The system shows the warning "Please select the item to return"
      </exception_flow>
   </usecase>
    <usecase usecase_id="UC003">
      <usecase_name>Calculate Fine</usecase_name>
      <actor_goal>Librarian</actor_goal>
      <pre_conditions>Member return item</pre_conditions>
      <post_conditions>The system shows detail of calculated fine</post_conditions>
      <priority>High</priority>
      <flow_of_event>
         1.The system calculated fine
         2.The system shows detail of calculated fine
      </flow_of_event>
      <alternative_flow>
         -
      </alternative_flow>
      <exception_flow>
         -
      </exception_flow>
   </usecase>
</usecases>
```

ข3 ไฟล์นำเข้าออนโทโลยีความต้องการ .owl ระบบวิเคราะห์คำนวณค่าไต

```
Prefix(:=<http://www.semanticweb.org/four_/GFRCalculated#>)
Prefix(owl:=<http://www.w3.org/2002/07/owl#>)
Prefix(rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Prefix(xml:=<http://www.w3.org/XML/1998/namespace>)
Prefix(xsd:=<http://www.w3.org/2001/XMLSchema#>)
Prefix(rdfs:=<http://www.w3.org/2000/01/rdf-schema#>)


Ontology(<http://www.semanticweb.org/four_/GFRCalculated>

Declaration(Class(:ESRD))
Declaration(Class(:Failure))
Declaration(Class(:GFR))
Declaration(Class(:Injury))
Declaration(Class(:Loss))
Declaration(Class(:Risk))
Declaration(Class(:Stage))
Declaration(Class(:UO))
Declaration(ObjectProperty(:hasGFR))
Declaration(ObjectProperty(:hasUO))
Declaration(ObjectProperty(:hasPair))
Declaration(DataProperty(:Age))
Declaration(DataProperty(:Height))
Declaration(DataProperty(:SCr))
Declaration(DataProperty(:Sex))
Declaration(NamedIndividual(:sex1))
Declaration(NamedIndividual(:sex2))
#############################
#   Object Properties
#############################
# Object Property: :hasGFR (:hasGFR)

ObjectPropertyDomain(:hasGFR :Stage)
ObjectPropertyRange(:hasGFR :GFR)

# Object Property: :hasUO (:hasUO)
```

ข3 ไฟล์นำเข้าออนโทโลยีความต้องการ .owl ระบบวิเคราะห์คำนวณค่าไต (ต่อ)

```
ObjectPropertyDomain(:hasUO :Stage)
ObjectPropertyRange(:hasUO :UO)

# Object Property: :hasPair (:hasPair)

ObjectPropertyDomain(:hasPair :GFR)
ObjectPropertyRange(:hasPair :UO)

#############################
#   Data Properties
#############################
# Data Property: :Age (:Age)

DataPropertyDomain(:Age :GFR)
DataPropertyRange(:Age xsd:integer)

# Data Property: :Height (:Height)

DataPropertyDomain(:Height :GFR)
DataPropertyRange(:Height xsd:integer)

# Data Property: :SCr (:SCr)

DataPropertyDomain(:SCr :GFR)
DataPropertyRange(:SCr xsd:float)

# Data Property: :Sex (:Sex)

DataPropertyDomain(:Sex :GFR)
DataPropertyRange(:Sex xsd:string)

#############################
#   Classes
#############################
# Class: :ESRD (:ESRD)

SubClassOf(:ESRD :Stage)
```

ข3 ไฟล์นำเข้าออนโทโลยีความต้องการ .owl ระบบวิเคราะห์คำนวณค่าไต (ต่อ)

```
DisjointClasses(:ESRD :Failure)
DisjointClasses(:ESRD :Injury)
DisjointClasses(:ESRD :Loss)
DisjointClasses(:ESRD :Risk)

# Class: :Failure (:Failure)

SubClassOf(:Failure :Stage)
DisjointClasses(:Failure :Injury)
DisjointClasses(:Failure :Loss)
DisjointClasses(:Failure :Risk)

# Class: :GFR (:GFR)

EquivalentClasses(:GFR DataSomeValuesFrom(:Height
DataIntersectionOf(DatatypeRestriction(xsd:integer xsd:minInclusive "0"^^xsd:integer)
DatatypeRestriction(xsd:integer xsd:maxInclusive "300"^^xsd:integer))))
EquivalentClasses(:GFR DataSomeValuesFrom(:Sex xsd:string))

# Class: :Injury (:Injury)

SubClassOf(:Injury :Stage)
DisjointClasses(:Injury :Loss)
DisjointClasses(:Injury :Risk)

# Class: :Loss (:Loss)

SubClassOf(:Loss :Stage)
DisjointClasses(:Loss :Risk)

# Class: :Risk (:Risk)

SubClassOf(:Risk :Stage)
```

ข3 ไฟล์นำเข้าออนโทโลยีความต้องการ .owl ระบบวิเคราะห์คำนวณค่าไต (ต่อ)

```
##############################
#   Named Individuals
##############################
# Individual: :sex1 (:sex1)

DataPropertyAssertion(:Sex :sex1 "Female"^^xsd:string)

# Individual: :sex2 (:sex2)

DataPropertyAssertion(:Sex :sex2 "Male"^^xsd:string)



)
```

ข4 ไฟล์นำเข้ายูสเคส .xml ระบบวิเคราะห์คำนวณค่าไต

```
<?xml version="1.0" encoding="UTF-8"?>
<usecases>
   <usecase usecase_id="UC001">
      <usecase_name>GFR Calculated</usecase_name>
      <actor_goal>Nurse</actor_goal>
      <pre_conditions>Select the patient from the database</pre_conditions>
      <post_conditions>The system records information to the database</post_conditions>
      <priority>High</priority>
      <flow_of_event>
         1.The system shows the GUI to input data
         2.Nurse inputs medical records into sex, age, and SCr
         3.Nurse save medical records [A1]
         4.The system calculated GFR from inputs [A2]
         5.The system interpreted the stage of GFR
         6.The system displays the stage of GFR
      </flow_of_event>
      <alternative_flow>
         [A1] If click "Cancel" button, the system will not records [E1]
         [A2] If pairing the stage of GFR failed [E2][E3]
      </alternative_flow>
      <exception_flow>
         [E1] The system shows the warning "Confirm cancellation?"
         [E2] The system shows the warning "Pairing failed"
         [E3] The system shows the warning "The stage of GFR failed"
      </exception_flow>
   </usecase>
   <usecase usecase_id="UC002">
      <usecase_name>GFR Interpreted</usecase_name>
      <actor_goal>Nurse</actor_goal>
      <pre_conditions>The system has already recorded GFR and UO</pre_conditions>
      <post_conditions>The system records the stage of GFR</post_conditions>
      <priority>High</priority>
      <flow_of_event>
         1.The system shows the result of the UO
         2.The system pair GFR and UO [A1]
         3.The system interpreted the stage of GFR and recording to the database
         4.The system shows the stage of GFR
      </flow_of_event>
```

ข4 ไฟล์นำเข้ายูสเคส .xml ระบบวิเคราะห์คำนวณค่าไต (ต่อ)

```
        <alternative_flow>
            [A1] If pairing the stage of GFR failed [E1][E2]
        </alternative_flow>
        <exception_flow>
            [E1] The system shows the warning "Pairing failed"
            [E2] The system shows the warning "The stage of GFR failed"
        </exception_flow>
    </usecase>
    <usecase usecase_id="UC003">
        <usecase_name>GFR Displayed</usecase_name>
        <actor_goal>Nurse</actor_goal>
        <pre_conditions>The system has already recorded GFR to the database</pre_conditions>
        <post_conditions>The system displays the stage of GFR</post_conditions>
        <priority>High</priority>
        <flow_of_event>
            1.Select the patient from the database
            2.The system displays the stage of GFR
        </flow_of_event>
        <alternative_flow>
            -
        </alternative_flow>
        <exception_flow>
            -
        </exception_flow>
    </usecase>
</usecases>
```

ภาคผนวก ค

กรณีทดสอบของทั้ง 2 กรณีศึกษา

ภาคผนวก ค

กรณีทดสอบทั้ง 2 กรณีศึกษา

ค1 ตารางกรณีทดสอบระบบจัดการห้องสมุด

Test Case	Member	Item	borrowDate	returnDate	maxDays Borrow	daysOf LateReturn	Fine	Comments
1	AdminStaff	Book	1/7/2019	2/7/2019	7	0	0	Valid (no late return)
2	AdminStaff	Book	5/7/2019	15/7/2019	7	3	9	Valid (late return)
3	AdminStaff	Book	8/7/2019	30/6/2019	7	-1	-3	Invalid (return date)
4	AdminStaff	Book	8/7/2019	None	7	None	None	Invalid (none return date)
5	AdminStaff	Book	None	2/7/2019	7	None	None	Invalid (none borrow date)
6	AdminStaff	Book	None	15/7/2019	7	None	None	Invalid
7	AdminStaff	Book	None	30/6/2019	7	None	None	Invalid
8	AdminStaff	Book	None	None	7	None	None	Invalid
9	AdminStaff	CD	1/5/2019	3/5/2019	7	2	0	Valid (no late return)

ค1 ตารางกรณีทดสอบระบบจัดการห้องสมุด (ต่อ)

Test Case	Member	Item	borrowDate	returnDate	maxDays Borrow	daysOf LateReturn	Fine	Comments
10	AdminStaff	CD	10/5/2019	27/5/2019	7	10	30	Valid (late return)
11	AdminStaff	CD	15/5/2019	3/5/2019	7	-5	-15	Invalid (return date)
12	AdminStaff	CD	15/5/2019	None	7	None	None	Invalid (none return date)
13	AdminStaff	CD	None	3/5/2019	7	None	None	Invalid (none borrow date)
14	AdminStaff	CD	None	27/5/2019	7	None	None	Invalid
15	AdminStaff	CD	None	3/5/2019	7	None	None	Invalid
16	AdminStaff	CD	None	None	7	None	None	Invalid
17	AdminStaff	DVD	1/5/2019	3/5/2019	7	0	0	Valid (no late return)
18	AdminStaff	DVD	5/5/2019	19/5/2019	7	7	21	Valid (late return)
19	AdminStaff	DVD	12/5/2019	3/5/2019	7	-2	-6	Invalid (return date)
20	AdminStaff	DVD	12/5/2019	None	7	None	None	Invalid (none return date)
21	AdminStaff	DVD	None	3/5/2019	7	None	None	Invalid (none borrow date)
22	AdminStaff	DVD	None	19/5/2019	7	None	None	Invalid
23	AdminStaff	DVD	None	3/5/2019	7	None	None	Invalid
24	AdminStaff	DVD	None	None	7	None	None	Invalid

ค1 ตารางกรณีทดสอบระบบจัดการห้องสมุด (ต่อ)

Test Case	Member	Item	borrowDate	returnDate	maxDays Borrow	daysOf LateReturn	Fine	Comments
25	AdminStaff	None	20/8/2019	25/8/2019	None	None	None	Invalid
26	AdminStaff	None	20/8/2019	30/8/2019	None	None	None	Invalid
27	AdminStaff	None	20/8/2019	10/8/2019	None	None	None	Invalid
28	AdminStaff	None	20/8/2019	None	None	None	None	Invalid
29	AdminStaff	None	None	25/8/2019	None	None	None	Invalid
30	AdminStaff	None	None	30/8/2019	None	None	None	Invalid
31	AdminStaff	None	None	10/8/2019	None	None	None	Invalid
32	AdminStaff	None	None	None	None	None	None	Invalid
33	Grad	Book	1/5/2019	10/5/2019	14	0	0	Valid (no late return)
34	Grad	Book	7/5/2019	5/6/2019	14	15	45	Valid (late return)
35	Grad	Book	25/5/2019	1/5/2019	14	-10	-30	Invalid (return date)
36	Grad	Book	25/5/2019	None	14	None	None	Invalid (none return date)
37	Grad	Book	None	10/5/2019	14	None	None	Invalid (none borrow date)
38	Grad	Book	None	5/6/2019	14	None	None	Invalid
39	Grad	Book	None	1/5/2019	14	None	None	Invalid

ค1 ตารางกรณีทดสอบระบบจัดการห้องสมุด (ต่อ)

Test Case	Member	Item	borrowDate	returnDate	maxDays Borrow	daysOf LateReturn	Fine	Comments
40	Grad	Book	None	None	14	None	None	Invalid
41	Grad	CD	1/5/2019	3/5/2019	7	2	0	Valid (no late return)
42	Grad	CD	10/5/2019	27/5/2019	7	10	30	Valid (late return)
43	Grad	CD	15/5/2019	3/5/2019	7	-5	-15	Invalid (return date)
44	Grad	CD	15/5/2019	None	7	None	None	Invalid (none return date)
45	Grad	CD	None	3/5/2019	7	None	None	Invalid (none borrow date)
46	Grad	CD	None	27/5/2019	7	None	None	Invalid
47	Grad	CD	None	3/5/2019	7	None	None	Invalid
48	Grad	CD	None	None	7	None	None	Invalid
49	Grad	DVD	1/5/2019	3/5/2019	7	0	0	Valid (no late return)
50	Grad	DVD	5/5/2019	19/5/2019	7	7	21	Valid (late return)
51	Grad	DVD	12/5/2019	3/5/2019	7	-2	-6	Invalid (return date)
52	Grad	DVD	12/5/2019	None	7	None	None	Invalid (none return date)
53	Grad	DVD	None	3/5/2019	7	None	None	Invalid (none borrow date)
54	Grad	DVD	None	19/5/2019	7	None	None	Invalid

ค1 ตารางกรณีทดสอบระบบจัดการห้องสมุด (ต่อ)

Test Case	Member	Item	borrowDate	returnDate	maxDays Borrow	daysOf LateReturn	Fine	Comments
55	Grad	DVD	None	3/5/2019	7	None	None	Invalid
56	Grad	DVD	None	None	7	None	None	Invalid
57	Grad	None	20/8/2019	25/8/2019	None	None	None	Invalid
58	Grad	None	20/8/2019	30/8/2019	None	None	None	Invalid
59	Grad	None	20/8/2019	10/8/2019	None	None	None	Invalid
60	Grad	None	20/8/2019	None	None	None	None	Invalid
61	Grad	None	None	25/8/2019	None	None	None	Invalid
62	Grad	None	None	30/8/2019	None	None	None	Invalid
63	Grad	None	None	10/8/2019	None	None	None	Invalid
64	Grad	None	None	None	None	None	None	Invalid
65	Lecturer	Book	1/8/2019	7/8/2019	30	0	0	Valid (no late return)
66	Lecturer	Book	1/7/2019	5/8/2019	30	5	15	Valid (late return)
67	Lecturer	Book	10/7/2019	9/6/2019	30	-1	-3	Invalid (return date)
68	Lecturer	Book	10/7/2019	None	30	None	None	Invalid (none return date)
69	Lecturer	Book	None	7/8/2019	30	None	None	Invalid (none borrow date)

ค1 ตารางกรณีทดสอบระบบจัดการห้องสมุด (ต่อ)

Test Case	Member	Item	borrowDate	returnDate	maxDays Borrow	daysOf LateReturn	Fine	Comments
70	Lecturer	Book	None	5/8/2019	30	None	None	Invalid
71	Lecturer	Book	None	9/6/2019	30	None	None	Invalid
72	Lecturer	Book	None	None	30	None	None	Invalid
73	Lecturer	CD	1/5/2019	3/5/2019	7	2	0	Valid (no late return)
74	Lecturer	CD	10/5/2019	27/5/2019	7	10	30	Valid (late return)
75	Lecturer	CD	15/5/2019	3/5/2019	7	-5	-15	Invalid (return date)
76	Lecturer	CD	15/5/2019	None	7	None	None	Invalid (none return date)
77	Lecturer	CD	None	3/5/2019	7	None	None	Invalid (none borrow date)
78	Lecturer	CD	None	27/5/2019	7	None	None	Invalid
79	Lecturer	CD	None	3/5/2019	7	None	None	Invalid
80	Lecturer	CD	None	None	7	None	None	Invalid
81	Lecturer	DVD	1/5/2019	3/5/2019	7	0	0	Valid (no late return)
82	Lecturer	DVD	5/5/2019	19/5/2019	7	7	21	Valid (late return)
83	Lecturer	DVD	12/5/2019	3/5/2019	7	-2	-6	Invalid (return date)
84	Lecturer	DVD	12/5/2019	None	7	None	None	Invalid (none return date)

ค1 ตารางกรณีทดสอบระบบจัดการห้องสมุด (ต่อ)

Test Case	Member	Item	borrowDate	returnDate	maxDays Borrow	daysOf LateReturn	Fine	Comments
85	Lecturer	DVD	None	3/5/2019	7	None	None	Invalid (none borrow date)
86	Lecturer	DVD	None	19/5/2019	7	None	None	Invalid
87	Lecturer	DVD	None	3/5/2019	7	None	None	Invalid
88	Lecturer	DVD	None	None	7	None	None	Invalid
89	Lecturer	None	20/8/2019	25/8/2019	None	None	None	Invalid
90	Lecturer	None	20/8/2019	30/8/2019	None	None	None	Invalid
91	Lecturer	None	20/8/2019	10/8/2019	None	None	None	Invalid
92	Lecturer	None	20/8/2019	None	None	None	None	Invalid
93	Lecturer	None	None	25/8/2019	None	None	None	Invalid
94	Lecturer	None	None	30/8/2019	None	None	None	Invalid
95	Lecturer	None	None	10/8/2019	None	None	None	Invalid
96	Lecturer	None	None	None	None	None	None	Invalid
97	UnderGrad	Book	1/7/2019	2/7/2019	7	0	0	Valid (no late return)
98	UnderGrad	Book	5/7/2019	15/7/2019	7	3	9	Valid (late return)
99	UnderGrad	Book	8/7/2019	30/6/2019	7	-1	-3	Invalid (return date)

ค1 ตารางกรณีทดสอบระบบจัดการห้องสมุด (ต่อ)

Test Case	Member	Item	borrowDate	returnDate	maxDays Borrow	daysOf LateReturn	Fine	Comments
100	UnderGrad	Book	8/7/2019	None	7	None	None	Invalid (none return date)
101	UnderGrad	Book	None	2/7/2019	7	None	None	Invalid (none borrow date)
102	UnderGrad	Book	None	15/7/2019	7	None	None	Invalid
103	UnderGrad	Book	None	30/6/2019	7	None	None	Invalid
104	UnderGrad	Book	None	None	7	None	None	Invalid
105	UnderGrad	CD	1/5/2019	3/5/2019	7	2	0	Valid (no late return)
106	UnderGrad	CD	10/5/2019	27/5/2019	7	10	30	Valid (late return)
107	UnderGrad	CD	15/5/2019	3/5/2019	7	-5	-15	Invalid (return date)
108	UnderGrad	CD	15/5/2019	None	7	None	None	Invalid (none return date)
109	UnderGrad	CD	None	3/5/2019	7	None	None	Invalid (none borrow date)
110	UnderGrad	CD	None	27/5/2019	7	None	None	Invalid
111	UnderGrad	CD	None	3/5/2019	7	None	None	Invalid
112	UnderGrad	CD	None	None	7	None	None	Invalid
113	UnderGrad	DVD	1/5/2019	3/5/2019	7	0	0	Valid (no late return)
114	UnderGrad	DVD	5/5/2019	19/5/2019	7	7	21	Valid (late return)

ค1 ตารางกรณีทดสอบระบบจัดการห้องสมุด (ต่อ)

Test Case	Member	Item	borrowDate	returnDate	maxDays Borrow	daysOf LateReturn	Fine	Comments
115	UnderGrad	DVD	12/5/2019	3/5/2019	7	-2	-6	Invalid (return date)
116	UnderGrad	DVD	12/5/2019	None	7	None	None	Invalid (none return date)
117	UnderGrad	DVD	None	3/5/2019	7	None	None	Invalid (none borrow date)
118	UnderGrad	DVD	None	19/5/2019	7	None	None	Invalid
119	UnderGrad	DVD	None	3/5/2019	7	None	None	Invalid
120	UnderGrad	DVD	None	None	7	None	None	Invalid
121	UnderGrad	None	20/8/2019	25/8/2019	None	None	None	Invalid
122	UnderGrad	None	20/8/2019	30/8/2019	None	None	None	Invalid
123	UnderGrad	None	20/8/2019	10/8/2019	None	None	None	Invalid
124	UnderGrad	None	20/8/2019	None	None	None	None	Invalid
125	UnderGrad	None	None	25/8/2019	None	None	None	Invalid
126	UnderGrad	None	None	30/8/2019	None	None	None	Invalid
127	UnderGrad	None	None	10/8/2019	None	None	None	Invalid
128	UnderGrad	None	None	None	None	None	None	Invalid
129	None	Book	1/7/2019	2/7/2019	None	None	None	Invalid

ค1 ตารางกรณีทดสอบระบบจัดการห้องสมุด (ต่อ)

Test Case	Member	Item	borrowDate	returnDate	maxDays Borrow	daysOf LateReturn	Fine	Comments
130	None	Book	5/7/2019	15/7/2019	None	None	None	Invalid
131	None	Book	8/7/2019	30/6/2019	None	None	None	Invalid
132	None	Book	8/7/2019	None	None	None	None	Invalid
133	None	Book	None	2/7/2019	None	None	None	Invalid
134	None	Book	None	15/7/2019	None	None	None	Invalid
135	None	Book	None	30/6/2019	None	None	None	Invalid
136	None	Book	None	None	None	None	None	Invalid
137	None	CD	1/5/2019	3/5/2019	None	None	None	Invalid
138	None	CD	10/5/2019	27/5/2019	None	None	None	Invalid
139	None	CD	15/5/2019	3/5/2019	None	None	None	Invalid
140	None	CD	15/5/2019	None	None	None	None	Invalid
141	None	CD	None	3/5/2019	None	None	None	Invalid
142	None	CD	None	27/5/2019	None	None	None	Invalid
143	None	CD	None	3/5/2019	None	None	None	Invalid
144	None	CD	None	None	None	None	None	Invalid

ค1 ตารางกรณีทดสอบระบบจัดการห้องสมุด (ต่อ)

Test Case	Member	Item	borrowDate	returnDate	maxDays Borrow	daysOf LateReturn	Fine	Comments
145	None	DVD	1/5/2019	3/5/2019	None	None	None	Invalid
146	None	DVD	5/5/2019	19/5/2019	None	None	None	Invalid
147	None	DVD	12/5/2019	3/5/2019	None	None	None	Invalid
148	None	DVD	12/5/2019	None	None	None	None	Invalid
149	None	DVD	None	3/5/2019	None	None	None	Invalid
150	None	DVD	None	19/5/2019	None	None	None	Invalid
151	None	DVD	None	3/5/2019	None	None	None	Invalid
152	None	DVD	None	None	None	None	None	Invalid
153	None	None	20/8/2019	25/8/2019	None	None	None	Invalid
154	None	None	20/8/2019	30/8/2019	None	None	None	Invalid
155	None	None	20/8/2019	10/8/2019	None	None	None	Invalid
156	None	None	20/8/2019	None	None	None	None	Invalid
157	None	None	None	25/8/2019	None	None	None	Invalid
158	None	None	None	30/8/2019	None	None	None	Invalid
159	None	None	None	10/8/2019	None	None	None	Invalid
160	None	None	None	None	None	None	None	Invalid

ค2 ตารางกรณีทดสอบระบบวิเคราะห์คำนวณค่าไต

Test Case	Sex	Age	Height	SCr	GFR	UO	Stage	Comments
1	Female	10	142	0.8	73	450	Injury	Valid
2	Female	18	165	4.5	15	27	Loss	Valid
3	Female	2	74	-25.1	-1	-15	None	Invalid
4	Female	6	115	16.5	2	25	None	Invalid
5	Female	10	-142	0.8	-73	-450	None	Invalid
6	Female	18	-165	4.5	-15	-27	None	Invalid
7	Female	2	-74	-25.1	1	15	None	Invalid
8	Female	6	-115	16.5	-2	-25	None	Invalid
9	Female	5	350	0.8	180	450	None	Invalid
10	Female	7	658	4.5	60	1420	None	Invalid
11	Female	14	980	-25.1	-16	-15	None	Invalid
12	Female	17	745	16.5	18	25	None	Invalid
13	Female	44	155	0.1	1511	655	Risk	Valid
14	Female	85	169	1.2	50	196	Failure	Valid
15	Female	85	172	-1.2	-50	-196	None	Invalid

ค2 ตารางกรณีทดสอบระบบวิเคราะห์คำนวณค่าไต (ต่อ)

Test Case	Sex	Age	Height	SCr	GFR	UO	Stage	Comments
16	Female	60	148	160.5	66830	2541	None	Invalid
17	Female	44	-155	0.1	1511	655	Risk	Invalid
18	Female	85	-169	1.2	50	196	Failure	Invalid
19	Female	85	-172	-1.2	-50	-196	None	Invalid
20	Female	60	-148	160.5	66830	2541	None	Invalid
21	Female	44	32547	0.1	1511	655	None	Invalid
22	Female	85	6574	1.2	50	196	None	Invalid
23	Female	85	840	-1.2	-50	-196	None	Invalid
24	Female	60	399	160.5	66830	2541	None	Invalid
25	Female	-2544	142	0.8	73	450	None	Invalid
26	Female	-2544	165	4.5	15	27	None	Invalid
27	Female	-2544	74	-25.1	-1	-15	None	Invalid
28	Female	-2544	115	16.5	2	25	None	Invalid
29	Female	-2544	-142	0.8	-73	-450	None	Invalid
30	Female	-2544	-165	4.5	-15	-27	None	Invalid

ค2 ตารางกรณีทดสอบระบบวิเคราะห์คำนวณค่าไต (ต่อ)

Test Case	Sex	Age	Height	SCr	GFR	UO	Stage	Comments
31	Female	-2544	-74	-25.1	1	15	None	Invalid
32	Female	-2544	-115	16.5	-2	-25	None	Invalid
33	Female	-2544	350	0.8	180	450	None	Invalid
34	Female	-2544	658	4.5	60	1420	None	Invalid
35	Female	-2544	980	-25.1	-16	-15	None	Invalid
36	Female	-2544	745	16.5	18	25	None	Invalid
37	Female	500	142	0.8	4	12	None	Invalid
38	Female	500	165	4.5	40	68	None	Invalid
39	Female	500	74	-4.5	-40	-68	None	Invalid
40	Female	500	115	16.5	197	1420	None	Invalid
41	Female	500	-142	0.8	4	12	None	Invalid
42	Female	500	-165	4.5	40	68	None	Invalid
43	Female	500	-74	-4.5	-40	-68	None	Invalid
44	Female	500	-115	16.5	197	1420	None	Invalid
45	Female	500	350	0.8	4	12	None	Invalid

ค2 ตารางกรณีทดสอบระบบวิเคราะห์คำนวณค่าไต (ต่อ)

Test Case	Sex	Age	Height	SCr	GFR	UO	Stage	Comments
46	Female	500	658	4.5	40	68	None	Invalid
47	Female	500	980	-4.5	-40	-68	None	Invalid
48	Female	500	745	16.5	197	1420	None	Invalid
49	Male	2	76	0.5	62	580	Injury	Valid
50	Male	7	124	7.8	6	15	ESRD	Valid
51	Male	14	159	-1.2	-54	-35	None	Invalid
52	Male	17	190	15.3	5	3	None	Invalid
53	Male	2	-76	0.5	-62	-580	None	Invalid
54	Male	7	-124	7.8	-6	-15	None	Invalid
55	Male	14	-159	-1.2	54	35	None	Invalid
56	Male	17	-190	15.3	-5	-3	None	Invalid
57	Male	2	365	0.5	301	580	None	Invalid
58	Male	7	1580	7.8	83	301	None	Invalid
59	Male	14	784	-1.2	-269	-5958	None	Invalid
60	Male	17	1140	15.3	30	248	None	Invalid

ค2 ตารางกรณีทดสอบระบบวิเคราะห์คำนวณค่าไต (ต่อ)

Test Case	Sex	Age	Height	SCr	GFR	UO	Stage	Comments
61	Male	29	177	0.3	356	555	Risk	Valid
62	Male	65	104	7.2	7	10	ESRD	Valid
63	Male	112	140	-2.7	-16	-6	None	Invalid
64	Male	67	194	26.6	5394	10245	None	Invalid
65	Male	29	-138	0.3	356	555	None	Invalid
66	Male	112	-153	2.7	16	6	None	Invalid
67	Male	112	-177	-2.7	-16	-6	None	Invalid
68	Male	67	-149	26.6	5394	10245	None	Invalid
69	Male	29	365	0.3	356	555	None	Invalid
70	Male	112	1580	2.7	16	6	None	Invalid
71	Male	112	784	-2.7	-16	-6	None	Invalid
72	Male	67	1140	26.6	5394	10245	None	Invalid
73	Male	-29	76	0.5	62	580	None	Invalid
74	Male	-29	124	7.8	6	15	None	Invalid
75	Male	-29	159	-1.2	-54	-35	None	Invalid

ค2 ตารางกรณีทดสอบระบบวิเคราะห์คำนวณค่าไต (ต่อ)

Test Case	Sex	Age	Height	SCr	GFR	UO	Stage	Comments
76	Male	-29	190	15.3	5	3	None	Invalid
77	Male	-29	-76	0.5	-62	-580	None	Invalid
78	Male	-29	-124	7.8	-6	-15	None	Invalid
79	Male	-29	-159	-1.2	54	35	None	Invalid
80	Male	-29	-190	15.3	-5	-3	None	Invalid
81	Male	-29	365	0.5	301	580	None	Invalid
82	Male	-29	1580	7.8	83	301	None	Invalid
83	Male	-29	784	-1.2	-269	-5958	None	Invalid
84	Male	-29	1140	15.3	30	248	None	Invalid
85	Male	200	138	0.4	24	16	None	Invalid
86	Male	200	153	7.2	426	487	None	Invalid
87	Male	200	177	-7.2	-426	-487	None	Invalid
88	Male	200	149	110.2	11558	65487	None	Invalid
89	Male	200	-138	0.4	24	16	None	Invalid
90	Male	200	-153	7.2	426	487	None	Invalid

ค2 ตารางกรณีทดสอบระบบวิเคราะห์คำนวณค่าไต (ต่อ)

Test Case	Sex	Age	Height	SCr	GFR	UO	Stage	Comments
91	Male	200	-177	-7.2	-426	-487	None	Invalid
92	Male	200	-149	110.2	11558	65487	None	Invalid
93	Male	200	365	0.4	24	16	None	Invalid
94	Male	200	1580	7.2	426	487	None	Invalid
95	Male	200	784	-7.2	-426	-487	None	Invalid
96	Male	200	1140	110.2	11558	65487	None	Invalid
97	None	10	142	0.8	73	450	None	Invalid
98	None	10	142	0.8	73	450	None	Invalid
99	None	18	165	4.5	15	27	None	Invalid
100	None	2	74	-25.1	-1	-15	None	Invalid
101	None	6	115	16.5	2	25	None	Invalid
102	None	10	-142	0.8	-73	-450	None	Invalid
103	None	18	-165	4.5	-15	-27	None	Invalid
104	None	2	-74	-25.1	1	15	None	Invalid
105	None	6	-115	16.5	-2	-25	None	Invalid

ค2 ตารางกรณีทดสอบระบบวิเคราะห์คำนวณค่าไต (ต่อ)

Test Case	Sex	Age	Height	SCr	GFR	UO	Stage	Comments
106	None	5	350	0.8	180	450	None	Invalid
107	None	7	658	4.5	60	1420	None	Invalid
108	None	14	980	-25.1	-16	-15	None	Invalid
109	None	44	155	0.1	1511	655	None	Invalid
110	None	85	169	1.2	50	196	None	Invalid
111	None	85	172	-1.2	-50	-196	None	Invalid
112	None	60	148	160.5	66830	2541	None	Invalid
113	None	44	-155	0.1	1511	655	None	Invalid
114	None	85	-169	1.2	50	196	None	Invalid
115	None	85	-172	-1.2	-50	-196	None	Invalid
116	None	60	-148	160.5	66830	2541	None	Invalid
117	None	44	32547	0.1	1511	655	None	Invalid
118	None	85	6574	1.2	50	196	None	Invalid
119	None	85	840	-1.2	-50	-196	None	Invalid
120	None	60	399	160.5	66830	2541	None	Invalid

ค2 ตารางกรณีทดสอบระบบวิเคราะห์คำนวณค่าไต (ต่อ)

Test Case	Sex	Age	Height	SCr	GFR	UO	Stage	Comments
121	None	-29	76	0.5	62	580	None	Invalid
122	None	-29	124	7.8	6	15	None	Invalid
123	None	-29	159	-1.2	-54	-35	None	Invalid
124	None	-29	190	15.3	5	3	None	Invalid
125	None	-29	-76	0.5	-62	-580	None	Invalid
126	None	-29	-124	7.8	-6	-15	None	Invalid
127	None	-29	-159	-1.2	54	35	None	Invalid
128	None	-29	-190	15.3	-5	-3	None	Invalid
129	None	-29	365	0.5	301	580	None	Invalid
130	None	-29	1580	7.8	83	301	None	Invalid
131	None	-29	784	-1.2	-269	-5958	None	Invalid
132	None	-29	1140	15.3	30	248	None	Invalid
133	None	200	138	0.4	24	16	None	Invalid
134	None	200	153	7.2	426	487	None	Invalid
135	None	200	177	-7.2	-426	-487	None	Invalid

ค2 ตารางกรณีทดสอบระบบวิเคราะห์คำนวณค่าไต (ต่อ)

Test Case	Sex	Age	Height	SCr	GFR	UO	Stage	Comments
136	None	200	149	110.2	11558	65487	None	Invalid
137	None	200	-138	0.4	24	16	None	Invalid
138	None	200	-153	7.2	426	487	None	Invalid
139	None	200	-177	-7.2	-426	-487	None	Invalid
140	None	200	-149	110.2	11558	65487	None	Invalid
141	None	200	365	0.4	24	16	None	Invalid
142	None	200	1580	7.2	426	487	None	Invalid
143	None	200	784	-7.2	-426	-487	None	Invalid
144	None	200	1140	110.2	11558	65487	None	Invalid

ประวัติผู้เขียน

ชื่อ-สกุล นางสาวสุไรยา เจริญเระ
รหัสประจำตัวนักศึกษา 6030223005
วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วิทยาศาสตรมหาบัณฑิต (เทคโนโลยีสารสนเทศ)	มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต	2565

ทุนการศึกษา (ที่ได้รับในระหว่างการศึกษา)

ทุนอุดหนุนการวิจัยจากกองทุนวิจัยวิทยาลัยการคอมพิวเตอร์

การตีพิมพ์เผยแพร่ผลงาน

สุไรยา เจริญเระ, และ อดิศักดิ์ อินทนา. (2562). “Enhancing Software Testing with Ontology Engineering Approach.”, The 23rd International Computer Science and Engineering Conference 2019, Novotel Phuket Resort ภูเก็ต: 30 ตุลาคม-1 พฤศจิกายน 2562.