การสร้างต้นแบบลักษณะทดสอบได้โดยอัตโนมัติจากข้อกำหนดความต้องการที่ถูกเก็บเกี่ยว

นายนาวิน เฟื่องฟู

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2556

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)
are the thesis authors' files submitted through the University Graduate School.

# AUTOMATED TESTING FEATURING PROTOTYPE GENERATION FROM HARVESTED REQUIREMENTS SPECIFICATION

Mr. Nawin Phuangphoo

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Computer Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2013

Copyright of Chulalongkorn University

| | |
|---|---|
| หัวข้อวิทยานิพนธ์ | การสร้างต้นแบบลักษณะทดสอบได้โดยอัตโนมัติจากข้อกำหนดความต้องการที่ถูกเก็บเกี่ยว |
| โดย | นายนาวิน เฟื่องฟู |
| สาขาวิชา | วิศวกรรมคอมพิวเตอร์ |
| อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก | ศาสตราจารย์ ดร. บุญเสริม กิจศิริกุล |

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร. บัณฑิต เอื้ออาภรณ์)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. สุกรี สินธุภิญโญ)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ศาสตราจารย์ ดร. บุญเสริม กิจศิริกุล)

.....กรรมการภายนอกมหาวิทยาลัย

(อาจารย์ ดร. ภาสกร อภิรักษ์วรพินิต)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

นาวิน เฟื่องฟู : การสร้างต้นแบบลักษณะทดสอบได้โดยอัตโนมัติจากข้อกำหนดความต้องการที่ถูกเก็บเกี่ยว. (AUTOMATED TESTING FEATURING PROTOTYPE GENERATION FROM HARVESTED REQUIREMENTS SPECIFICATION) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ศ. ดร. บุญเสริม กิจศิริกุล, 64 หน้า.

ต้นแบบเป็นวิธีการโดยทั่วไปที่แนะนำสำหรับการยันสอบใช้งานได้จริงของความต้องการระหว่างขั้นต้นๆ ของโครงการซอฟต์แวร์ อย่างไรก็ตาม การสร้างต้นแบบด้วยบุคคลนั้นใช้ทรัพยากรด้านต่างๆ อยู่มาก นอกจากนั้นต้นแบบที่พัฒนาได้ตามการเปลี่ยนแปลงความต้องการอาจก่อให้เกิดความไม่ถูกต้องตรงกันระหว่างสิ่งต่างๆ ที่เกี่ยวข้อง งานวิจัยนี้จึงได้นำเสนอวิธีการเพื่อสร้างต้นแบบลักษณะทดสอบได้โดยอัตโนมัติจากข้อกำหนดความต้องการที่ถูกเก็บเกี่ยว การพัฒนาระบบเป็นส่วนหนึ่งของกระบวนการเก็บเกี่ยวความต้องการ ซึ่งเป็นระบบบูรณาการเพื่ออำนวยความสะดวกต่อกิจกรรมการควบคุมคุณภาพที่มีความสามารถในการจัดการความต้องการที่จัดเตรียมมาจากเมทริกซ์ตามรอย โดยต้นแบบที่สร้างขึ้นนั้นสามารถต่อยอดนำไปสร้างเป็นตัวระบบจริงได้ สามารถทดสอบตนเองได้และสามารถปรับปรุงเพื่อรักษาความสัมพันธ์ระหว่างชิ้นงานที่เกี่ยวข้อง วิธีการดังกล่าวสนับสนุนสภาพการทำงานแบบชาญฉลาด ซึ่งช่วยพัฒนาขีดความสามารถและสมรรถภาพของกระบวนการ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก .....

ปีการศึกษา 2556

# # 5470247021 : MAJOR COMPUTER ENGINEERING
KEYWORDS: PROTOTYPE / USER INTERFACE TESTING / REQUIREMENTS SPECIFICATION / CONTENT HARVESTER / PROCESS IMPROVEMENT

NAWIN PHUANGPHOO: AUTOMATED TESTING FEATURING PROTOTYPE GENERATION FROM HARVESTED REQUIREMENTS SPECIFICATION. ADVISOR: PROF. BOONSERM KIJSIRIKUL, Dr., 64 pp.

Prototyping is a common technique suggested for requirements validation during the early phase of software project. However, manually develop a prototype is resource consuming. In addition, the evolving prototype due to requirements changes may cause inconsistencies among associated artifacts. This paper thus presents an approach to automated testing featuring prototype generation from harvested requirements specification. The implementation is part of Requirements Harvester— RH, which is a system to facilitate integrating quality control activities with requirements management ability provided by a traceability matrix. The prototypes can be incrementally built into the full blown system. They are self-test and capable of maintaining the consistency of related work products. The proposed automation approach promotes the working smarter environments that could improve process capability and performance.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Department: Computer Engineering Student's Signature .....
Field of Study: Computer Engineering Advisor's Signature .....
Academic Year: 2013

# กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความอนุเคราะห์จากศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล อาจารย์ที่ปรึกษาวิทยานิพนธ์และรองศาสตราจารย์ ดร.ญาใจ ลิ่มปิยะกรณ์ ได้สละเวลาให้ความรู้ คำปรึกษา ตรวจสอบและแก้ไขข้อผิดพลาดและปัญหาต่างๆ ตลอดจนการกำกับดูแลและคอยติดตามความก้าวหน้า ทำให้การวิจัยนี้สำเร็จไปได้ด้วยดี ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ และ ดร.ภาสกร อภิรักษ์วรพินิต กรรมการสอบวิทยานิพนธ์ ที่กรุณาเสียสละเวลา ให้คำแนะนำ ตรวจสอบ และแก้ไขวิทยานิพนธ์ฉบับนี้

ขอขอบพระคุณบิดา มารดา และญาติพี่น้องที่ให้การสนับสนุนและเป็นกำลังใจที่ดีให้เสมอมาและสนับสนุนด้านทุนทรัพย์ในการศึกษารวมไปถึงทุกท่านที่มีส่วนช่วยเหลือในการทำวิทยานิพนธ์ครั้งนี้ ซึ่งมิได้กล่าวนามในที่นี้

ท้ายที่สุด ผู้วิจัยขอขอบพระคุณเพื่อนๆ ทุกคน ที่คอยติดตามและให้กำลังใจ รวมถึงท่านอื่นๆ ที่มิได้กล่าวลงนามไว้ ณ ที่นี้ที่มีส่วนทำให้วิทยานิพนธ์สำเร็จลุล่วงไปได้ด้วยดีผู้วิจัยหวังเป็นอย่างยิ่งว่าวิทยานิพนธ์ฉบับนี้จะเป็นประโยชน์บ้างไม่มากก็น้อยสำหรับผู้ที่สนใจจะศึกษารายละเอียดต่อไป

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

**สารบัญ**

หน้า


บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ญ

สารบัญภาพ ..... ฎ

บทที่ 1 บทนำ ..... 1

1.1 ความเป็นมาและความสำคัญของปัญหา ..... 1

1.2 วัตถุประสงค์ของการวิจัย ..... 2

1.3 ขอบเขตงานวิจัย ..... 2

1.4 ประโยชน์ที่คาดว่าจะได้รับ ..... 2

1.5 วิธีดำเนินการวิจัย ..... 2

1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์ ..... 3

1.7 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์ ..... 3

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง ..... 4

2.1 ทฤษฎีที่เกี่ยวข้อง ..... 4

2.1.1 การเก็บเกี่ยวเนื้อหา (Content Harvester) [2] ..... 4

2.1.2 การทำต้นแบบ (Prototyping) ..... 5

2.1.2.1 การพัฒนาต้นแบบ ..... 5

2.1.2.2 ประโยชน์ของการสร้างต้นแบบอัตโนมัติ ..... 5

2.1.3 นิพจน์ปรกติ ..... 6

2.2 งานวิจัยที่เกี่ยวข้อง ..... 8

2.2.1 A Method to Automate User Interface Testing Using Variable Finite State Machines [6] ..... 8

2.2.2 Automated Prototyping of User Interfaces Based on UML Scenarios [7] ..... 9

บทที่ 3 การออกแบบระบบ ..... 10

3.1 ส่วนนำเข้าและสกัดข้อมูล ..... 12

หน้า

3.1.1 ข้อกำหนดการติดป้าย (Tagging Specification) ..... 12
3.1.2 ประเภทข้อมูลเข้า ..... 13
3.2 ส่วนการสร้างต้นแบบ ..... 16
3.2.1 ระบุองค์ประกอบส่วนติดต่อผู้ใช้คุณสมบัติ และการส่งผ่านของแต่ละหน้าจอ ..... 16
3.2.2 สร้างส่วนติดต่อผู้ใช้ ..... 19
3.2.3 จับคู่องค์ประกอบส่วนติดต่อผู้ใช้และรายละเอียดการควบคุมการเปลี่ยนหน้าจอ ..... 21
3.3 ส่วนการสร้างคำสั่งทดสอบ (Test Script) ..... 22
3.3.1 การท่องกราฟ ..... 24
3.3.2 การเชื่อมโยงเข้ากับทฤษฎีเครื่องจักรสถานะจำกัดซึ่งเปลี่ยนแปลงได้ ..... 26
3.3.2.1 กรณีสถานะแบบต่างๆ ของเครื่องจักร ..... 27
3.4 ส่วนการทดสอบอัตโนมัติ ..... 33
3.4.1 วิธีการสร้างเหตุการณ์ข้อมูลเข้าแก่ระบบแต่กำเนิด (Native System Input Events) ..... 34
3.5 ส่วนการปรับปรุงเอกสาร ..... 37
3.6 ส่วนการปรับปรุงต้นแบบ ..... 38
บทที่ 4 การพัฒนาระบบ ..... 40
4.1 สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนา ..... 40
4.1.1 ฮาร์ดแวร์ ..... 40
4.1.2 ซอฟต์แวร์ ..... 40
4.1.3 ขั้นตอนการติดตั้งซอฟต์แวร์ ..... 40
4.2 ขั้นตอนการพัฒนาระบบ ..... 41
4.2.1 การพัฒนาขั้นตอนวิธีการเตรียมข้อมูลเพื่อนำเข้า ..... 41
4.2.2 การแก้ไขปรับปรุงข้อมูลต้นแบบ ..... 41
4.2.3 การกำหนดความสัมพันธ์ ..... 42
4.2.4 การสร้างต้นแบบ ..... 42
4.2.5 การทดสอบต้นแบบ ..... 42
บทที่ 5 การทดสอบระบบ ..... 44
5.1 การทดสอบความถูกต้องของฟังก์ชันการทำงานของระบบ ..... 44

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

หน้า

5.2 การทดสอบความถูกต้องของกระบวนการทำงานของระบบ ..... 47
5.3 การประเมินผลระบบสร้างต้นแบบแบบอัตโนมัติ ..... 49
5.4 การประเมินผลระบบทดสอบต้นแบบแบบอัตโนมัติ ..... 51
5.5 สรุปผลการทดลอง ..... 54
บทที่ 6 สรุปผลการวิจัย และข้อเสนอแนะ ..... 55
6.1 สรุปผลการวิจัย ..... 55
6.2 ข้อจำกัด ..... 55
6.3 แนวทางสำหรับการวิจัยต่อในอนาคต ..... 56
รายการอ้างอิง ..... 57
ภาคผนวก ก ตัวอย่างแสดงการทำงานกรณีทดสอบ ..... 59
ประวัติผู้เขียนวิทยานิพนธ์ ..... 64

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# สารบัญตาราง

หน้า

ตารางที่ 1 รายชื่อสัญลักษณ์ทั่วไปของนิพจน์ปรกติ ..... 7
ตารางที่ 2 ตัวอย่างรูปแบบของข้อความที่สามารถนำมาใช้เป็นป้ายบอกตำแหน่ง ..... 12
ตารางที่ 3 ข้อมูลเกี่ยวกับองค์ประกอบหรือเขตข้อมูล ..... 13
ตารางที่ 4 ข้อมูลเกี่ยวกับเหตุการณ์/การกระทำ ..... 14
ตารางที่ 5 รายการสำคัญของเอกสารของไมโครซอฟต์เวิร์ด ..... 15
ตารางที่ 6 รายการสำคัญขององค์ประกอบในหน้าการเข้าสู่ระบบ ..... 16
ตารางที่ 7 ข้อมูลขององค์ประกอบต่างๆ รวมไปถึงส่วนของรายละเอียดเพิ่มเติม ..... 17
ตารางที่ 8 การเทียบเคียงหลักการของเครื่องจักรสถานะจำกัดซึ่งเปลี่ยนแปลงได้ ..... 22
ตารางที่ 9 ตัวอย่างการดำเนินการโดยสากลทั่วไปกับแต่ละองค์ประกอบ ..... 35
ตารางที่ 10 ส่วนพื้นฐานต่างๆ ของคลาสในภาษาจาวา ..... 38
ตารางที่ 11 ผลการทดสอบความถูกต้องของฟังก์ชันการทำงานของระบบ ..... 44
ตารางที่ 12 ผลการทดสอบความถูกต้องของกระบวนการทำงานของระบบ ..... 47
ตารางที่ 13 เหตุการณ์ที่ใช้ในการเปรียบเทียบวัดจับเวลาระหว่างการสร้างต้นแบบด้วยมือเพียงอย่างเดียวกับด้วยระบบ จากเอกสารครั้งแรก ..... 49
ตารางที่ 14 เหตุการณ์ที่ใช้ในการเปรียบเทียบวัดจับเวลาระหว่างการสร้างต้นแบบด้วยมือเพียงอย่างเดียวกับด้วยระบบ เมื่อเอกสารเกิดการเปลี่ยนแปลง ..... 49
ตารางที่ 15 เปรียบเทียบเวลา(นาที)ที่ใช้การสร้างต้นแบบด้วยมือเพียงอย่างเดียวกับด้วยระบบในการเปลี่ยนแปลงแต่ละครั้ง ..... 50
ตารางที่ 16 เหตุการณ์ที่ใช้ในการเปรียบเทียบวัดจับเวลาระหว่างการทดสอบต้นแบบด้วยมือเพียงอย่างเดียวกับด้วยระบบ จากเอกสารครั้งแรก ..... 52
ตารางที่ 17 เหตุการณ์ที่ใช้ในการเปรียบเทียบวัดจับเวลาระหว่างการทดสอบต้นแบบด้วยมือเพียงอย่างเดียวกับด้วยระบบ เมื่อเอกสารเกิดการเปลี่ยนแปลง ..... 52
ตารางที่ 18 เปรียบเทียบเวลา(นาที)ที่ใช้การทดสอบต้นแบบด้วยมือเพียงอย่างเดียวกับด้วยระบบในการเปลี่ยนแปลงแต่ละครั้ง ..... 52

# สารบัญภาพ

หน้า

ภาพที่ 1 กระบวนวิธีเก็บเกี่ยวเนื้อหา ..... 4
ภาพที่ 2 เปรียบเทียบผลลัพธ์ของกลยุทธ์ “การทำงานให้หนักขึ้น” และ “การทำงานให้ฉลาดขึ้น” ..... 6
ภาพที่ 3 สถาปัตยกรรมระบบ Requirements Harvester ..... 11
ภาพที่ 4 กระบวนการสร้างและทดสอบต้นแบบในงานวิจัย ..... 11
ภาพที่ 5 Document Object Model ของตารางในโปรแกรมไมโครซอฟต์เวิร์ด ..... 15
ภาพที่ 6 ตัวอย่างตารางรายละเอียดองค์ประกอบและเหตุการณ์/การกระทำ ..... 19
ภาพที่ 7 ตัวอย่างภาพหน้าจอเข้าสู่ระบบในเอกสาร ซึ่งได้มาจากการบันทึกหน้าจอของต้นแบบที่ทำด้วยมือ ..... 19
ภาพที่ 8 ตัวอย่างภาพหน้าจอเข้าสู่ระบบที่สร้างขึ้นมาจากระบบ ด้วยข้อมูลที่สกัดมาจากเอกสารดังภาพที่ 6 ..... 20
ภาพที่ 9 ตัวอย่างภาพหน้าจอการบันทึกข้อมูลบริษัทในเอกสาร ซึ่งได้มาจากการบันทึกหน้าจอของต้นแบบที่ทำด้วยมือ ..... 20
ภาพที่ 10 ตัวอย่างภาพหน้าจอการบันทึกข้อมูลบริษัทที่สร้างขึ้นมาจากระบบ ด้วยข้อมูลที่สกัดมาจากเอกสารดังภาพที่ 6 ..... 21
ภาพที่ 11 แผนภาพตัวอย่างสถานะของส่วนติดต่อผู้ใช้ตามภาพที่ 8 ..... 23
ภาพที่ 12 เครื่องจากสถานะกรณีตัวอย่างที่ 1 ..... 27
ภาพที่ 13 เครื่องจากสถานะกรณีตัวอย่างที่ 2 ..... 28
ภาพที่ 14 เครื่องจากสถานะกรณีตัวอย่างที่ 3 ..... 29
ภาพที่ 15 เครื่องจากสถานะกรณีตัวอย่างที่ 4 ..... 29
ภาพที่ 16 กราฟตัวอย่างแสดงเส้นทางทดสอบที่สัมพันธ์กับภาพที่ 11 ..... 31
ภาพที่ 17 ตัวอย่างรูปแบบแบบข้อความที่แสดงถึงกราฟและเส้นทางการทดสอบ ..... 33
ภาพที่ 18 ขั้นตอนการปรับปรุงเอกสารข้อกำหนดความต้องการเพื่อความสอดคล้องกับข้อมูลที่เปลี่ยนแปลงไปเนื่องมาจากการแก้ไขต้นแบบ ..... 37
ภาพที่ 19 หน้าจอการระบุกฎการนำเข้า ..... 41
ภาพที่ 20 หน้าจอการแก้ไขปรับปรุงความสัมพันธ์ ..... 42
ภาพที่ 21 ซอสโค้ดจากต้นแบบที่ระบบสร้างขึ้นในส่วนการตรวจสอบเขตข้อมูลต่างๆ ..... 43
ภาพที่ 22 กราฟความแตกต่างของเวลาที่ใช้ระหว่างการสร้างต้นแบบด้วยมือกับการสร้างต้นแบบอัตโนมัติ ..... 50

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

หน้า

ภาพที่ 23 กราฟแสดงเปอร์เซนต์ความแตกต่างของเวลากับจำนวนครั้งของการสร้างและปรับปรุงต้นแบบระหว่างวิธีการทำด้วยมือกับวิธีการโดยอัตโนมัติ ..... 51
ภาพที่ 24 กราฟความแตกต่างของเวลาที่ใช้ระหว่างการทดสอบต้นแบบด้วยมือกับการทดสอบต้นแบบอัตโนมัติ ..... 53
ภาพที่ 25 กราฟแสดงเปอร์เซนต์ความแตกต่างของเวลากับจำนวนครั้งของการทดสอบต้นแบบระหว่างวิธีการทำด้วยมือกับวิธีการโดยอัตโนมัติ ..... 53
ภาพที่ 26 เมนูหลักและเมนูย่อยการนำเข้าเอกสารความต้องการ ..... 59
ภาพที่ 27 กล่องเลือกข้อกำหนดความต้องการที่จะนำเข้า ..... 60
ภาพที่ 28 ตัวอย่างการปรับปรุงค่าของต้นแบบในหน้าจอเข้าสู่ระบบ ..... 61
ภาพที่ 29 กราฟตัวอย่างแสดงกราฟความสัมพันธ์ ..... 62
ภาพที่ 30 กล่องกำหนดประเภทข้อมูล ..... 63

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาและความสำคัญของปัญหา

ความผิดพลาดของการพัฒนาความต้องการ (Requirements Development) เป็นสาเหตุหลักประการหนึ่งของความล้มเหลวของโครงการซอฟต์แวร์ ระเบียบวิธีการพัฒนาซอฟต์แวร์ปัจจุบันจึงให้ความสำคัญกับการทวนสอบ (Verification) และการยันสอบใช้งานได้จริง (Validation) เพื่อตรวจจับและแก้ไขข้อบกพร่องของข้อกำหนดความต้องการซอฟต์แวร์ตั้งแต่ระยะแรกๆ ของโครงการ เทคนิคหนึ่งที่นิยมใช้กันมาก คือ การทำต้นแบบ (Prototyping) เพื่อให้ลูกค้าและทีมผู้พัฒนาระบบมีความเข้าใจถึงระบบซอฟต์แวร์ที่กำลังจะพัฒนาได้ถูกต้องตรงกัน

การปรับปรุงกระบวนการ (Process Improvement) จึงเข้ามามีบทบาทสำคัญต่อกระบวนการพัฒนาซอฟต์แวร์ในปัจจุบัน โดยเป็นที่ยอมรับกันว่าการทำงานอย่างชาญฉลาดมากขึ้น (Working Smarter) เป็นวิธีการที่มีประสิทธิภาพและประสิทธิผลกว่าการทำงานอย่างหนักมากขึ้น (Working Harder) [1] ความเป็นอัตโนมัติ (Automation) คือ ปัจจัยหนึ่งที่เป็นที่ต้องการอย่างสูงในอุตสาหกรรมพัฒนาซอฟต์แวร์ เนื่องจากกระบวนการอัตโนมัติช่วยอำนวยความสะดวก ลดความผิดพลาด ลดระยะเวลาและต้นทุนที่ต้องใช้ในการทำงานและการบริหารงาน ทั้งยังช่วยให้การทวนสอบและการยันสอบใช้งานได้จริงกระทำได้ง่ายรวดเร็วยิ่งขึ้น

ปัจจุบัน มีเครื่องมือช่วยการสร้างระบบต้นแบบซอฟต์แวร์ให้สะดวกง่ายดายขึ้น เช่น Screen Generator ซึ่งมีความสามารถในการลากและวางส่วนประกอบต่างๆ ตามที่ผู้ใช้ต้องการ และ Modeling User Interfaces ซึ่งให้ผู้ใช้สร้างโมเดลตามรูปแบบที่กำหนดจากนั้นจึงนำข้อมูลนั้นมาสร้างเป็นต้นแบบซอฟต์แวร์ เป็นต้น อย่างไรก็ตาม เครื่องมือเหล่านั้นสร้างระบบต้นแบบตามที่ผู้ใช้กำหนดขึ้นเท่านั้น แต่ไม่ได้มีความเชื่อมโยงหรือสร้างจากข้อกำหนดความต้องการ (Requirements Specification) โดยตรง

ดังนั้น ผู้วิจัยจึงมีแนวคิดเพื่อนำเสนอวิธีการอัตโนมัติในการสร้างต้นแบบโดยตรงจากข้อกำหนดความต้องการที่อยู่ในรูปแบบซึ่งสามารถเก็บเกี่ยวเนื้อหาได้ ในที่นี้หมายถึง รูปแบบ XML รวมทั้งนำเสนอวิธีการสนับสนุนการทดสอบอัตโนมัติเพื่อสาธิตการยันสอบใช้ได้จริงของระบบต้นแบบกับผู้ใช้ การเก็บเกี่ยวเนื้อหาจากแหล่งข้อมูลที่อยู่ในรูปแบบ XML มีข้อดีคือ การแลกเปลี่ยนข้อมูลระหว่างแอปพลิเคชันหรือแพลทฟอร์มสามารถกระทำได้ง่าย และ XMLเป็นภาษาที่โปรแกรมประมวลผลคำ (Word Processor) นิยมใช้อย่างแพร่หลาย ตัวอย่างเช่น โปรแกรมไมโครซอฟต์เวิร์ด Microsoft Word, โปรแกรมโอเพ่นออฟฟิศ Open Office (Open Document Format) เป็นต้น ข้อดีของวิธีการที่คิดค้นนำเสนอในงานวิจัยนี้ คือ ช่วยให้เอกสารข้อกำหนดความต้องการและระบบต้นแบบมีความสอดคล้องตรงกัน (Consistency) เสมอ ในกรณีที่มีการเปลี่ยนแปลงแก้ไขจากเอกสารข้อกำหนดความต้องการ หรือจากระบบต้นแบบ นอกจากนี้ แนวทางงานวิจัยนี้ยังเป็นการสนับสนุนแนวคิดการทำกระบวนการให้มีความเป็นอัตโนมัติ อันเป็นกลยุทธ์การทำงานอย่างชาญฉลาดมากขึ้น

ซึ่งทำให้การใช้ทรัพยากรขององค์กรเป็นไปในทิศทางการปรับปรุงกระบวนการมากกว่าการสิ้นเปลืองทรัพยากรเพื่อแก้ไขงานหรือเพื่อทำงานซ้ำงานเดิมๆ

## 1.2 วัตถุประสงค์ของการวิจัย

นำเสนอวิธีการและพัฒนาระบบสำหรับสร้างต้นแบบลักษณะทดสอบได้โดยอัตโนมัติจากข้อกำหนดความต้องการที่ถูกเก็บเกี่ยว เพื่อสนับสนุนงานปรับปรุงกระบวนการ

## 1.3 ขอบเขตงานวิจัย

1. ระบบสร้างต้นแบบทำงานกับเอกสารข้อมูลนำเข้า (ข้อกำหนดความต้องการ) ที่สร้างจากไมโครซอฟต์เวิร์ด โดยข้อกำหนดความต้องการต้องบรรจุเนื้อหาตามที่กำหนดไว้ในวิทยานิพนธ์
2. ผลลัพธ์ต้นแบบที่พัฒนาขึ้นเป็นจาวา (Java) แอปพลิเคชัน
3. ประเมินความถูกต้องของต้นแบบที่พัฒนาขึ้น โดยเปรียบเทียบกับข้อกำหนดความต้องการต้นทาง

## 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้วิธีการและระบบสร้างต้นแบบลักษณะทดสอบได้โดยอัตโนมัติจากข้อกำหนดความต้องการโดยตรง ซึ่งช่วยลดความไม่สอดคล้องตรงกันระหว่างผลิตภัณฑ์งานทั้งสอง ในกรณีที่เกิดการเปลี่ยนแปลง
2. ระบบที่พัฒนาขึ้นช่วยลดเวลาและแรงงานในการพัฒนาซอฟต์แวร์
3. ระบบที่พัฒนาขึ้นช่วยสนับสนุนงานปรับปรุงกระบวนการและส่งเสริมกลยุทธ์การทำงานที่ฉลาดขึ้นในระยะยาว

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

## 1.5 วิธีดำเนินการวิจัย

1. ศึกษางานวิจัยและองค์ความรู้ที่เกี่ยวข้อง ได้แก่ ข้อกำหนดความต้องการ การพัฒนาและวิธีทดสอบส่วนติดต่อผู้ใช้การทำต้นแบบ
2. เลือกเทคโนโลยีที่ใช้ในการพัฒนาระบบ
3. ออกแบบระบบ
4. พัฒนาระบบบนสภาพแวดล้อมที่เลือก
5. ทดสอบระบบและประเมินผลวิธีวิจัย
6. ตีพิมพ์ผลงานทางวิชาการ
7. สรุปผลการวิจัยและข้อเสนอแนะ และจัดทำวิทยานิพนธ์

**1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์**

วิทยานิพนธ์นี้แบ่งเนื้อหาออกเป็น 6 บทดังต่อไปนี้ บทที่ 1บทนำกล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ของการวิจัย ขอบเขตของการวิจัย ประโยชน์ที่คาดว่าจะได้รับ รวมถึงวิธีดำเนินการวิจัย บทที่ 2 กล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องในงานวิจัยนี้ บทที่ 3 กล่าวถึงการออกแบบขั้นตอนการดำเนินงาน บทที่ 4 กล่าวถึงการออกแบบการพัฒนาระบบบทที่ 5 กล่าวถึงการทดสอบระบบ และบทที่ 6 สรุปผลการวิจัย ข้อเสนอแนะ และแนวทางสำหรับการวิจัยต่อในอนาคต

**1.7 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์**

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นบทความวิชาการในหัวข้อเรื่อง “AUTOMATED TESTING FEATURING PROTOTYPE GENERATION FROM HARVESTED REQUIREMENTS SPECIFICATION” โดย นาวิน เฟื่องฟู และ ญาใจ ลิ่มปิยะกรณ์ ในหนังสือรวมบทความการประชุมวิชาการนานาชาติ 2012 International Conference on Advanced Software Engineering & Its Applications ณ ประเทศเกาหลีใต้ วันที่ 29 พฤศจิกายน 2555 หน้า 88 - 95

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# บทที่ 2
# ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

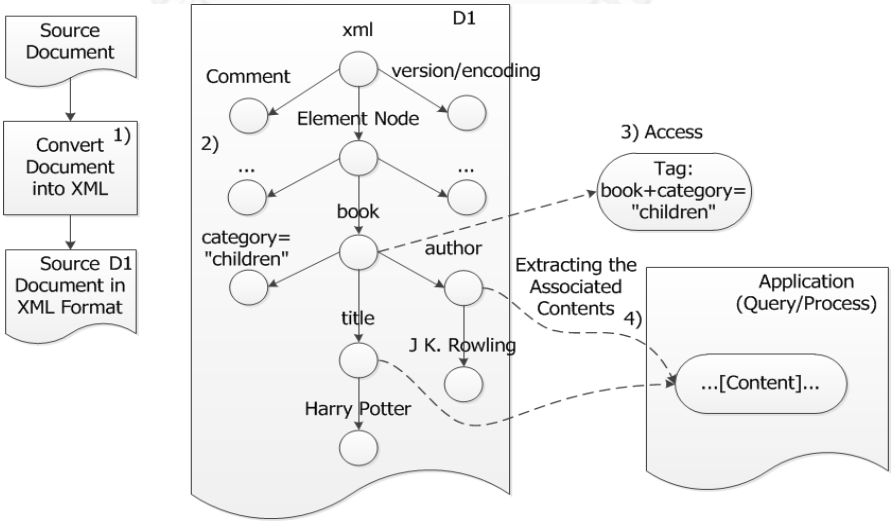
## 2.1 ทฤษฎีที่เกี่ยวข้อง

### 2.1.1 การเก็บเกี่ยวเนื้อหา (Content Harvester) [2]

Jackson [3] ได้ให้คำจำกัดความเกี่ยวกับข้อกำหนด (Specifications) ความต้องการ (Requirements) และโปรแกรม (Programs) ไว้ว่า “ข้อกำหนดคือ ปรากฏการณ์ที่ส่วนติดต่อระหว่างเครื่องจักรและสภาพแวดล้อมความต้องการ คือ ปรากฏการณ์สภาพแวดล้อมและโปรแกรม คือ ปรากฏการณ์เครื่องจักร” ดังนั้น การเก็บเกี่ยวเนื้อหาจากข้อกำหนดความต้องการจะสามารถนำมาซึ่งกระบวนการอัตโนมัติ เพราะข้อกำหนดความต้องการคือ สิ่งที่ช่วยประสานระหว่างความต้องการกับเครื่องจักร กล่าวอีกนัยหนึ่ง คือ ข้อกำหนดความต้องการช่วยให้เครื่องจักรสามารถทำความเข้าใจความต้องการของมนุษย์ได้

การเก็บเกี่ยวเนื้อหา หมายถึง การอ่านและวิเคราะห์เลือกข้อมูลที่เป็นประโยชน์ออกมาจากเอกสารทั้งที่ไร้โครงสร้างและมีโครงสร้างตามรูปแบบที่เหมือนๆ กัน และทำให้ข้อมูลนั้นพร้อมที่จะนำไปใช้ โดยทั่วไประเบียบวิธี (Methodology) เก็บเกี่ยวเนื้อหาตามภาพที่ 1 มีขั้นตอนดังนี้

1. แปลงเอกสารต้นทางเป็นรูปแบบ XML
2. ระบุบริเวณของเนื้อหาที่สนใจในรูปแบบเครื่องหมายข้อความ (Textual Markers) หรือสร้างป้าย (Tagging) ตามชื่อที่ผู้ใช้กำหนด (User-Defined Names)
3. เข้าถึงป้ายที่ถูกเลือกไว้
4. สกัดเนื้อหาที่เกี่ยวข้อง




ภาพที่ 1 กระบวนวิธีเก็บเกี่ยวเนื้อหา

**2.1.2 การทำต้นแบบ (Prototyping)**

ต้นแบบส่วนติดต่อผู้ใช้หรือต้นแบบแนวราบช่วยให้เห็นมุมมองของระบบทั้งหมดหรือระบบย่อย โดยมุ่งเน้นเรื่องของการติดต่อกับผู้ใช้มากกว่าระบบการทำงานชั้นล่างๆ เช่น การติดต่อฐานข้อมูล [4] หลักสำคัญของการทำต้นแบบ คือ ควรสร้างต้นแบบให้สำเร็จให้เร็วที่สุด เพื่อใช้สำรวจความต้องการของระบบและส่วนติดต่อผู้ใช้ได้ตั้งแต่ช่วงแรกๆ ของการพัฒนา

ต้นแบบสามารถจำแนกได้เป็น 2 ประเภท คือ ต้นแบบวิวัฒนาการ (Evolutionary Prototype) และต้นแบบใช้แล้วทิ้ง (Throwaway Prototype) โดยต้นแบบวิวัฒนาการสามารถพัฒนาไปสู่ระบบจริงที่ส่งมอบ การพัฒนาเป็นแบบวิธีการส่วนเพิ่ม (Incremental) เริ่มจากการเพิ่มเติมความต้องการ จากนั้นพัฒนาวนซ้ำตามข้อกำหนด การใช้และยันสอบระบบต้นแบบ กระทั่งข้อกำหนดเพียงพอต่อความต้องการจึงหยุด ซึ่งตรงข้ามกับต้นแบบใช้แล้วทิ้งที่พัฒนาขึ้นเพื่อช่วยระบุและยันสอบความต้องการของระบบ ซึ่งแม้ว่าจะสามารถทดสอบส่วนติดต่อกับผู้ใช้บนต้นแบบใช้แล้วทิ้งได้ แต่ไม่สามารถทดสอบสมรรถนะ (Performance Test) การทำงานได้ เนื่องจากไม่มีส่วนประมวลผลหรือส่วนที่ติดต่อกับฐานข้อมูลเหมือนในระบบจริง อย่างไรก็ตาม บางส่วนประกอบของต้นแบบใช้แล้วทิ้ง สามารถนำกลับมาใช้ใหม่ในระบบส่งมอบได้

2.1.2.1 การพัฒนาต้นแบบ

วิธีการพัฒนาต้นแบบระบบซอฟต์แวร์คล้ายกับวัฏจักรการพัฒนาซอฟต์แวร์ทั่วไป ได้แก่ วิธีการส่วนเพิ่ม วิธีการวงวน (Spiral) และวิธีการพัฒนาแอปพลิเคชันอย่างรวดเร็ว (Rapid Application Development— RAD) โดยทั่วไปขั้นตอนในการพัฒนาต้นแบบ [5] ประกอบด้วย

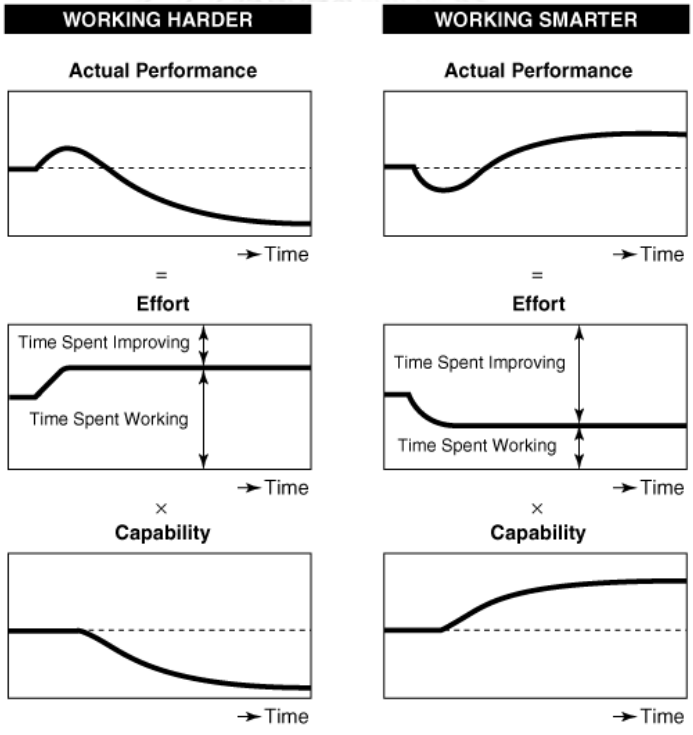
1. บ่งชี้ความต้องการพื้นฐาน
2. พัฒนาต้นแบบตั้งต้น
3. ทบทวน (Review)
4. แก้ไขและปรับปรุงต้นแบบ

วิธีการพัฒนาต้นแบบดังกล่าวข้างต้น หากนำมาสร้างให้มีความเป็นอัตโนมัติ จะสามารถลดต้นทุนและเวลาที่ใช้ในแต่ละขั้นตอน นอกจากนี้ ยังสามารถจัดการความสอดคล้องตรงกันระหว่างข้อกำหนดความต้องการและระบบต้นแบบได้ในกรณีที่มีการเปลี่ยนแปลงความต้องการเกิดขึ้น แนวทางดังกล่าวเป็นการปฏิบัติงานเชิงป้องกัน (Preventive) เพื่อลดข้อผิดพลาดของการพัฒนาผลิตภัณฑ์ซอฟต์แวร์ที่ไม่ตรงกับความต้องการตั้งแต่ตอนต้นโครงการ

2.1.2.2 ประโยชน์ของการสร้างต้นแบบอัตโนมัติ

ความเป็นอัตโนมัติลดงานที่ต้องทำ ลดต้นทุนการพัฒนา และสนับสนุนการปรับปรุงอย่างต่อเนื่อง (Continual Improvement) ซึ่งเป็นกลยุทธ์การทำงานอย่างชาญฉลาดขึ้น [1] ความเป็นอัตโนมัติยังทำให้ผู้พัฒนาไม่ต้องทำงานเดิมซ้ำๆ เช่น ไม่จำเป็นต้องเขียนผลิตภัณฑ์ซอฟต์แวร์ขึ้นใหม่ทั้งหมด แต่พวกเขาสามารถสร้างแม่แบบการแปลงระหว่างข้อกำหนดความต้องการซอฟต์แวร์และตัวแอปพลิเคชัน เพื่อนำผลิตผลจากการพัฒนากลับไปใช้ใหม่ในงานต่อๆ ไป นอกจากนี้ ความเป็นอัตโนมัติสามารถลดข้อผิดพลาดจากการทำงานของมนุษย์ ส่งเสริมให้ผลิตภัณฑ์มีคุณภาพมากยิ่งขึ้น

และการลดเวลาที่ใช้ในการแก้ไขงาน ทำให้เวลาที่ต้องใช้ในการพัฒนาโดยรวมลดลง ภาพที่ 2 แสดงการเปรียบเทียบผลลัพธ์ของกลยุทธ์ระหว่างการทำงานให้ฉลาดขึ้นกับการทำงานให้หนักขึ้น “การทำงานให้หนักขึ้น” ส่งผลให้สมรรถนะจริงเพิ่มขึ้นในช่วงแรกสั้นๆ แต่จะลดลงทีละน้อยเมื่อเวลาผ่านไป เรียกได้ว่าเป็นสถานการณ์ “ดีก่อนแล้วแย่ลง” เวลาและความพยายามส่วนมากเสียไปกับการทำงานมากกว่าใช้ปรับปรุงกระบวนการ ทำให้ความสามารถกระบวนการลดลงเมื่อเวลาผ่านไป ในทางกลับกัน “การทำงานให้ฉลาดขึ้น” นำไปสู่การเพิ่มสมรรถนะอย่างแท้จริงเมื่อเวลาผ่านไป เรียกได้ว่าเป็นสถานการณ์ “แย่ก่อนแล้วดีขึ้น” เวลาและความพยายามส่วนมากถูกนำไปใช้ในการปรับปรุงกระบวนการมากกว่าการทำงานประจำ ทำให้กระบวนการมีความสามารถเพิ่มขึ้นเมื่อเวลาผ่านไป การทำงานอย่างชาญฉลาดอาจจะต้องมีต้นทุนบางอย่างในขั้นต้น แต่เป็นการลงทุนที่คุ้มค่าต่อผลลัพธ์ที่ได้โดยรวมในภายหลัง หากเทียบเคียงกับงานในวิทยานิพนธ์นี้ ต้นทุนคือ ค่าใช้จ่ายในการทำวิจัยเพื่อคิดค้นวิธีการและพัฒนาเครื่องมือที่จะสามารถสร้างต้นแบบอัตโนมัติจากข้อกำหนดความต้องการ ซึ่งจะช่วยลดภาระงานหลายๆ งานที่ต้องปฏิบัติในกระบวนการพัฒนาซอฟต์แวร์ ทำให้เวลาโดยรวมของโครงการและข้อบกพร่อง (Defect) จากความเข้าใจความต้องการผิดพลาดลดลง




ภาพที่ 2 เปรียบเทียบผลลัพธ์ของกลยุทธ์ “การทำงานให้หนักขึ้น” และ“การทำงานให้ฉลาดขึ้น”

### 2.1.3 นิพจน์ปรกติ

นิพจน์ปรกติ (Regular Expression – regexp/regex) เป็นคำศัพท์ทางด้านวิทยาการคอมพิวเตอร์ หมายถึงสายอักขระที่อธิบายถึงรูปแบบของสายอักขระตามโครงสร้างรูปแบบที่กำหนด โดยนิพจน์ปรกตินั้นใช้อยู่แพร่หลายในโปรแกรมการแก้ไขข้อความ (Text editor) ในการค้นหาและ

ปรับเปลี่ยนข้อความ ภาษาโปรแกรมหลายภาษายังรองรับการใช้นิพจน์ปรกติสำหรับการจัดการและปรับเปลี่ยนสายอักขระ นิพจน์ปรกตินี้สามารถนำมาใช้ในการตรวจสอบรูปแบบของสายอักขระได้

นิพจน์ปรกติ ใช้อธิบายรูปแบบสายอักขระ ซึ่งการอธิบายมักจะกระชับและรัดกุมโดยที่ไม่ต้องอธิบายเป็นรายชื่อทั้งหมด ยกตัวอย่างเช่นคำว่า ปกติ และ ปรกติ สามารถอธิบายได้ว่า "ป(ร?)กติ" ซึ่งส่วนใหญ่มีโครงสร้างการใช้ดังตารางที่ 1

ตารางที่ 1 รายชื่อสัญลักษณ์ทั่วไปของนิพจน์ปรกติ

| สัญลักษณ์ | คุณลักษณะ | คำอธิบาย |
|---|---|---|
| $ | การระบุตำแหน่ง | เครื่องหมายสกุลเงิน ดอลลาร์ ใช้เขียนเป็นตัวท้ายสุดของนิพจน์ปรกติ เพื่อบ่งบอกว่านิพจน์ที่ต้องการค้นหาอยู่ท้ายบรรทัดหรือท้ายสายอักขระเท่านั้น สามารถใช้คู่กับ ^ |
| [ ] | การระบุช่วงอักขระ | วงเล็บเหลี่ยม ใช้เขียนแทนอักขระหนึ่งตัวที่อยู่ในขอบเขต ซึ่งระบุไว้ภายในวงเล็บ เช่น "[bcdf]an" มีความหมายว่า ban can dan และ fan |
| ^ | การระบุช่วงอักขระ | เครื่องหมายหมวก ใช้เขียนภายในวงเล็บเหลี่ยมเป็นตัวแรก เพื่อบ่งบอกว่าเป็นอักขระหนึ่งตัวที่ไม่ได้อยู่ในขอบเขต เช่น "[^bcdf]an" มีความหมายว่า aan ean gan han ian เป็นต้น |
| - | การระบุช่วงอักขระ | ยัติภังค์ ใช้เขียนภายในวงเล็บเหลี่ยมระหว่างอักขระสองตัว เพื่อบ่งบอกว่าเป็นอักขระหนึ่งตัวที่มีรหัสแอสกีหรือยูนิโคดอยู่ในช่วงนั้น เช่น "[a-z]an" มีความหมายว่า aan ban can เรื่อยไปจนถึง zan |
| . | สัญลักษณ์เพิ่มเติม | มหัพภาค ใช้บ่งบอกว่าอักขระในตำแหน่งนั้นของนิพจน์จะเป็นตัวอักษรอะไรก็ได้ เช่น ".an" มีความหมายว่า aan ban can 0an 1an -an เป็นต้น |
| \ | สัญลักษณ์เพิ่มเติม | แบ็กสแลช ใช้ระบุ escape character หรือสัญลักษณ์ที่ต้องการใช้ในนิพจน์ แต่ซ้ำกับสัญลักษณ์ที่กล่าวมาทั้งหมดด้านบน เช่น "[0-9]\.[0-9]\$" มีความหมายว่า 2.3$ 9.0$ 1.7$ เป็นต้น |

โดยเครื่องหมายต่างๆ สามารถสร้างนิพจน์ปรกติที่ซ้ำซ้อนและมีประสิทธิภาพสูงได้ ซึ่งคล้ายกับการสร้างนิพจน์คณิตศาสตร์

## 2.2 งานวิจัยที่เกี่ยวข้อง

### 2.2.1 A Method to Automate User Interface Testing Using Variable Finite State Machines [6]

งานวิจัยได้นำเสนอเครื่องจักรสถานะจำกัดซึ่งเปลี่ยนแปลงได้ (Variable Finite State Machines — VFSM) โดยเครื่องจักรสถานะจำกัดนี้สามารถแปลงเป็นเครื่องจักรสถานะจำกัดที่เทียบเท่า และสามารถนำไปสร้างการทดสอบส่วนติดต่อกับผู้ใช้โดยอัตโนมัติได้ องค์ประกอบหลักของวิธีการทำให้เครื่องจักรสถานะนี้เล็กและเป็นธรรมชาติมากกว่าเครื่องจักรสถานะจำกัดแบบอื่นๆ โดยที่สามารถสร้างได้แม่นยำกว่าใช้เวลาน้อยกว่าและเป็นวิธีการที่ยืดหยุ่น งานวิจัยได้ทดสอบโดยการนำไปใช้กับระบบ Navigator II ของแผนกซ่อมบำรุงอากาศยาน เปรียบเทียบกับโมเดลอื่นๆ ในหลายกรณี แล้วเปรียบเทียบกับผลลัพธ์ด้านต่างๆ ได้แก่ สถานะ การเปลี่ยนสถานะ และความผิดพลาดที่ตรวจพบ เป็นต้น

งานวิจัยได้นิยามเครื่องจักร ($M_V$) มี 7 ทูเปิล (S, I, O, T, $\varphi$, V, $\zeta$) โดยที่ S, I และ O ยังคงความหมายเดิมตามเครื่องจักรสถานะจำกัด คือ เป็นเซตจำกัดของ สถานะ, ข้อมูลเข้า, ข้อมูลออก ตามลำดับ มี $s_0$เป็นสถานะเริ่มต้น

V คือ เซตของตัวแปร ซึ่งแต่ละเซตคือ เซตของค่าที่เป็นไปได้ของตัวแปรนั้นๆ กล่าวคือ Vเป็นเซตของเซต

$V = \{V_1, V_2, ..., V_n\}$ โดยที่ n เป็นจำนวนของตัวแปรใน $M_V$และ $V_i$เป็นเซตของค่าที่เป็นไปได้ของตัวแปรตัวที่ i

ทั้ง T ($S \times I \rightarrow S$) และ $\varphi$ ($S \times I \rightarrow O$) ต้องถูกปรับเปลี่ยนให้เหมาะสมกับตัวแปรที่เพิ่มเข้าไป T ในตอนนี้จึงมีสมการตัวแปร (n+2) สมการตัวแปร

คือ $D_T \rightarrow S$ โดย $D_T \subset D$ และ $D = S \times I \times V_1 \times V_2 \times ... \times V_n$

ส่วน $\varphi$ กลายเป็น $D_T \rightarrow O$

ขอบเขตของ T และ $\varphi$ เป็นเซตย่อยของ D เพราะไม่ใช่ว่าการจับกลุ่มของตัวแปรทุกแบบสามารถเป็นไปได้ระหว่างการทำงาน

$\zeta$ เป็นเซตของสมการการเปลี่ยนสถานะตัวแปร เมื่อการเปลี่ยนสถานะเกิดขึ้น $\zeta$ ใช้พิจารณาว่าค่าใดตัวแปรถูกเปลี่ยนแปลง เหมือนกับ T และ $\varphi$ ทั้งสถานะปัจจุบัน ข้อมูลเข้าและค่าของตัวแปรทั้งหมดต้องอยู่ในขอบเขตของ $\zeta$ แต่ละตัวแปรต้องมีค่าเริ่มต้นที่เกิดขึ้นตอนเริ่มทำงาน

แนวทางงานวิจัยนี้ได้ประยุกต์ใช้เครื่องจักรสถานะจำกัดซึ่งเปลี่ยนแปลงได้มาพัฒนาในส่วนของการสร้างคำสั่งทดสอบ (Test Script) ซึ่งเป็นส่วนสำคัญหลักส่วนหนึ่งในการสร้างกลไกการทดสอบอัตโนมัติให้แก่ต้นแบบที่สร้างขึ้น

**2.2.2 Automated Prototyping of User Interfaces Based on UML Scenarios [7]**

งานวิจัยได้นำเสนอวิธีการเพื่อเชื่อมโยงแบบจำลองยูเอ็มแอล (UML Model) กับต้นแบบส่วนติดต่อผู้ใช้ ประกอบด้วย5 ขั้นตอนหลักโดยกระบวนการสำคัญเน้นที่การจัดการซีเนริโอ (Scenarios) เพื่อสร้างข้อกำหนดพื้นฐานให้แก่แอปพลิเคชัน

หลักการสำคัญของกระบวนการเป็นการดึงซีเนริโอจากแผนภาพการร่วมมือยูเอ็มแอล (UML Collaboration Diagrams) และเพิ่มข้อมูลทางส่วนติดต่อผู้ใช้ให้แก่ซีเนริโอโดยจะเปลี่ยนข้อมูลดังกล่าวเป็นแผนภูมิสถานะยูเอ็มแอล (UML Statechart) อ้างอิงวิธีการจากงานวิจัยของ Khriss et al. (1998) และ Schönberger et al. (2001) จากนั้นจึงนำผลลัพธ์ที่ได้ไปสร้างเป็นต้นแบบส่วนติดต่อผู้ใช้ต่อไป

แผนภาพยูเอ็มแอลที่ใช้ ประกอบด้วยแผนภาพคลาส (Class Diagram), แผนภาพยูสเคส (Use Case Diagram), แผนภาพการร่วมมือ (CollaborationDiagram) และแผนภาพแผนภูมิสถานะ (Statechart Diagram) โดยมีการใช้ภาษาข้อจำกัดวัตถุ (Object Constraint Language — OCL) สำหรับระบุข้อจำกัดต่างๆ เพื่อเพิ่มเติมรายละเอียดสำหรับการประมวลผล

ต้นแบบที่ได้เป็นต้นแบบโดดเดี่ยว (Standalone Prototype) มีเมนูให้เลือกระหว่างยูสเคสต่างๆ โดยมีหน้าจอต่างๆ เป็นมุมมองคงที่ (Static Aspect) และจับคู่วัตถุกับกล่องควบคุม เพื่อแสดงมุมมองเคลื่อนไหว (Dynamic Aspect)

ผู้วิจัยได้ศึกษางานวิจัยข้างต้นเพื่อให้ทราบถึงความต้องการพื้นฐานที่ควรกระทำได้ของระบบต้นแบบที่จะสร้าง รวมทั้งวิเคราะห์คุณสมบัติของต้นแบบที่งานวิจัยข้างต้นยังขาดไป แต่ควรมีการพัฒนาขึ้นเพิ่มเติม อย่างไรก็ตาม แนวทางงานวิจัยที่นำเสนอใช้วิธีการพัฒนาและใช้แหล่งข้อมูลนำเข้าเพื่อสร้างต้นแบบที่แตกต่างจากงานวิจัยข้างต้น รวมทั้งเพิ่มเติมความสามารถต่างๆ ได้แก่ การตรวจสอบค่าข้อมูลเข้าจริงในเบื้องต้น การอำนวยความสะดวกในการจัดวางองค์ประกอบหน้าจอ และความสามารถในการปรับปรุงข้อมูลกลับไปยังเอกสารต้นทาง ในกรณีที่มีการเปลี่ยนแปลงต้นแบบเป็นต้น

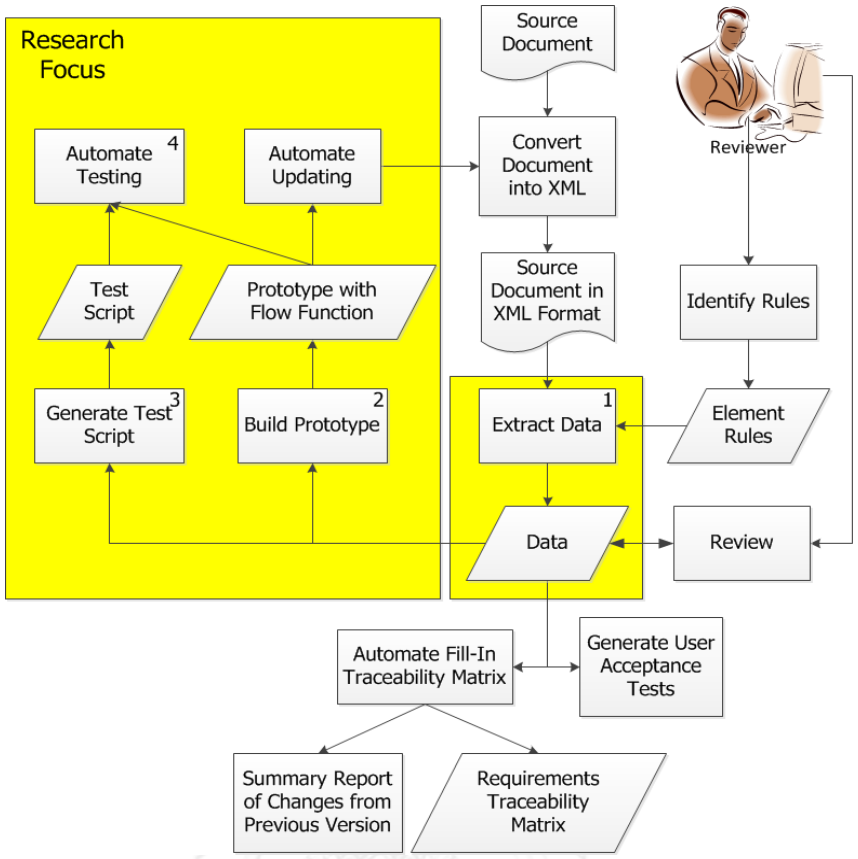
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# บทที่ 3
# การออกแบบระบบ

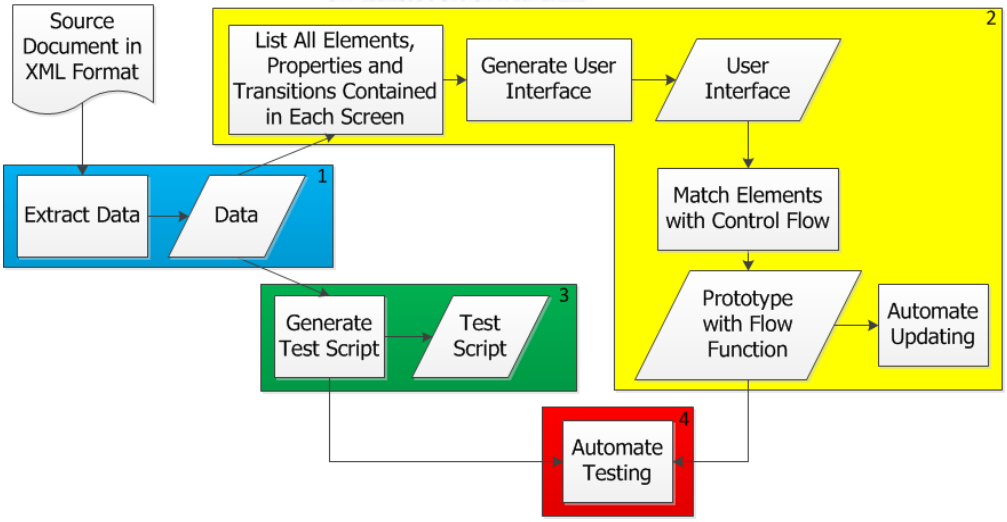
งานวิจัยนี้เป็นส่วนหนึ่งของระบบ Requirements Harvester (ภาพที่ 3) ซึ่งพัฒนาขึ้นโดยมีจุดประสงค์เพื่อปรับปรุงกระบวนการให้มีความเป็นอัตโนมัติ มุ่งเน้นการบูรณาการงานควบคุมคุณภาพ (Quality Control) อำนวยความสะดวกในการรวมกลุ่มของกิจกรรมการควบคุมคุณภาพผนวกกับความสามารถในการจัดการความต้องการ มุ่งเน้นต่อการจัดการความเปลี่ยนแปลงซึ่งเป็นสิ่งที่หลีกเลี่ยงได้ยากในโครงการซอฟต์แวร์ความสามารถในการบริหารจัดการความต้องการที่เปลี่ยนแปลงจึงเป็นสิ่งจำเป็น โดยปกติแล้วการตรวจสอบย้อนกลับโดยเมทริกซ์ตามรอยถูกนำมาใช้เป็นเครื่องมือในการให้ความสามารถในการติดตามการเปลี่ยนแปลงใดๆ และสะท้อนให้เห็นถึงการดำเนินการต่างๆ โดยเครื่องมือจัดการความต้องการเชิงพาณิชย์หลายตัว อันได้แก่ IBM's Rational® RequisitePro®, QSS's DOORS, TBI's Caliber-RM™, Integrated Chipware's RTM Workshop, or Open Source Management Requirements Tool (OSMRT)มีคุณสมบัติตรวจสอบย้อนกลับที่มีประสิทธิภาพ สถาปัตยกรรมของ RH ประกอบด้วย กิจกรรมการทวนสอบและการยันสอบใช้ได้จริง ผนวกเข้ากับการจัดการความต้องการ (Requirements Management) ซึ่งส่งผลให้มีความสามารถจัดการการเปลี่ยนแปลงความต้องการ ด้วยการสนับสนุนโดยเครื่องมือที่ใช้กันทั่วไป คือ เมตริกซ์ความสามารถตามรอย (Traceability Matrix)

ปัจจุบัน ระบบ Requirements Harvester ประกอบด้วยระบบย่อย 4 ระบบ คือ

1. เมตริกซ์ความสามารถตามรอย พัฒนาขึ้นในงานวิจัย [8]
2. ส่วนนำข้อมูลเข้าเมตริกซ์ความสามารถตามรอยโดยอัตโนมัติ พัฒนาขึ้นในงานวิจัย [9] ส่วนนี้สามารถพิจารณาเป็นส่วนเก็บเกี่ยวเนื้อหาเพื่อส่งข้อมูลไปยังแอปพลิเคชันอื่นได้ เช่น นำไปสร้างรายการและความสัมพันธ์ในเมตริกซ์ความสามารถตามรอยแบบอัตโนมัติ หรือนำไปสร้างกรณีทดสอบอัตโนมัติ เป็นต้น
3. ส่วนการสร้างกรณีทดสอบการยอมรับของผู้ใช้ พัฒนาขึ้นในงานวิจัย [10] โดยใช้ข้อมูลที่เก็บเกี่ยวจากข้อกำหนดความต้องการที่ส่งมาจากส่วนประกอบที่สองข้างต้น ผลลัพธ์กรณีทดสอบที่สร้างขึ้นสามารถส่งกลับไปเป็นข้อมูลนำเข้าส่วนประกอบที่สอง เพื่อสร้างรายการและความสัมพันธ์ลำดับชั้นต่อไปในเมตริกซ์ความสามารถตามรอย [8] แบบอัตโนมัติ
4. ส่วนการสร้างต้นแบบอัตโนมัติจากข้อกำหนดความต้องการที่ถูกเก็บเกี่ยว เป็นระบบย่อยที่พัฒนาขึ้นในงานวิจัยนี้ ประกอบด้วยส่วนประกอบ4 ส่วนหลัก คือ 1) ส่วนนำเข้าและสกัดข้อมูล 2) ส่วนการสร้างต้นแบบ 3) ส่วนการสร้างคำสั่งทดสอบ และ 4) ส่วนการทดสอบอัตโนมัติ




ภาพที่ 3 สถาปัตยกรรมระบบ Requirements Harvester




ภาพที่ 4 กระบวนการสร้างและทดสอบต้นแบบในงานวิจัย

ความสำคัญของงานวิจัยนี้คือ การเพิ่มประสิทธิภาพของ RH ด้วยการสร้างต้นแบบลักษณะทดสอบได้โดยอัตโนมัติจากข้อกำหนดความต้องการเก็บเกี่ยว โดยกระบวนการสร้างและทดสอบต้นแบบในงานวิจัยประกอบด้วย 4ส่วนหลัก คือ 1) ส่วนนำเข้าและสกัดข้อมูล 2) ส่วนการสร้าง

ต้นแบบ 3) ส่วนการสร้างคำสั่งทดสอบ และ 4) ส่วนการทดสอบอัตโนมัติตามภาพที่ 4 และมีความสามารถเพิ่มเติมเพื่อสนับสนุนการทำงานที่ต่อเนื่อง คือ การปรับปรุงเอกสารและการปรับปรุงต้นแบบมีรายละเอียดดังนี้

### 3.1 ส่วนนำเข้าและสกัดข้อมูล

ส่วนนำเข้าและสกัดข้อมูลเป็นส่วนเริ่มต้นของกระบวนการสร้างต้นแบบอัตโนมัติจากเอกสารข้อกำหนดความต้องการ ขั้นตอนนี้ประยุกต์ใช้กระบวนวิธีการเก็บเกี่ยวเนื้อหา เพื่อสกัดข้อมูลที่ต้องการ การทำงานในส่วนนี้จำเป็นต้องข้อมูลสำคัญต่างๆ โดยมีการกำหนดรูปแบบการเขียนเอกสารข้อกำหนดความต้องการให้บรรจุรายละเอียด ดังต่อไปนี้

#### 3.1.1 ข้อกำหนดการติดป้าย (Tagging Specification)

ข้อกำหนดการติดป้ายจำเป็นต่อการระบุตำแหน่งหรือขอบเขตบริเวณของเนื้อหาที่สนใจตามขั้นตอนที่ 2 ของกระบวนวิธีเก็บเกี่ยวเนื้อหา เพื่อใช้ในการค้นหาและเข้าใจว่าเนื้อหาส่วนใดคืออะไร เปรียบเสมือนการอ่านเอกสารโดยทั่วไปของมนุษย์ที่ใช้รูปแบบบ่งชี้หัวข้อดังเช่นการทำหมายเลขข้อและการทำตัวหนาในรายงานฉบับนี้ ข้อกำหนดการติดป้ายที่จำเป็นต้องระบุไว้ในเอกสารข้อกำหนดความต้องการ ประกอบด้วย

1. ข้อความ (Text)

ข้อความที่กำหนดให้เป็นป้ายบอกตำแหน่งในการสกัดข้อมูล ตัวอย่างเช่น ข้อความหัวข้อของส่วนสำคัญต่างๆในรายงาน ได้แก่ “บทที่ #”, “ภาพที่ #” และ “ตารางที่ #” เป็นต้น

2. สัญลักษณ์ (Content)

เป็นการใช้สัญลักษณ์ต่างๆ ในการบ่งชี้ตำแหน่งสำคัญ ตัวอย่างเช่น การกำหนดการอ้างอิงในรายงานด้วยเครื่องหมาย “[ ]” โดยมีเลขภายในช่องลำดับการอ้างอิง เครื่องหมาย “[“ และ “]” เป็นป้ายบอกตำแหน่งให้ผู้อ่านได้รับรู้ว่า ประโยคก่อนหน้าอ้างอิงไปยังเอกสารในส่วนอ้างอิง เป็นต้น

3. รูปแบบ (Style)

รูปแบบของข้อความที่กำหนดให้เป็นป้ายบอกตำแหน่งในการสกัดข้อมูล ได้แก่ สกุลอักษร รูปแบบอักษร (ตัวหนา ตัวเอียง เส้นใต้ข้อความ อื่นๆ) และขนาด เป็นต้น รูปแบบทั่วไปที่พบได้แสดงด้วยตารางที่ 2

ตารางที่ 2 ตัวอย่างรูปแบบของข้อความที่สามารถนำมาใช้เป็นป้ายบอกตำแหน่ง

| รูปแบบ | คำอธิบาย |
|---|---|
| สกุลอักษร | Angsana New, Cordia New, Times New Roman เป็นต้น |
| รูปแบบอักษร | ตัวหนา, ตัวเอียง, มีเส้นใต้ข้อความ เป็นต้น |
| ลักษณะเส้นใต้ | เส้นประ, เส้นประสั้นสลับยาว, จุดไข่ปลา เป็นต้น |
| สีอักษร | ดำ, แดง, เขียว เป็นต้น |
| สีพื้นหลังอักษร | ดำ, แดง, เขียว เป็นต้น |
| ขนาดอักษร | 16หน่วย, 18หน่วย, 30หน่วย เป็นต้น |

| | |
|---|---|
| ลักษณะพิเศษ | การทำตัวยก ตัวห้อย เป็นต้น |

4. ลำดับชั้น (Level)

ลำดับชั้นของข้อความที่กำหนดให้เป็นป้ายบอกตำแหน่งในการสกัดข้อมูล โดยการเก็บเกี่ยวเนื้อหาที่ใช้สามารถมองเอกสารแบบเป็นลำดับชั้นได้ กล่าวคือ สามารถมองเห็นรูปแบบของหัวข้อหลัก หัวข้อรองและหัวข้อย่อยๆ อื่นๆ ตัวอย่างการใช้งานโดยทั่วไปเช่น กำหนดให้องค์ประกอบหน้าจออยู่ในหัวข้อที่มีข้อความใดๆ ก็ได้ แต่เป็นหัวข้อลำดับย่อยที่สามของคำอธิบายแต่ละหน้าจอ เป็นต้น

ป้ายบอกตำแหน่งดังกล่าวอาจรวบรวมเป็นกลุ่มรูปแบบหนึ่งๆ โดยมีการตั้งชื่อไว้ เพื่อเรียกใช้งาน ความสามารถนี้มีในโปรแกรมการจัดการเอกสารทั่วไปหลายๆ โปรแกรมด้วยกัน เช่น ไมโครซอฟต์เวิร์ด และ โอเพ่นออฟฟิศ ซึ่งเป็นการตั้งชื่อป้ายบอกตำแหน่งขึ้นมานั่นเอง

ป้ายบอกตำแหน่งแบบต่างๆ จำเป็นในการตั้งค่าเพื่อบ่งชี้ให้ระบบเข้าใจประเภทของข้อมูล ณ ตำแหน่งต่างๆ ในเอกสาร เพื่อสกัดออกมาเป็นสารสนเทศที่เป็นประโยชน์ ตามชนิดต่างๆ ที่เรากำหนดเอาไว้

### 3.1.2 ประเภทข้อมูลเข้า

ประเภทข้อมูลเข้า คือ การพิจารณาชนิดของเนื้อความจากเอกสารในบริเวณหนึ่งๆ ว่าเป็นเนื้อความที่บ่งบอกถึงอะไร ซึ่งใช้ข้อกำหนดการติดป้ายเป็นข้อบ่งชี้หลัก และมีรูปแบบที่ตกลงกันไว้เป็นข้อบ่งชี้รอง ตัวอย่างเช่น กำหนดให้ข้อมูลเกี่ยวกับ ‘a’ อยู่ใน ‘ตาราง’ ภายใต้หัวข้อ ‘A’ ตามตัวอย่างนี้ ‘A’ คือ ข้อบ่งชี้หลัก เป็นข้อกำหนดการติดป้ายแบบระบุข้อความการเก็บข้อมูลใน ‘ตาราง’ เป็นข้อบ่งชี้รอง ทั้งหมดนี้ช่วยบ่งชี้บริเวณและวิธีการเก็บเกี่ยวเนื้อหาที่เหมาะสมของข้อมูลเข้าประเภท ‘a’ โดยข้อบ่งชี้เหล่านี้ เราสามารถกำหนดได้อย่างยืดหยุ่น คือ ปรับเปลี่ยนได้ง่าย ตามคำจำกัดความที่กล่าวมาข้างต้น ประเภทข้อมูลเข้าที่จำเป็นต้องระบุไว้ในเอกสารข้อกำหนดความต้องการประกอบด้วย

1. ข้อมูลเกี่ยวกับองค์ประกอบหรือเขตข้อมูล (Elements/Fields)

เป็นข้อมูลเกี่ยวกับองค์ประกอบของส่วนติดต่อผู้ใช้ (User Interface Elements) ต่างๆ เช่น เขตข้อมูลข้อความ (Text Field) ป้ายข้อความ (Text Label) เป็นต้นเพื่อให้โปรแกรมสามารถสกัดข้อมูลเหล่านี้ไปสร้างเป็นองค์ประกอบหรือเขตข้อมูลดังกล่าวในต้นแบบ โดยตารางที่ 3 แสดงข้อมูลเกี่ยวกับองค์ประกอบหรือเขตข้อมูลดังกล่าว

ตารางที่ 3 ข้อมูลเกี่ยวกับองค์ประกอบหรือเขตข้อมูล

| รายการ | คำอธิบาย |
|---|---|
| ชื่อเขตข้อมูล | ชื่อของเขตข้อมูลนั้นๆ เช่น“ช่องกรอกชื่อผู้ใช้”, “ช่องกรอกรหัสผู้ใช้” เป็นต้น |
| ประเภทเขตข้อมูล | เขตข้อมูลข้อความ ป้ายข้อความ กล่องการเลือก รูปภาพ อื่นๆ |
| ประเภทเข้า/ออก | ประเภทของเขตข้อมูล เป็นเขตการรับข้อมูล (เขตข้อมูลเข้า) หรือเขตการ |

| | |
|---|---|
| | รายงานข้อมูล (เขตข้อมูลออก) |
| ความสำคัญ | ข้อมูลในเขตข้อมูลเป็นข้อมูลจำเป็นหรือข้อมูลเสริมที่ไม่จำเป็นต้องมี ข้อมูลจำเป็นของเขตข้อมูลเข้าหมายถึงเขตข้อมูลต่างๆ ที่ผู้ใช้ต้องกรอกก่อนทำการยินยอมส่งผ่านหน้าจอ |
| รูปแบบข้อมูล | รูปแบบการรับหรือการแสดงผลข้อมูลในเขตข้อมูลนั้น เช่น ตัวอักษรเท่านั้น, ตัวเลขเท่านั้น หรือตัวอักษรภาษาอังกฤษตัวพิมพ์เล็กผสมตัวเลขเท่านั้น เป็นต้น สามารถขยายขอบเขตให้ยืดหยุ่นได้ด้วยการใช้นิพจน์ปรกติเข้าช่วย |
| รายละเอียดเพิ่มเติม | ตำแหน่งของเขตข้อมูล ขนาดของเขตข้อมูล ป้ายชื่อกำกับเขตข้อมูล อื่นๆ |

2. ข้อมูลเกี่ยวกับเหตุการณ์/การกระทำ (Events/Actions)

เป็นข้อมูลเกี่ยวกับองค์ประกอบส่วนติดต่อผู้ใช้ซึ่งควบคุมส่วนติดต่อผู้ใช้ เช่น ปุ่ม (Button) เขตเชื่อมโยง (Link) เป็นต้น โดยตารางที่ 4 แสดงข้อมูลเกี่ยวกับเหตุการณ์/การกระทำดังกล่าว

ตารางที่ 4 ข้อมูลเกี่ยวกับเหตุการณ์/การกระทำ

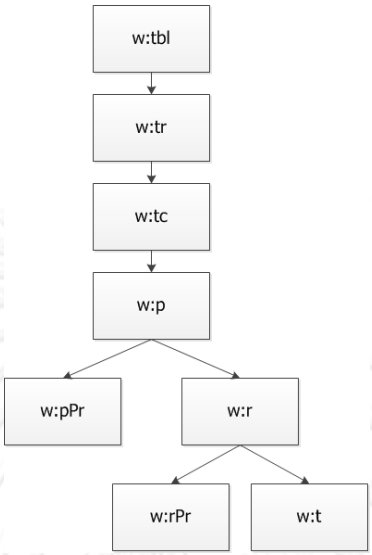
| **รายการ** | **คำอธิบาย** |
|---|---|
| ชื่อเขตข้อมูล | ชื่อของเขตข้อมูลนั้นๆ เช่น“ปุ่มเข้าระบบ”, “ปุ่มล้างข้อมูล”, “ปุ่มแก้ไขรายการจัดซื้อ” เป็นต้น |
| ประเภทเขตข้อมูล | ปุ่มกด เขตเชื่อมโยง อื่นๆ |
| ไอคอน | ไอคอนของเขตข้อมูล เช่น ไอคอนแทนปุ่มกด เป็นต้น |
| รายละเอียดเพิ่มเติม | ชื่อปลายทางการเปลี่ยนหน้าจอ ชื่อปลายทางการเปลี่ยนหน้าจอกรณีผิดปกติ ป้ายชื่อกำกับเขตข้อมูล อื่นๆ |

3. แม่แบบหน้าจอ (Screen Template)

ใช้เป็นแม่แบบของหน้าจอสำหรับการสร้างส่วนติดต่อผู้ใช้สามารถเป็นภาพร่างเขียนด้วยลายมือ, ภาพพิมพ์, ภาพจับภาพหน้าจอหรืออื่นใด ข้อมูลส่วนนี้ไม่จำเป็นต้องมี แต่หากมีในเอกสารเราสามารถสกัดเข้าโปรแกรมเพื่อนำมาเป็นแนวทางการกำหนดตำแหน่งองค์ประกอบแก่ผู้ใช้งานให้วางตำแหน่งและปรับตั้งขนาดเขตข้อมูลสะดวกขึ้นได้

รูปแบบของข้อมูลเข้าที่จะทำการสกัดจะสกัดจากต้นไม้ของ Document Object Model (DOM) ตัวอย่าง Document Object Model ของตารางในโปรแกรมไมโครซอฟต์เวิร์ดจะแสดงในภาพที่ 5 ในงานวิจัยครั้งนี้เนื้อหาจากการเก็บเกี่ยวข้อกำหนดความต้องการจะถูกเก็บไว้ในต้นไม้โครงสร้าง DOM ซึ่งจะช่วยให้ข้อมูลการเข้าถึงโดยตรงและช่วยให้การปรับเปลี่ยนข้อมูล นอกจากนี้ยังช่วยให้การสามารถปรับปรุงโดยอัตโนมัติลงในเอกสารข้อกำหนดความต้องการได้เมื่อมีการเปลี่ยนแปลงในต้นแบบที่เกี่ยวข้อง ในความเป็นจริงเมื่อมีการแปลงเอกสารต้นฉบับที่มีเอกสารประเภท "doc" หรือ "docx" ซึ่งเป็นเป้าหมายที่ระบบทดสอบที่สร้างขึ้นรองรับนั้น เอกสารXMLและเอกสารชนิดอื่นๆ เพิ่มเติมจำนวนมากจะถูกสร้างขึ้นนอกเหนือจากเอกสาร XMLหลักที่จะมีโครงสร้างของโปรแกรมไมโครซอฟต์เวิร์ดดังกล่าวซึ่งการสกัดไฟล์บางอย่าง เช่น ภาพในเอกสารหรือกลุ่ม

รูปแบบที่กำหนดขึ้นของข้อความ เป็นต้น ต้องใช้การประมวลผลข้อมูลในเอกสารหลายเอกสารย่อย เพื่อให้ได้ข้อมูลที่จำเป็นไปสู่แหล่งภาพหรือรับรู้รายละเอียดของรูปแบบ อันได้แก่ ชื่อ รูปแบบอ้างอิงหลัก(ถ้ามี) รูปแบบแยกย่อย (ขนาด ตัวหนา ตัวเอียง อื่นๆ) และลำดับชั้นของข้อความ รายการของเอกสารบางส่วนได้แสดงไว้ในตารางที่ 5




ภาพที่ 5 Document Object Model ของตารางในโปรแกรมไมโครซอฟต์เวิร์ด

ตารางที่ 5 รายการสำคัญของเอกสารของไมโครซอฟต์เวิร์ด

| รายการ | ความสำคัญ | คำอธิบาย |
|---|---|---|
| word/document.xml | เอกสารหลักของไมโครซอฟต์เวิร์ด | เป็นเอกสารที่รวบรวมข้อมูลหลักๆ ไว้ เช่น เนื้อหาของเอกสาร การอ้างอิงไปยังรูปแบบข้อความที่ตั้งไว้แบบต่างๆ |
| word/_rels/document.xml.rels | เอกสารเชื่อมโยงความสัมพันธ์ | สามารถหาความสัมพันธ์จากข้อความอ้างอิงภาพในเอกสารหลัก (word/document.xml) ที่เชื่อมโยงไปยังภาพในแฟ้มข้อมูลสื่อของไมโครซอฟต์เวิร์ด (word/media/) |
| word/media/ | แฟ้มข้อมูลที่รวบรวมเอกสารชนิดรูปภาพและวัตถุอื่นๆ ที่เกี่ยวข้องกับสื่อ | เป็นแฟ้มข้อมูลที่รวบรวมเอกสารชนิดรูปภาพและวัตถุอื่นๆ ที่เกี่ยวข้องกับสื่อเอาไว้ ตัวอย่างเช่น วัตถุของโปรแกรมvisio ที่ถูกคัดลอกมาเข้าในเอกสาร จะไม่ใช่เป็นการคัดลอกรูปภาพมา แต่เป็นการคัดลอกวัตถุเลย |
| word/numbering.xml | เอกสารบรรจุรูปแบบของลำดับตัวเลข | บรรจุรูปแบบของลำดับตัวเลขตามข้อความอ้างอิงในเอกสารหลัก |

| | | |
|---|---|---|
| word/styles.xml | เอกสารบรรจุรูปแบบของรูปแบบข้อความ | บรรจุรูปแบบของรูปแบบข้อความตามข้อความอ้างอิงในเอกสารหลัก |

เอกสารดังกล่าวจะเก็บผู้ในรูปของเอกสารบีบอัดชนิด “zip”หากเป็นไมโครซอฟต์เวิร์ดชนิด “docx” สามารถทดสอบได้โดยการเปลี่ยนสกุลเอกสารเป็น “zip” แล้วนำไปใช้งานในโปรแกรมจัดการเอกสารบีบอัดชนิด “zip” ได้ ผู้พัฒนาสามารถแตกเอกสารบีบอัดนี้เป็นเอกสารย่อยๆ เพื่อนำไปใช้งานได้ อย่างไรก็ตาม ระบบที่พัฒนาเพื่อทดสอบงานวิจัยนี้ใช้วิธีการอ่านระเบียนข้อมูลของไฟล์บีบอัดโดยตรงผ่านไลบรารี่ของภาษาที่ใช้ในการพัฒนา เพราะมีความสะดวกเหมาะสมมากกว่าและไม่ต้องคำนึงถึงพื้นที่ในการแตกไฟล์บีบอัด การปรับปรุงเอกสารที่จะกล่าวถึงภายหลังในบทเดียวกันนี้ก็เช่นกัน ผู้พัฒนาใช้การปรับปรุงระเบียนภายในระบบโดยตรง และให้ผลออกมาเป็นไฟล์บีบอัดให้ในรูปแบบ docx/zip เลย เพื่อให้สามารถนำไปใช้งานได้ทันที โดยปราศจากขั้นตอนที่ต้องทำเพิ่มเติม

**3.2 ส่วนการสร้างต้นแบบ**

ส่วนการสร้างต้นแบบทำหน้าที่สร้างต้นแบบอัตโนมัติจากข้อมูลต่างๆ ที่ได้จากการเก็บเกี่ยวเนื้อหาในขั้นตอนที่1 ส่วนการสร้างต้นแบบ ประกอบด้วย 3 ขั้นตอนย่อยดังภาพที่ 4 ดังนี้

**3.2.1 ระบุองค์ประกอบส่วนติดต่อผู้ใช้คุณสมบัติ และการส่งผ่านของแต่ละหน้าจอ**

เป็นการระบุองค์ประกอบส่วนติดต่อผู้ใช้คุณสมบัติ และการส่งผ่านของแต่ละหน้าจอด้วยข้อมูลที่สกัดได้จากขั้นตอนที่ 1 โดยระบบทำการสกัดข้อมูลแล้วประมวลผลตามข้อมูลที่สกัดได้มานำมาแบ่งแยกเป็นองค์ประกอบส่วนติดต่อผู้ใช้ต่างๆ ตามแต่ละหน้าจอ คุณสมบัติของแต่ละองค์ประกอบ และการส่งผ่านหน้าจอด้วยการกระทำการหนึ่งๆ ต่อส่วนติดต่อผู้ใช้หนึ่งๆ ตัวอย่างเช่นระบบทำการสกัดข้อมูลได้ว่าหน้าการเข้าสู่ระบบ มีรายละเอียดดังต่อไปนี้

ตารางที่ 6 รายการสำคัญขององค์ประกอบในหน้าการเข้าสู่ระบบ

| รายการ | ชนิดเขตข้อมูล | คำอธิบาย |
|---|---|---|
| เขตข้อมูลชื่อผู้ใช้ | เขตข้อมูลประเภทข้อความ | เขตข้อมูลให้ผู้ใช้กรอกชื่อผู้ใช้ |
| เขตข้อมูลรหัสผู้ใช้ | เขตข้อมูลประเภทข้อความรหัส | เขตข้อมูลให้ผู้ใช้กรอกรหัสผู้ใช้ |
| กล่องการเลือกจดจำผู้ใช้ | กล่องการเลือก | เขตข้อมูลให้ผู้ใช้เลือกให้ระบบจดจำผู้ใช้ |
| เขตเชื่อมโยงการลืมรหัสผู้ใช้ | เขตเชื่อมโยง | เขตเชื่อมโยงให้ผู้ใช้เลือกเมื่อลืมรหัสผู้ใช้ ทำการเชื่อมโยงไปยังหน้าการกู้รหัสหากผู้ใช้กดเลือกเขตเชื่อมโยงนี้ |
| ปุ่มการเข้าสู่ระบบ | ปุ่ม | ปุ่มให้ผู้ใช้เลือกเมื่อกรอกข้อมูลที่ตั้งไว้ว่ามี |

| | | |
|---|---|---|
| | | ความสำคัญชนิดจำเป็นในเอกสารครบถ้วน และถูกต้องตามการตรวจสอบที่ตั้งไว้ |
| ปุ่มการล้างข้อมูลเข้าระบบ | ปุ่ม | ปุ่มให้ผู้ใช้เลือกเมื่อเมื่อต้องการล้างข้อมูลที่กรอกไว้ในหน้าเข้าสู่ระบบ |

ข้อมูลต่างๆ ดังตารางที่ 6 เป็นตัวอย่างข้อมูลส่วนหนึ่งที่ระบบสามารถสกัดออกมาได้ นอกจากนี้ ยังมีข้อมูลอื่นๆ หากระบุไว้ ดังตารางที่ 3 และตารางที่ 4 ข้อมูลขององค์ประกอบต่างๆ รวมไปถึงส่วนของรายละเอียดเพิ่มเติมแสดงดังตารางที่ 7 ตัวอย่างของข้อมูลจริงในเอกสารแสดงดังภาพที่ 6

ตารางที่ 7 ข้อมูลขององค์ประกอบต่างๆ รวมไปถึงส่วนของรายละเอียดเพิ่มเติม

| **รายการ** | **แหล่งข้อมูลทั่วไป** | **การนำมาใช้งาน** |
|---|---|---|
| ตำแหน่ง | รับค่าจากการกดเลือกตำแหน่งโดยผู้ใช้งานโปรแกรม หรือโดยไฟล์ข้อมูลในรูปแบบ XML ที่บันทึกไปจากโปรแกรม | ใช้เพื่อเป็นตำแหน่งตั้งต้นขององค์ประกอบนั้นๆ |
| ขนาดขององค์ประกอบ | รับค่าจากการเลือกตั้งค่าจากผู้ใช้งานโปรแกรม หรือโดยไฟล์ข้อมูลในรูปแบบ XML ที่บันทึกไปจากโปรแกรม หากไม่ระบุใช้ค่ามาตรฐาน | ใช้เพื่อเป็นขนาดขององค์ประกอบนั้นๆ |
| ข้อความขององค์ประกอบ | สกัดจากเอกสารข้อกำหนดความต้องการ | ใช้เพื่อเป็นข้อความแทนองค์ประกอบนั้นๆ |
| รายการหล่นลงสำหรับเลือก | สกัดจากเอกสารข้อกำหนดความต้องการ หรือรับเพิ่มเติมจากผู้ใช้ | ใช้เป็นข้อมูลในกรณีเขตข้อมูลเป็นประเภทรายการหล่นลงสำหรับเลือกหรือรายการในรูปแบบใกล้เคียง ว่ามีรายการย่อยใดในเขตข้อมูลนี้บ้าง |
| นิพจน์ปรกติสำหรับตรวจสอบประเภทข้อมูลในเขตข้อมูล | สกัดจากเอกสารข้อกำหนดความต้องการในคอลัมน์รูปแบบข้อมูล | ใช้เป็นรูปแบบการตรวจสอบข้อมูลในเขตข้อมูลว่าถูกต้องหรือไม่ หากไม่ถูกต้อง และเกิดการเปลี่ยนแปลงส่วนติดต่อผู้ใช้ จะดำเนินการเปลี่ยนแปลงในกรณีผิดปกติ |
| รูปแบบวันที่ในตัวเลือกวันที่ | สกัดจากเอกสารข้อกำหนดความต้องการ ถ้าไม่ได้ตั้งไว้ระบบจะใช้ค่ามาตรฐาน | ใช้ระบุรูปแบบวันที่กรณีเป็นเขตข้อมูลแสดงวันที่ |
| กล่องข้อความชี้แนะ | สกัดจากเอกสารข้อจำกัด | ใช้เป็นข้อความแสดงเมื่อผู้ใช้งานชี้ไปยัง |

| | | |
|---|---|---|
| | | เขตข้อมูลดังกล่าว เพื่ออธิบายเขตข้อมูล |
| ประเภทเขตข้อมูลเข้า/ออก | สกัดจากเอกสารข้อกำหนดความต้องการในคอลัมน์ประเภทเข้า/ออก | ประเภทเขตข้อมูลว่าเป็นเขตข้อมูลที่รับข้อมุลเข้า หรือเป็นเขตข้อมูลที่แสดงข้อมูลออก |
| สถานะความสำคัญ | สกัดจากเอกสารข้อกำหนดความต้องการในคอลัมน์ความสำคัญ | สถานะความสำคัญของข้อมูลในเขตข้อมูลดังกล่าว หากกำหนดว่าต้องการการเปลี่ยนแปลงส่วนติดต่อผู้ใช้จะถือว่าเป็นกรณีผิดปกติ หากขาดเขตข้อมูลนี้ |
| ความต้องการเขตข้อมูล | สกัดจากเอกสารข้อกำหนดความต้องการ หรือรับเพิ่มเติมจากผู้ใช้ | ใช้เพื่อระบุกรณีของการดำเนินการเปลี่ยนแปลงในกรณีข้อมูลในเขตข้อมูลสำคัญไม่ครบ หรือข้อมูลในเขตข้อมูลต่างๆ ไม่ถูกต้อง ว่าเป็นกรณีปกติหรือผิดปกติ หากไม่มีความต้องการเขตข้อมูล ต้นแบบก็จะไม่ตรวจสอบข้อมูลในเขตข้อมูล แต่จะดำเนินการเปลี่ยนแปลงแบบปกติไปเลย ตัวอย่างเช่น เขตเชื่อมโยงการลืมรหัสผู้ใช้ไม่ต้องการข้อมูลในเขตข้อมูลต่างๆ ในการเปลี่ยนไปสู้หน้ากู้รหัสผู้ใช้ |
| การเชื่อมโยงกรณีปกติ | สกัดข้อมูลจากตารางเกี่ยวกับเหตุการณ์/การกระทำ | ใช้ในการเปลี่ยนแปลงส่วนติดต่อผู้ใช้จากหน้าปัจจุบันไปสู่หน้าที่ระบุด้วยองค์ประกอบนี้ในกรณีปกติ |
| การเชื่อมโยงกรณีผิดปกติ | สกัดข้อมูลจากตารางเกี่ยวกับเหตุการณ์/การกระทำ | ใช้ในการเปลี่ยนแปลงส่วนติดต่อผู้ใช้จากหน้าปัจจุบันไปสู่หน้าที่ระบุด้วยองค์ประกอบนี้ในกรณีผิดปกติ |
| ข้อความติดป้าย (Label) เขตข้อมูล | สกัดจากเอกสารข้อกำหนดความต้องการ หรือรับเพิ่มเติมจากผู้ใช้ | ใช้เป็นข้อความป้ายขององค์ประกอบนั้นๆ เช่น ป้ายคำอธิบายเขตข้อมูลชื่อผู้ใช้เป็นคำว่า “ชื่อผู้ใช้” เป็นต้น |
| ตำแหน่งป้ายเขตข้อมูล | ตำแหน่งมาตรฐานตามการตั้งค่าโปรแกรม | ใช้เป็นตำแหน่งป้ายขององค์ประกอบนั้นๆ |
| ภาพไอคอน | สกัดจากเอกสารข้อกำหนดความต้องการ หรือรับเพิ่มเติมจากผู้ใช้ | ใช้เพื่อเป็นภาพไอคอนแก่องค์ประกอบนั้นๆ เช่น ภาพของปุ่มในกรณีองค์ประกอบเป็นประเภทปุ่ม |
| หมายเหตุ | สกัดจากเอกสารข้อกำหนดความต้องการ | ใช้เพื่อเพิ่มข้อความหมายเหตุเหนือองค์ประกอบนั้นๆ ในโค้ดต้นแบบ |

### 3.2.2 สร้างส่วนติดต่อผู้ใช้

สร้างส่วนติดต่อผู้ใช้ขึ้นโดยอัตโนมัติจากข้อมูลที่สกัดและประมวลได้ในส่วนที่ผ่านมาโดยระบบจะแสดงรายละเอียดของแต่ละองค์ประกอบส่วนติดต่อผู้ใช้ที่เก็บเกี่ยวมาจากขั้นตอนนำเข้าและสกัดข้อมูลให้ผู้ใช้ตรวจสอบและสามารถเพิ่มเติมรายละเอียดสำคัญที่ขาดหายไปของแต่ละองค์ประกอบส่วนติดต่อผู้ใช้บนแม่แบบหน้าจอที่แสดงเป็นพื้นหลัง ซึ่งระบบจะอำนวยความสะดวกแก่ผู้ใช้ได้ตามความเหมาะสมของการให้ข้อมูลเพิ่มเติม เช่น การระบุตำแหน่งโดยการกดชี้ตำแหน่งบนหน้าจอ การกำหนดขนาดพื้นที่โดยการกดชี้พิกัดที่มุมบนซ้ายและพิกัดมุมล่างขวาหรือการเพิ่มลดค่าตัวเลขด้วยการกดลูกศรเพิ่มลด เป็นต้น

Mapping Fields

| Field Name | I/O | M/O | Format | Detail |
|---|---|---|---|---|
| User Name | I | M | Digit | **text:** 510001 |
| Password | I | M | Varchar | **password:** * |
| Remember Password | I | O | CheckBox | **text:** Remember my password |
| Logo | O | O | Image | |

Notes: I/O denotes Input/Output
M/O denotes Mandatory/Optional

Events / Actions

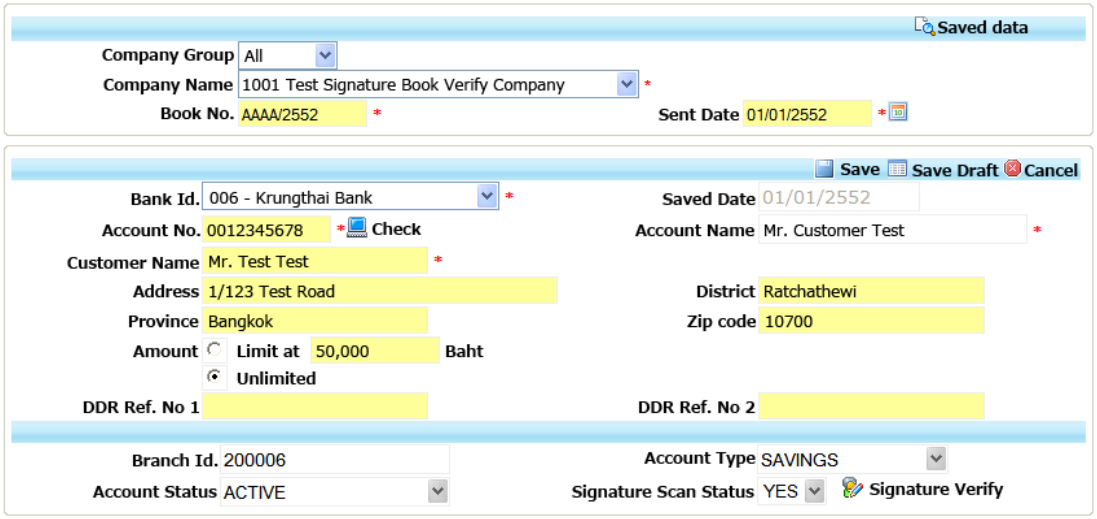
| Field Name | Type | Detail |
|---|---|---|
| Login | Button | **normlink:** Main Page |
| Clear | Button | |
| Recovery Password | Hyperlink | **text:** Forgot you password?<br>**normlink:** Recovery Page |

ภาพที่ 6 ตัวอย่างตารางรายละเอียดองค์ประกอบและเหตุการณ์/การกระทำ

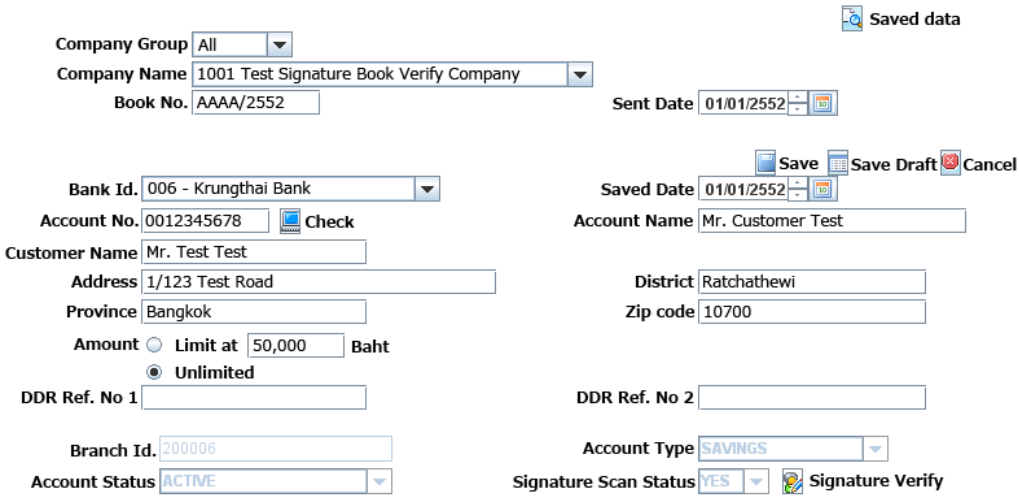
ภาพที่ 7 ตัวอย่างภาพหน้าจอเข้าสู่ระบบในเอกสาร
ซึ่งได้มาจากการบันทึกหน้าจอของต้นแบบที่ทำด้วยมือ

ภาพที่ 8 ตัวอย่างภาพหน้าจอเข้าสู่ระบบที่สร้างขึ้นมาจากระบบ
ด้วยข้อมูลที่สกัดมาจากเอกสารดังภาพที่ 6

ภาพที่ 7 และภาพที่ 8 กับ ภาพที่ 9 และภาพที่ 10 แสดงการเปรียบเทียบต้นแบบจากเอกสาร ในกรณีนี้ภาพตัวอย่างภาพที่ 7 เป็นภาพบันทึกหน้าจอของต้นแบบที่ทำด้วยมือ เนื่องจากผู้พัฒนาซอฟต์แวร์ใช้การสร้างต้นแบบด้วยมือ จึงต้องทำการสร้างต้นแบบก่อนเพื่อบันทึกภาพลงในเอกสารความต้องการในภายหลัง ส่วนภาพที่ 8 แสดงให้เห็นถึงผลลัพธ์ที่ได้จากระบบที่พัฒนาขึ้นเพื่อสร้างต้นแบบอัตโนมัติจากเอกสารข้อกำหนดความต้องการ จะเห็นได้ว่ารายละเอียดดังกล่าวค่อนข้างสอดคล้องกัน จะขาดก็เพียงองค์ประกอบปลีกย่อยที่ไม่ได้ระบุไว้ในเอกสารข้อกำหนดความต้องการเป็นข้อแตกต่างของการสร้างต้นแบบด้วยมือในแต่ละบุคคล




ภาพที่ 9 ตัวอย่างภาพหน้าจอการบันทึกข้อมูลบริษัทในเอกสาร
ซึ่งได้มาจากการบันทึกหน้าจอของต้นแบบที่ทำด้วยมือ




ภาพที่ 10 ตัวอย่างภาพหน้าจอการบันทึกข้อมูลบริษัทที่สร้างขึ้นมาจากระบบ
ด้วยข้อมูลที่สกัดมาจากเอกสารดังภาพที่ 6

### 3.2.3 จับคู่องค์ประกอบส่วนติดต่อผู้ใช้และรายละเอียดการควบคุมการเปลี่ยนหน้าจอ

จับคู่องค์ประกอบส่วนติดต่อผู้ใช้และรายละเอียดการควบคุมการเปลี่ยนหน้าจอกับส่วนติดต่อผู้ใช้ที่ได้สร้างขึ้น เพื่อให้ได้ออกมาเป็นต้นแบบสุดท้าย คือ ต้นแบบที่สามารถเปลี่ยนหน้าจอได้ ส่วนนี้โปรแกรมจะนำข้อมูลทั้งหมดจากขั้นตอนก่อนหน้ามาประมวลผลเพื่อใช้ในกระบวนการดังกล่าว

องค์ประกอบสามารถสนับสนุนในระบบคือ องค์ประกอบโดยทั่วไป ได้แก่เขตข้อมูลประเภทข้อความ พื้นที่ข้อมูลประเภทข้อความ เขตข้อมูลประเภทข้อความรหัส กล่องตัวเลือก กล่องตัวเลือกข้อเดี่ยว ปุ่ม เขตเชื่อมโยง รายการเลือกแบบหล่นลง รายการพื้นที่แถบเลื่อนหน้าจอ ตัวเลือกวันที่ ตัวเลือกไฟล์ ตัวเลือกสี ฯลฯ ตามแต่ผู้พัฒนาระบบจะประยุกต์เอาวิธีการนี้ไปใช้งาน

จากนั้นโปรแกรมจะจับคู่องค์ประกอบข้างต้น เช่น ปุ่มและเขตเชื่อมโยง กับการไหลของส่วนควบคุมหรือการเปลี่ยนแปลง (ทรานซิชัน) ที่ประมวลจากเอกสารข้อกำหนดความต้องการและข้อมูลเพิ่มเติมที่ผู้ใช้กรอกสู่ระบบภายหลัง ผลลัพธ์ที่ได้ก็คือ ต้นแบบที่สามารถทำงานได้ ถึงแม้ว่าต้นแบบดังกล่าวยังไม่สามารถทำงานได้อย่างเต็มที่ เช่น ไม่มีการตรวจสอบชื่อผู้ใช้และรหัสผู้ใช้กับฐานข้อมูลในกรณีเข้าสู่ระบบ แต่ต้นแบบสามารถแสดงการเปลี่ยนแปลงหน้าจอต่างๆ ได้ ซึ่งเป็นวัตถุประสงค์ของการทำต้นแบบ นอกจากนั้นต้นแบบดังกล่าวเป็นต้นแบบแบบบูรณาการ คือ เราสามารถนำต้นแบบดังกล่าวไปพัฒนาระบบจริง เช่น เชื่อมต่อกับฐานข้อมูลผู้ใช้ หากยังคงสภาพโครงของระบบที่สร้างขึ้นไว้ กล่าวคือ พัฒนาโปรแกรมต่อจากต้นแบบที่สร้างขึ้น โปรแกรมที่พัฒนาดังกล่าว สามารถนำมาปรับปรุงด้วยระบบพัฒนาต้นแบบนี้ต่อไปได้ และสามารถปรับเปลี่ยนการทดสอบให้สมจริงขึ้นได้ ตามตัวอย่างเดิมก็คือ ระบบต้นแบบที่ปรับปรุงต่อจะสามารถเชื่อมโยงกับฐานข้อมูลได้ จากโค้ดในส่วนที่ผู้พัฒนาได้พัฒนาเพิ่มเข้าไปภายหลัง

ข้อดีของการนำระบบที่พัฒนาต่อจากต้นแบบมาใช้งานกับระบบพัฒนาต้นแบบแบบต่อยอดก็คือ การที่สามารถจะปรับปรุงอัตโนมัติจากข้อกำหนดความต้องการที่ถูกปรับเปลี่ยนเล็กๆ น้อยๆ

ภายหลัง ตามขั้นตอนที่จะนำเสนอต่อไปในหัวข้อการปรับปรุงต้นแบบหรือสามารถนำข้อแก้ไขจากโปรแกรมต้นแบบไปปรับปรุงเอกสารตามขั้นตอนที่จะนำเสนอต่อไปในหัวข้อการปรับปรุงเอกสาร คุณลักษณะดังกล่าวช่วยส่งเสริมความยั่งยืนในการพัฒนาผลิตภัณฑ์ในโครงการซอฟต์แวร์ คงความสอดคล้องต้องกันของผลิตภัณฑ์ต่างๆ ของโครงการ

### 3.3 ส่วนการสร้างคำสั่งทดสอบ (Test Script)

Yourdon และ Constantine [11] กล่าวไว้ว่า ในการสร้างต้นแบบ การจัดการความผิดพลาดมีสี่องค์ประกอบ ความผิดพลาดต้องถูกตรวจสอบด้วยกระบวนการบางอย่าง การกระทำในทันทีเป็นสิ่งที่ต้องทำเพื่อจัดการกับความผิดพลาดนั้น สุดท้ายแล้วผู้บัญญัติอาจจัดการแก้ไขหรือปรับแก้ให้ถูกต้อง ความผิดพลาดควรถูกตรวจจับได้ให้เร็วที่สุดเท่าที่ทำได้ นั่นคือ ใกล้กับต้นทางที่ส่วนติดต่อส่วนใดส่วนหนึ่ง อ้างตามคำกล่าวข้างต้น คุณสมบัติการทดสอบตนเองได้ถูกเพิ่มไว้ในต้นแบบที่สร้างขึ้นด้วยวิธีการที่นำเสนอในรายงานนี้ นั่นคือ ส่วนติดต่อผู้ใช้และการเคลื่อนไหวที่สร้างขึ้นโดยอัตโนมัติจากข้อกำหนดความต้องการเหล่านี้ เอื้อให้สามารถดำเนินผ่านแต่ละส่วนติดต่อผู้ใช้เพื่อทวนสอบกลุ่มความต้องการโดยอัตโนมัติด้วยการดำเนินไปตามการไหลของส่วนควบคุม การทดสอบด้วยเครื่องจักรสถานะจำกัดได้ถูกนำเสนอในจุดนี้

เริ่มจากการสร้างคำสั่งทดสอบต้นแบบจากข้อมูลต่างๆ ที่ทราบ โดยใช้หลักการของเครื่องจักรสถานะและอาศัยการท่องกราฟ (Graph Traversal) ประกอบ โดยเทียบเคียงหลักการของเครื่องจักรสถานะจำกัดซึ่งเปลี่ยนแปลงได้ ($M_V$) ได้ดังตารางที่ 8

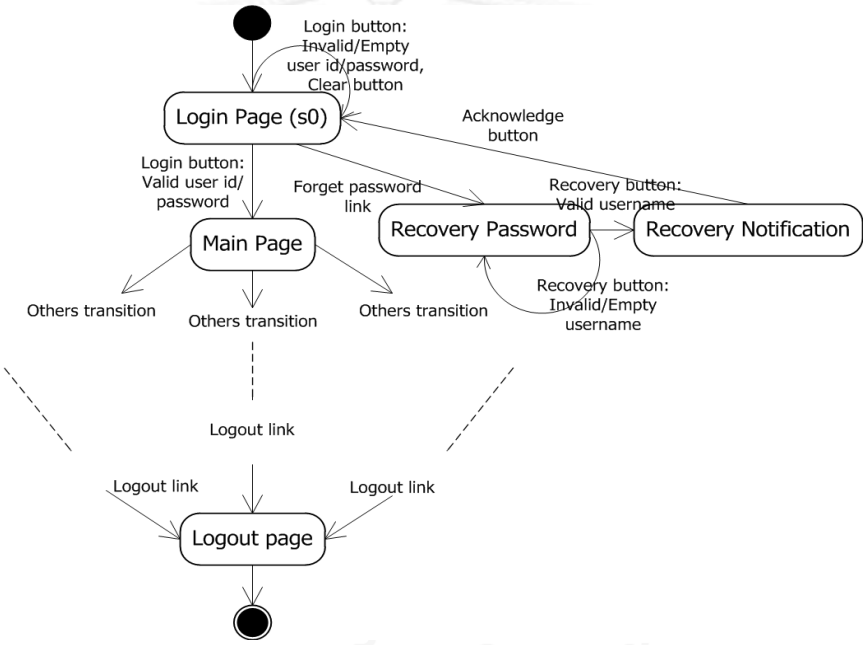
ตารางที่ 8 การเทียบเคียงหลักการของเครื่องจักรสถานะจำกัดซึ่งเปลี่ยนแปลงได้

| รายการ | ตัวแปร | รายการเทียบเท่า |
|---|---|---|
| เซตจำกัดของสถานะ | S | เซตของหน้าจอ |
| สถานะเริ่มต้น | $s_0 \in S$ | หน้าจอเริ่มต้น |
| เซตจำกัดของข้อมูลเข้า | I | เซตของการกระทำที่เกิดกับองค์ประกอบส่วนติดต่อผู้ใช้ที่ใช้ควบคุม |
| เซตของตัวแปร | V | เซตของค่าที่เป็นไปได้ของตัวแปร ซึ่งกำหนดตามข้อมูลเข้าจากองค์ประกอบส่วนติดต่อผู้ใช้ |

ส่วนติดต่อผู้ใช้และการเปลี่ยนแปลงของมันสามารถสร้างขึ้นโดยอัตโนมัติจากข้อกำหนดความต้องการทางด้านซอฟต์แวร์ ดังนั้น หากต้นแบบของส่วนติดต่อผู้ใช้นั้นเอื้ออำนวยต่อการทดสอบโดยอัตโนมัติแล้ว คำสั่งทดสอบย่อมสามารถสร้างขึ้นโดยอัตโนมัติเช่นกันโดยใช้เทคนิคเดียวกันในการมองหาความสัมพันธ์ขององค์ประกอบแล้วนำมากำหนดเป็นการเปลี่ยนแปลงขององค์ประกอบที่จะทดสอบ ให้ครอบคลุมในแต่ละความสัมพันธ์ดังกล่าว

ภาพที่ 11 แสดงแผนภาพตัวอย่างสถานะของส่วนติดต่อผู้ใช้ที่วิเคราะห์ออกมาได้จากส่วนของต้นแบบในภาพที่ 8 S0 หมายถึงหน้าเข้าสู่ระบบหากเราแจกแจงออกมาจะเห็นว่าประกอบด้วยสามองค์ประกอบตัวนำเปลี่ยนสถานะ คือ ปุ่มเข้าสู่ระบบปุ่มล้างข้อมูล และเขตเชื่อมโยงการลืม

รหัสผ่านผู้ใช้ด้วยเหตุที่เขตข้อมูลชื่อผู้ใช้และเขตข้อมูลรหัสผ่านเป็นเขตข้อมูลที่มีความสำคัญระดับจำเป็น ปุ่มเข้าสู่ระบบต้องตรวจสอบข้อมูลที่จำเป็นเหล่านี้ว่ามีครบถ้วน ถ้ามันไม่เป็นไปตามเงื่อนไขเหล่านั้นก็จะโยนไปสู่การจัดการข้อยกเว้น ซึ่งในระบบทดสอบที่สร้างขึ้นก็จะเป็นฟังก์ชันฟังก์ชันหนึ่งที่ถูกเรียก ที่ผู้พัฒนาจะเขียนการทำงานใดๆ ลงไปก็ได้ เพื่อสนับสนุนกระบวนการต้นแบบแบบพัฒนาได้ สังเกตได้ว่าหน้าเข้าสู่ระบบ (S0) มีสามการเปลี่ยนสถานะ หนึ่งจากหน้าเข้าสู่ระบบไปยังหน้าเข้าสู่ระบบดังเดิม สองจากหน้าเข้าสู่ระบบไปยังหน้าหลักและสุดท้ายจากหน้าเข้าสู่ระบบไปยังหน้าการกู้คืนรหัสผ่าน การเปลี่ยนแปลงแบบแรกถูกเรียกได้โดยปุ่มเข้าสู่ระบบด้วยข้อมูลในเขตข้อมูลที่ต้องระบุไม่ครบถ้วนหรือข้อมูลไม่ถูกต้อง หรือจากการกดที่ปุ่มที่ล้างข้อมูล การเปลี่ยนแปลงแบบที่สองเกิดจากการกดที่ปุ่มเข้าสู่ระบบด้วยข้อมูลที่ถูกต้องในเขตข้อมูลต่างๆ และผลการเปลี่ยนแปลงแบบที่ที่สามจากการกดเขตเชื่อมโยงการลืมรหัสผ่านโดยไม่ต้องกรอกข้อมูลในเขตข้อมูลต่างๆ




ภาพที่ 11 แผนภาพตัวอย่างสถานะของส่วนติดต่อผู้ใช้ตามภาพที่ 8

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

การประยุกต์ใช้อัลกอริทึมของการท่องกราฟจะสร้างชุดของเส้นทางการทดสอบ โดยเส้นทางการทดสอบแต่ละเส้นแสดงถึงแต่ละกรณีทดสอบ และชุดของกรณีทดสอบแต่ละชุดประกอบด้วยคำสั่งทดสอบต่างๆคำสั่งทดสอบมีส่วนประกอบคือ

1. องค์ประกอบที่กระตุ้นการเปลี่ยนแปลง
2. การกระตุ้นการเปลี่ยนแปลง (การกระทำ)
3. หน้าแหล่งที่มา
4. หน้าปลายทาง

โดยกำหนดเครื่องสถานะจำกัดที่ อาศัยพื้นฐานความรู้ที่ศึกษาจาก[6] เราปรับใช้เครื่องจักรสถานะจำกัดซึ่งเปลี่ยนแปลงได้ (Variable Finite State Machines — VFSM) ในงานวิจัยโดยเครื่องจักรสถานะจำกัดซึ่งเปลี่ยนแปลงได้สามารถเปลี่ยนเป็นเครื่องจักรสถานะจำกัดอื่นๆ ที่เทียบเท่า

ได้ และสามารถนำมาประยุกต์ใช้กับการทดสอบส่วนติดต่อผู้ใช้ได้อย่างดี นอกจากนั้นองค์ประกอบหลักของเครื่องจักรนี้มียังขนาดเล็กและเป็นธรรมชาติเมื่อเทียบกับเครื่องจักรสถานะแบบอื่นๆ นอกจากนี้ ยังมีความยืดหยุ่นสูง ทำงานได้เร็ว และถูกต้องแม่นยำ เพราะเป็นการออกแบบเพื่อนำเสนอกระบวนการของโปรแกรมโดยเฉพาะ

คำสั่งทดสอบจะถูกใช้โดยหน่วยการทดสอบที่สร้างเหตุการณ์ป้อนเข้าแบบระบบสำหรับการทดสอบแบบอัตโนมัติโดยการควบคุมเมาส์และคีย์บอร์ดที่จะกระทำไปยังแต่ละตำแหน่งตามคำสั่งทดสอบที่สร้างขึ้น

การเปลี่ยนแปลงทั้งหมดของส่วนควบคุมการเปลี่ยนแปลงที่มีอยู่ในต้นแบบเป็นเรื่องยากที่จะทดสอบโดยมนุษย์ และหากมันสามารถทำได้มันก็เป็นเรื่องยากที่จะตรวจสอบเกณฑ์ต่างๆ ได้อย่างครอบคลุม

สำหรับการแสดงกราฟเราเลือกใช้ไลบรารี่กราปป้า (Grappa)เพื่อใช้งานโปรแกรม การจัดการมุมมองกราฟกราฟวิซ Graphviz ซึ่งเป็นโครงการโอเพ่นซอสในการจัดการงานนำเสนอกราฟโครงข่ายโดยการวิจัยของ AT&T เพื่อให้การแสดงผลภาพกราฟมีความสวยงามและเป็นระบบมากยิ่งขึ้น หน่วยย่อย “dot” ซึ่งเป็นตัวกรองการจัดการตัวแปรของมุมมองกราฟตัวหนึ่งในโปรแกรมถูกนำมาใช้งานในการจัดการด้านต่างๆ อาธิเช่น การกำหนดตำแหน่งและขนาดของแต่ละองค์ประกอบบนกราฟ โดยตัวแปลงเส้นทางจะรับรายการของวัตถุองค์ประกอบหน้าจอแสดงผลจากขั้นตอนก่อนหน้าที่สกัดข้อมูลที่เป็นประโยชน์มาจากเอกสาร แต่ละองค์ประกอบหน้าจอประกอบด้วยเขตข้อมูลต่างๆ แต่ละเขตข้อมูลประกอบด้วยสารสนเทศที่เกี่ยวข้องกับเขตข้อมูลนั้นๆ รวมถึงการเปลี่ยนแปลงหน้าจอที่เกี่ยวข้องดังที่ได้กล่าวมา เราสามารถนำสารสนเทศเหล่านี้มาวาดเป็นกราฟความสัมพันธ์องค์ประกอบที่จะแสดงความเชื่องโยงของหน้าจอและเขตข้อมูลกับการเปลี่ยนแปลงขององค์ประกอบเหล่านั้น กราฟนี้จะถูกนำไปใช้โดยส่วนสร้างเส้นทางทดสอบที่จะท่องไปตามเส้นทางต่างๆ ทั้งหมด วิธีการท่องกราฟทางลึกก่อน depth-first traversal (DFT) ถูกนำมาใช้ในงานนี้ ผลลัพธ์ของกระบวนการดังกล่าว ทำให้เราได้รายการของเส้นทางกราฟทีเทียบเท่ากับเส้นทางการทดสอบ

การแก้ไขใดๆ ที่เกิดขึ้นกับกราฟนี้จะปรับปรุงกลับไปที่รายการหน้าจอสำหรับการเปลี่ยนแปลงเอกสารที่เกี่ยวข้องและตัวต้นแบบ ตัวตรวจจับการกระทำจะคอยส่งเหตุการณ์ที่เปลี่ยนแปลงสู่กราฟ ดังนั้น หากมีการเปลี่ยนแปลงเกิดขึ้นส่วนสร้างเส้นทางการทดสอบจะสร้างเส้นทางการทดสอบใหม่ขึ้นมาตามการเปลี่ยนแปลงนั้น

### 3.3.1 การท่องกราฟ

วิธีการท่องกราฟทางลึกก่อน (DFT) ถูกนำมาใช้เป็นเทคนิคในการท่องกราฟในงานวิจัยนี้ เส้นทางการทดสอบถูกสร้างขึ้นจากการรวมทางที่เป็นไปได้ทั้งหมดจากเส้นที่เริ่มต้นจากโหนดเริ่มต้นไปยังโหนดใบ ตัวอย่างเช่นเส้นทางการทดสอบa -> b -> cและ a -> b-> dจะถูกสร้างขึ้นจากวิธีการท่องกราฟทางลึกก่อนด้วยลำดับของเส้น a -> b, b -> cและ b -> d ตามลำดับ รหัสเทียมของวิธีการท่องกราฟทางลึกก่อนนำเสนอได้ดังต่อไปนี้:

```
ทำ DFT (โหนดเริ่มต้น, รายการที่ว่างเปล่า, รายการเส้นทางที่ว่างเปล่า)

ฟังก์ชัน DFT (โหนด n, เส้นทางหนึ่ง onePath, รายการเส้นทางผลลัพธ์ resultPathList)
    ถ้า n ถูกประทับ
        เพิ่ม onePath ไปยังresultPathList
        กลับ
จบเงื่อนไขถ้า
    ประทับบน n (สำหรับทำเครื่องหมายว่าผ่านมันไปแล้ว)
วงวนสำหรับแต่ละเส้นเชื่อมe ในรายการเส้นเชื่อมของn
เส้นทางปัจจุบัน nowPath:= คัดลอกจากเส้นทาง onePath
เพิ่มเส้นเชื่อมe ไปยังnowPath
ทำ DFT (โหนดด้านหัวของ e, nowPath, resultPath)
จบวงวน
จบฟังก์ชัน
```

เนื่องจากกราฟความสัมพันธ์องค์ประกอบที่แสดงหน้าจอและเขตข้อมูลกับการเปลี่ยนแปลงขององค์ประกอบเหล่านั้นอาจจะไม่ถูกท่องไปทุกเส้นเชื่อมจากการเพิ่มต้นด้วยโหนดเพียงโหนดเดียว คือ เส้นเชื่อมไม่จำเป็นต้องเชื่อมทุกโหนดเข้าด้วยกัน จึงมีการแก้ไขเพิ่มเติมจากอัลกอริทึมของ DFT ดังต่อไปนี้:

```
ค่าตั้งต้น
n := โหนดเริ่มต้นที่ต้องการ
el := รายการของเส้นเชื่อมทั้งหมด
remain:= รายการของเส้นเชื่อมทั้งหมด
output := รายการเปล่า
โดย output คือ รายการผลลัพธ์สุดท้ายที่จะได้จากอัลกอริทึม

การทำงาน
วงวน ขณะที่รายการใน remainยังคงมากกว่า 0 รายการ
    all1 := DFT จาก n (ผลลัพธ์คือ รายการของเส้นทาง DFT เริ่มจาก n)
stamp := คีย์ประทับจาก n (องค์ประกอบที่ผ่านรอบนี้มีคีย์ประทับเดียวกัน)
วงวนสำหรับแต่ละเส้นเชื่อมe ที่อยู่ใน remain
        ถ้าคีย์ประทับของ e เป็นคีย์เดียวกันกับ stamp
            ลบ e ออกจากรายการ remain
จบเงื่อนไขถ้า
จบวงวน
```

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

วงวนสำหรับแต่ละเส้นทางp ในรายการ output
//เป็นการลบเส้นทางเก่าที่เป็นชุดย่อยของเส้นทางใหม่
ถ้าคีย์ประทับเส้นเชื่อมแรกของ p คือ คีย์ประทับตั้งต้น
ลบ p จากoutput
จบเงื่อนไขถ้า
จบวงวน
ถ้าจำนวนรายการใน remain ยังคงมากกว่า 0 รายการ
n: = โหนดหางของเส้นเชื่อมแรกในรายการ remain
จบเงื่อนไขถ้า
จบวงวน

ได้รายการ output คือ รายการเส้นทางทดสอบที่ครบถ้วน
หมายเหตุ คีย์ประทับตั้งต้นคือ ค่าตั้งต้นตัวแปร อาจกล่าวได้ว่าไม่เคยถูกตั้งค่ามาก่อน

### 3.3.2 การเชื่อมโยงเข้ากับทฤษฎีเครื่องจักรสถานะจำกัดซึ่งเปลี่ยนแปลงได้

แต่ละสถานะจะเป็นสถานภาพของหน้าและผลจากตัวแปรขาเข้าที่มีผลต่อการเปลี่ยนแปลงจากสถานะปัจจุบัน ในกรณีที่เรียบง่ายนั้นแต่ละหน้าจะเป็นหนึ่งสถานะ ซึ่งหมายถึง การเปลี่ยนแปลงที่มีอยู่ไม่ได้ขึ้นอยู่กับค่าเข้าก่อนจากสถานะก่อน แต่หากโปรแกรมมีความซับซ้อนก็สามารถแปลงโดยใช้กฎของเครื่องจักรมัวร์ ในกรณีนี้หนึ่งสถานะหนึ่งไม่ได้เป็นหน้าจอ แต่เป็นหน้าจอ x รูปแบบตัวแปรเข้า โดยสถานะทั้งหมดจะเท่ากับผลรวมของจำนวนหน้าจอ x รูปแบบตัวแปรเข้าของแต่ละหน้าจอ

วิธีการในการจัดการปัญหาทำได้โดยการตั้งค่าตรวจสอบ (variable identifier) ให้แก่แต่ละหน้าจอ ค่าตรวจสอบที่แตกต่างกันแสดงถึงสถานะที่แตกต่างกัน ส่งผลให้รูปแบบการเปลี่ยนแปลงสามารถแตกต่างกันได้ ตัวอย่างเช่น การเปลี่ยนแปลงจากหน้าเข้าสู่ระบบไปหน้าหลักด้วยปุ่มเข้าสู่ระบบ หากชื่อบัญชีผู้ใช้สถานะผู้ดูแลระบบส่งผลให้ปุ่ม ‘a’ ในหน้าหลักเชื่อมโยงไปยังหน้าจอ A ในขณะที่หากใช้ชื่อบัญชีผู้ใช้สถานะลูกค่า การกดปุ่ม ‘a’ ในหน้าหลักจะเชื่อมโยงไปยังหน้าจอ Bโดยหน้าจอ A และหน้าจอ B เป็นคนละหน้าจอกัน จะได้ว่าค่าตรวจสอบที่ตั้งให้กับหน้าจอหลักทั้งสองกรณีแตกต่างกัน ส่งผลให้หน้าจอหลักมีด้วยกัน 2 สถานะคือ รูปแบบ Admin และรูปแบบ Customer สามารถแจกแจงการเปลี่ยนแปลงที่กล่าวถึงข้างต้นได้ดังนี้

1. 1การเปลี่ยนแปลงจากหน้าเข้าสู่ระบบไปยังหน้าหลัก x identifier1 (Admin) ด้วยปุ่มเข้าสู่ระบบ
2. 1การเปลี่ยนแปลงจากหน้าเข้าสู่ระบบไปยังหน้าหลัก x identifier2 (Customer) ด้วยปุ่มเข้าสู่ระบบ
3. 1การเปลี่ยนแปลงจากหน้าหลัก x identifier1 (Admin) สู่หน้า A ด้วยปุ่ม‘a’
4. 1การเปลี่ยนแปลงจากหน้าหลัก x identifier2 (Customer) สู่หน้า B ด้วยปุ่ม‘a’

แสดงให้เห็นว่ามีทั้งหมด 5 สถานะด้วยกันจากทั้งหมด 4 หน้าจอที่แตกต่างกัน วิธีการดังกล่าวถูกประยุกต์ใช้ไปยังกรณีอื่นๆ เช่นกัน

3.3.2.1 กรณีสถานะแบบต่างๆ ของเครื่องจักร

จากทฤษฎีที่กล่าวมาข้างต้น ทำให้เราสามารถเชื่อมโยงการทำงานของต้นแบบเข้ากับเครื่องจักรสถานะจำกัดซึ่งเปลี่ยนแปลงได้ โดยเชื่อมโยงระหว่างการเปลี่ยนสถานะกับเงื่อนไขกระบวนการตรวจสอบต้นแบบตามเอกสารข้อกำหนดความต้องการ ซึ่งสามารถอธิบายการเชื่อมโยงได้ด้วยแผนภาพกรณีตัวอย่างแบบต่างๆ ดังต่อไปนี้

กรณีที่ 1:กรณีเบื้องต้น – 1 หน้าจอคือ 1 สถานะ

กำหนดให้มีหน้าเข้าสู่ระบบและหน้าหลัก ผู้ใช้กำลังอยู่ในหน้าเข้าสู่ระบบ

1. ผู้ใช้กดปุ่มเข้าสู่ระบบด้วยค่าที่ถูกต้องในเขตข้อมูลต่างๆ

หน้าจอเปลี่ยนเข้าสู่หน้าหลัก

2. ผู้ใช้กดปุ่มเข้าสู่ระบบด้วยค่าที่ไม่ถูกต้องในเขตข้อมูลต่างๆ

หน้าจอไม่เปลี่ยนแปลง

ขณะผู้ใช้อยู่ในหน้าหลัก

3. ผู้ใช้กดเขตเชื่อมโยงออกจากระบบ

หน้าจอเปลี่ยนเข้าสู่หน้าออกจากระบบ

ดังนั้น กรณีนี้ประกอบด้วยสามสถานะ Login, Main และ Logout

การเปลี่ยนแปลง Login -1-> Main, Login -2-> Login, Main -3-> Logout ดังแสดงในภาพที่ 12

ภาพที่ 12 เครื่องจากสถานะกรณีตัวอย่างที่ 1

กรณีที่ 2: 1 หน้าจอเป็นได้หลายสถานะ

โปรแกรมเริ่มด้วยหน้า ‘A’

หากผู้ใช้เข้าสู่ระบบด้วยบัญชีพนักงานทั่วไป หน้าจอจะเปลี่ยนไปยังหน้า ‘B’ เมื่อผู้ใช้กดเลือกเขตเชื่อมโยงการแก้ไขบัญชี หน้าจอจะเปลี่ยนไปอีกครั้งยังหน้า ‘C’

ในขณะที่หากผู้ใช้เข้าสู่ระบบด้วยบัญชีพนักงานแผนกบริหารจัดการทรัพยากรมนุษย์ หน้าจอจะเปลี่ยนไปยังหน้า ‘B’ เมื่อผู้ใช้กดเลือกเขตเชื่อมโยงการแก้ไขบัญชี หน้าจอจะเปลี่ยนไปอีกครั้งยังหน้า ‘D’ ซึ่งสามาระแก้ไขบัญชีรายละเอียดของพนักงานคนอื่นๆ ได้

โดยที่เขตเชื่อมโยงการแก้ไขบัญชีในทั้งสองกรณีเป็นเขตเชื่อมโยงเดียวกัน แต่ด้วยเงื่อนไขที่ต่างกันส่งผลให้เชื่อมโยงไปยังหน้าจอที่ต่างกัน

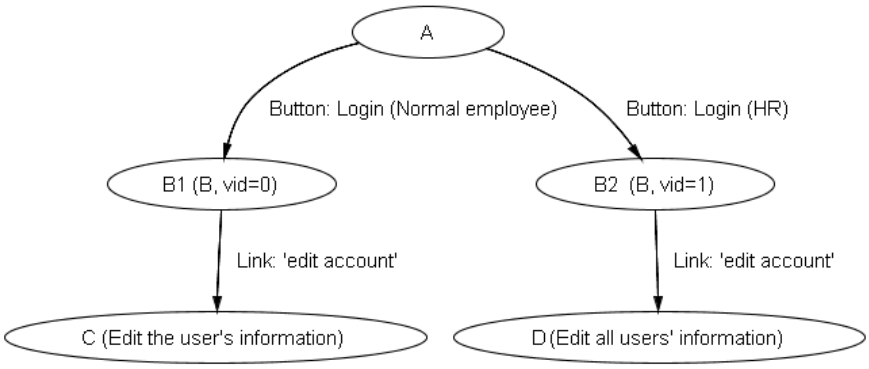
สามารถกำหนดสถานะและการเปลี่ยนแปลงต่างๆ ได้ดังนี้

4. A -> B1 (Normal employee)
5. A -> B2 (HR)
6. B1 -> C
7. B2 -> D

โดย B1 คือ หน้า Bกับค่าตรวจสอบเท่ากับ 0 และ B2 คือหน้า B กับค่าตรวจสอบเท่ากับ 1

ดังนั้น กรณีนี้ประกอบด้วยสามสถานะ Login, Main และ Logout

การเปลี่ยนแปลง Login -1-> Main, Login -2-> Login, Main -3-> Logout ดังแสดงในภาพที่ 13




ภาพที่ 13 เครื่องจากสถานะกรณีตัวอย่างที่ 2

กรณีที่ 3: หน้าจอไม่เปลี่ยนแปลงในขณะที่สถานะเปลี่ยนแปลงไป

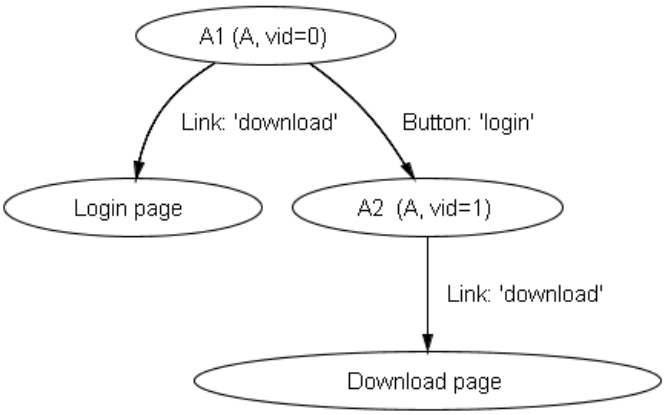
ตัวอย่างเหตุการณ์ หน้า ‘A’ ประกอบด้วยเขตข้อมูลเพื่อเข้าสู่ประบบ หลังจากผู้ใช้กดเข้าสู่ระบบ หน้าจอไม่เปลี่ยนแปลงไป แต่คุณสมบัติของเขตเชื่อมโยงบางอันเปลี่ยนแปลงไป

ตัวอย่างเช่น เขตเชื่อมโยงการดาวน์โหลดที่ต้องการการเข้าสู่ระบบ ซึ่งหากผู้ใช้เข้าสู่ระบบแล้วมันจะเชื่อมโยงไปยังหน้าดาวน์โหลด ถ้าหากผู้ใช้ยังไม่ได้เข้าสู่ระบบจะไม่เกิดการเปลี่ยนแปลงใดๆ ขึ้น

สามารถกำหนดสถานะและการเปลี่ยนแปลงต่างๆ ได้ดังนี้

8. A1 -(‘login’ button)-> A2
9. A1 -(‘download’ link)-> Login page
10. A2 -(‘download’ link)-> Download page

ดังแสดงในภาพที่ 14




ภาพที่ 14 เครื่องจากสถานะกรณีตัวอย่างที่ 3

หลักการเดียวกันนี้สามารถนำไปใช้กับโครงสร้างที่ซับซ้อน ตัวอย่างเช่น เฟรมภายในและองค์ประกอบซับซ้อนอื่นๆ ดังแสดงให้เห็นในกรณีต่อไปนี้

กรณีที่ 4: การประยุกต์ใช้กับเฟรมภายใน

ตัวอย่างเหตุการณ์ หน้า ‘A’ ประกอบด้วยเขตข้อมูลเพื่อเข้าสู่ประบบ หลังจากผู้ใช้กดเข้าสู่ระบบ หน้าจอไม่เปลี่ยนแปลงไป แต่คุณสมบัติของเขตเชื่อมโยงบางอันเปลี่ยนแปลงไป

ตัวอย่างเช่น เขตเชื่อมโยงการดาวน์โหลดที่ต้องการการเข้าสู่ระบบ ซึ่งหากผู้ใช้เข้าสู่ระบบ

หน้าจอ ‘A’ มีเฟรมภายใน ‘B’ หลังจากเข้าสู่ระบบเฟรมภายในจะถูกเปลี่ยนเป็นเฟรม ‘C’ ซึ่งหมายถึงเส้นทางการทดสอบจะต้องมีการเปลี่ยนแปลงไป ดังนั้น หน้าจอ A จึงไม่ใช่สถานะสถานะเดียวอีกต่อไป

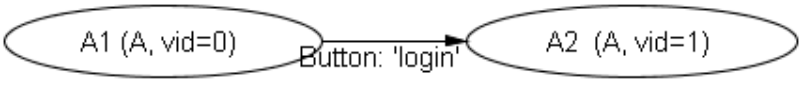
สามารถกำหนดสถานะและการเปลี่ยนแปลงต่างๆ ได้ดังนี้

11. A1 -(‘login’ button)-> A2

A1 คือ หน้า ‘A’ กับเฟรมภายใน ‘B’

A2 คือ หน้า ‘A’ กับเฟรมภายใน ‘C’

ดังแสดงในภาพที่ 15



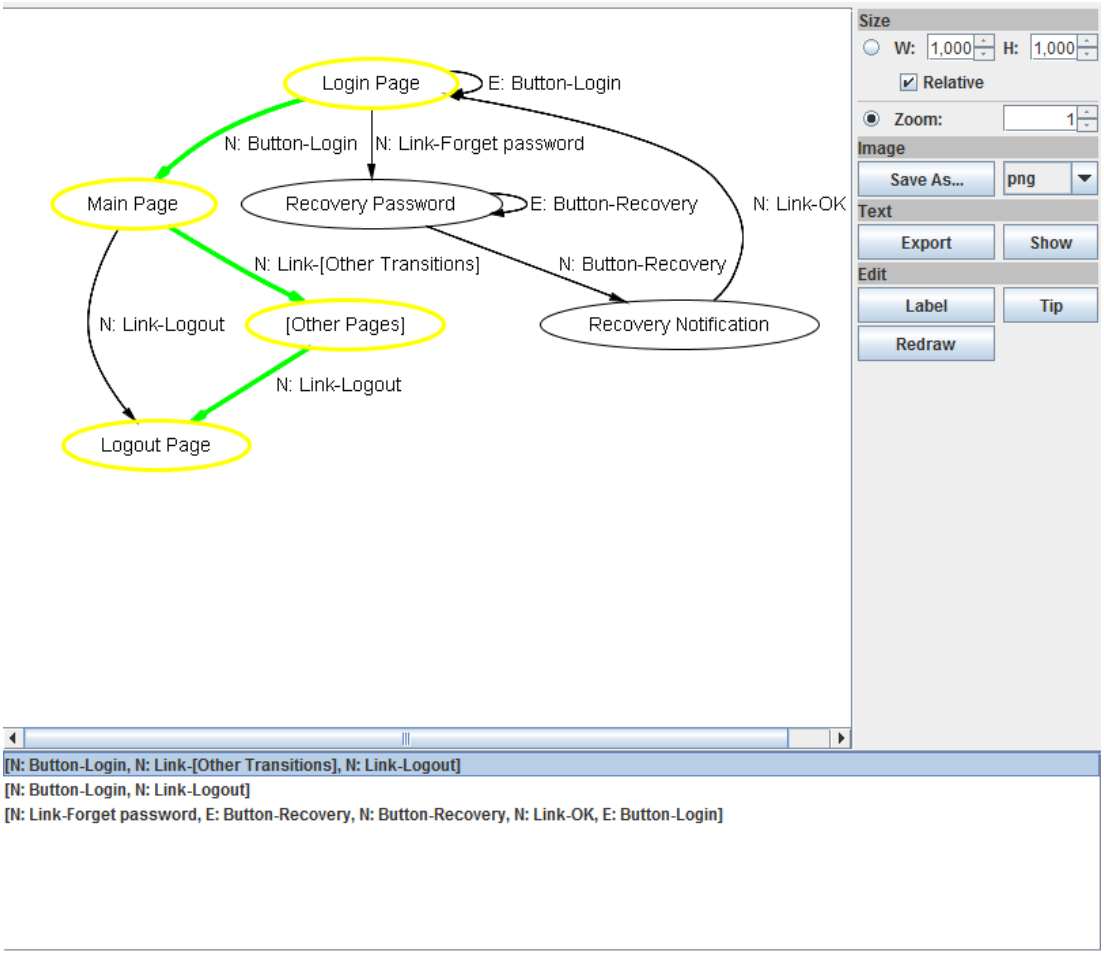

ภาพที่ 15 เครื่องจากสถานะกรณีตัวอย่างที่ 4

การเก็บรายการสถานะในรูปแบบ XML

```
<statelist>
<state>
        //Page ID
        <pid></pid>
        //Variation ID
        <vid></vid>
        <transition>
                //Transition ID
                <tid></tid>
                //Element ID
                <eid></eid>
                //Destination page ID
                <dpid></dpid>
                //Destination variation ID
                <dvid></dvid>
//Other attributes
                ...
        </transition>
</state>
//Other states
<state>...</state>
</statelist>
```

จากกระบวนการดังกล่าวข้างต้นสามารถนำมาสร้างเครื่องจักรสถานะจำกัดของต้นแบบตามภาพที่ 11 และแสดงให้เห็นถึงเส้นทางการทดสอบต่างๆ ได้ดังที่ปรากฏในภาพที่ 16 โดยมีรูปแบบแบบข้อความที่แสดงถึงกราฟและเส้นทางการทดสอบแสดงดังภาพที่ 17 รูปแบบแบบข้อความนี้เป็นข้อมูลที่ส่งไปให้แก่โปรแกรมกราฟวิซในการประมวลผลจัดมุมมองของกราฟให้สวยงาม

CHULALONGKORN UNIVERSITY




ภาพที่ 16 กราฟตัวอย่างแสดงเส้นทางทดสอบที่สัมพันธ์กับภาพที่ 11

หลังจากนี้รายการองค์ประกอบที่ประกอบด้วยประเภทของเขตข้อมูล การเปลี่ยนแปลง และเงื่อนไขการเปลี่ยนแปลง พร้อมๆ กับเส้นทางการทดสอบที่ได้ จะถูกส่งต่อไปยังส่วนการสร้างคำสั่งทดสอบ ซึ่งจะสร้างคำสั่งทดสอบที่เข้าคู่กับการกระทำที่เหมาะสมต่อองค์ประกอบแต่ละชิ้น ตามที่จะกล่าวในหัวข้อที่ 3.4 เรื่องการทดสอบอัตโนมัติ

Chulalongkorn University

```
digraph all_paths {
  graph [d
    minbox = "0,0,2,1",
    orientation = portrait,
    rankdir = TB,
    fontcolor = black,
    fontname = TimesRoman,
    ranksep = "0.75",
    clusterrank = local,
    fontstyle = normal,
    mclimit = 1,
    style = __default__,
    color = white,
    nodesep = "0.25",
    margin = "0.5,0.5",
    bb = "0,0,593,300",
    fontsize = 14
  ];

  node [d
    fontstyle = normal,
    height = "0.5",
    style = __default__,
    color = black,
    width = "0.75",
    fontcolor = black,
    label = "\N",
    fontname = TimesRoman,
    shape = ellipse,
    fontsize = 14,
    pos = "0,0"
  ];

  edge [d
    fontstyle = normal,
    weight = 1,
    style = __default__,
    color = black,
    dir = forward,
    minlen = 1,
    fontcolor = black,
    fontname = TimesRoman,
    fontsize = 14
  ];

  … node …
  N0_1330048752084 [d
    width = "1.7596",
    label = "Login Page",
    pos = "235,282"
  ];
  N5_1330048752085 [d
    width = "1.9013",
    label = "Logout Page",
    pos = "77,18"
  ];
  N3_1330048752085 [d
    width = "1.6728",
    label = "Main Page",
    pos = "60,194"
```

```
 ];                                    pos = "e,61.087,35.552
 N4_1330048752085 [d                48.8,176.22 36.373,155.44
   width = "2.011",                 19.274,119.01 29,88 34.109,71.709
   label = "[dOther Pages]",        44.574,55.896 54.433,43.528"
   pos = "216,106"                   ];
 ];                                    N1_1330048752085 ->
                                    N2_1330048752085 [d
… edge …                              lp = "434.5,150",
 N0_1330048752084 ->                  label = "N: Button-Recovery",
N0_1330048752084 [d                   pos = "e,421,122.53
   lp = "370,282",                  275.58,177.62 313.6,163.22
   label = "E: Button-Login",       370.54,141.65 411.48,126.14"
   pos = "e,291.78,273.88            ];
291.78,290.12 305.66,289.52           N0_1330048752084 ->
316,286.81 316,282 316,278.47      N1_1330048752085 [d
310.43,276.07 301.96,274.8"           lp = "315,238",
 ];                                   label = "N: Link-Forget
                                    password",
 N3_1330048752085 ->                  pos = "e,235,212.08 235,263.6
N5_1330048752085 [d                 235,251.75 235,235.82 235,222.29"
   lp = "81.5,106",                  ];
   label = "N: Link-Logout",        }
```

ภาพที่ 17 ตัวอย่างรูปแบบแบบข้อความที่แสดงถึงกราฟและเส้นทางการทดสอบ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

**3.4 ส่วนการทดสอบอัตโนมัติ**

งานวิจัยนี้ให้ความสำคัญในเรื่องของการทวนสอบและการยันสอบใช้งานได้จริง (V & V) เพื่อที่จะตรวจจับและลบข้อบกพร่องของโครงการให้เร็วที่สุดเท่าที่เป็นไปได้ ในงานศึกษาที่เขียนขึ้นนั้นมีหลายกระบวนวิธีหรือวิธีการด้วยกันในด้านกิจกรรมการทวนสอบและการยันสอบใช้งานได้จริง เช่น

1. การทดสอบโดยมนุษย์: การตรวจตลอดงาน (walkthough), การสังเกตโดยผู้ใช้และการแจกจ่ายสำเนาทดสอบ เป็นต้น
2. การทดสอบเลียนแบบมนุษย์: การจับภาพและทำซ้ำ, วิธีอัลกอริทึมทางพันธุกรรมและการทดสอบแบบสุ่ม เป็นต้น

3. การทดสอบอัตโนมัติ: การทดสอบด้วยเครื่องจักรสถานะ และการทดสอบด้วยการขับเคลื่อนเป้าหมาย เป็นต้น

แต่ละวิธีการมีข้อดีและข้อเสียต่างกันไป การทดสอบโดยมนุษย์ใช้ทรัพยากรสูง ใช้แรงงานมากและมีความไม่แน่นอนสูง การทดสอบเลียนแบบมนุษย์คงข้อเสียบางอย่างจากวิธีการทดสอบโดยมนุษย์ เพราะเป็นการเลียนแบบ แต่ลดการใช้ทรัพยากร เช่น ด้านแรงงานและเวลา เป็นต้น อย่างไรก็ตาม ทั้งสองวิธีการข้างต้นมีข้อเสียด้านการครอบคลุมกฎเกณฑ์และเส้นทางการทดสอบต่างๆ โดยเฉพาะหากระบบที่สร้างเป็นระบบที่ใหญ่ในระดับหนึ่ง การทดสอบโดยมนุษย์ให้ครบถ้วนนั้นแทบเป็นไปไม่ได้ในเวลาที่จำกัด งานวิจัยนี้เลือกวิธีการทดสอบด้วยเครื่องจักรสถานะ ซึ่งเป็นวิธีที่เอื้ออำนวยต่อความเป็นอัตโนมัติ อย่างไรก็ตาม วิธีการแบบอัตโนมัติมีข้อเสียคือ การทดสอบอาจจะไม่เป็นไปตามที่ผู้ใช้งานคาดหวัง หากผู้ใช้งานและระบบอัตโนมัตินั้น มีการสื่อสาร ความเข้าใจต่อระบบที่ไม่ตรงกัน ซึ่งในงานวิจัยนี้แก้ไขโดยให้มนุษย์สามารถเข้ามามีส่วนร่วมในการตรวจสอบการทดสอบได้โดยทันที จึงได้เลือกใช้การทดสอบด้วยวิธีการสร้างเหตุการณ์ข้อมูลเข้าแก่ระบบแต่กำเนิด (Native System Input Events)

### 3.4.1 วิธีการสร้างเหตุการณ์ข้อมูลเข้าแก่ระบบแต่กำเนิด (Native System Input Events)

วิธีการสร้างเหตุการณ์ข้อมูลเข้าแก่ระบบแต่กำเนิด คือ การสร้างเหตุการณ์ข้อมูลเข้าจริง ตัวอย่างเช่น การลากเคอร์เซอร์ การส่งค่าการพิมพ์ (Key Code) และการกดเมาส์แบบต่างๆ ทั้งหมดนี้จะถูกสร้างเหตุการณ์เสมือนกับที่องค์ประกอบส่วนติดต่อผู้ใช้ในการรับข้อมูลเข้าส่งมาเมื่อผู้ใช้กระทำกับองค์ประกอบส่วนติดต่อผู้ใช้ในการรับข้อมูลเข้านั้นๆ

การปฏิบัติการทดสอบตามคำสั่งทดสอบ โดยใช้วิธีการสร้างเหตุการณ์ข้อมูลเข้าแก่ระบบแต่กำเนิดของงานวิจัยนี้ จะทดสอบการเปลี่ยนสถานะต่างๆ จนครบทุกการเปลี่ยนสถานะด้วยวิธีการดังกล่าว ข้อดีของวิธีนี้คือ ผู้ตรวจสอบหรือผู้ตรวจทานโครงการสามารถเข้ามามีส่วนร่วมในการตรวจสอบการทดสอบต้นแบบโดยอัตโนมัติของระบบไปด้วยได้อย่างเต็มที่ เหมือนกับมนุษย์ทดสอบเอง ช่วยลบข้อด้อยในเรื่องความครอบคลุมด้วยการทดสอบโดยมนุษย์ออกไป และแก้ข้อเสียของระบบอัตโนมัติที่การทดสอบอาจไม่เป็นไปตามคาดหวังเนื่องจากมีความเข้าใจผิดของระบบ หากผู้ตรวจสอบพบการทดสอบที่ผิดพลาด สามารถกลับไปแก้ไขในคำสั่งทดสอบได้ เพื่อให้ระบบทำการทดสอบอย่างถูกต้อง

วิธีการสร้างเหตุการณ์ข้อมูลเข้าแก่ระบบแต่กำเนิดประกอบด้วย การเลื่อนเมาส์ การกดเมาส์ การปล่อยเมาส์ และการส่งค่าพิมพ์ต่างๆ ที่เลียนแบบการส่งข้อมูลเข้าของมนุษย์ โดยชุดของการดำเนินการที่กำหนดไว้ในแต่ละลำดับย่อยของเหตุการณ์ การดำเนินการแต่ละอย่างสามารถถือเป็นปฏิสัมพันธ์จากมนุษย์ ตัวอย่างการดำเนินการมีดังนี้

1. การกดองค์ประกอบ:

เลื่อนเมาส์>คลิกเมาส์ (ประกอบด้วย กดเมาส์ >ปล่อยเมาส์)

2. การพิมพ์ลงสู่องค์ประกอบ:

เลื่อนเมาส์>คลิกเมาส์>ส่งค่าการพิมพ์

3. การเลือกองค์ประกอบ (เช่นกล่องตัวเลือก รายการตัวเลือก ฯลฯ):
เลื่อนเมาส์>คลิกเมาส์
4. ลากองค์ประกอบ:
กดเมาส์>เลื่อนเมาส์>ปล่อยเมาส์

แต่ละองค์ประกอบมีการกำหนดค่าเริ่มต้นโดยอ้างอิงจากการดำเนินการโดยสากลทั่วไป ตัวอย่างการดำเนินการต่างๆ ของแต่ละองค์ประกอบ ดังตารางที่ 9

ตารางที่ 9 ตัวอย่างการดำเนินการโดยสากลทั่วไปกับแต่ละองค์ประกอบ

| **รายการ** | **การดำเนินการโดยสากลทั่วไป** |
|---|---|
| ปุ่ม | การกดองค์ประกอบ (กดที่ปุ่ม) |
| เขตข้อมูลประเภทข้อความ | การพิมพ์ลงสู่องค์ประกอบ |
| เขตเชื่อมโยง | การกดองค์ประกอบ |
| กล่องสำหรับเลือก | การเลือกองค์ประกอบ |

โค้ดของคำสั่งทดสอบจะได้รับการส่งผ่านยังหน่วยทดสอบเพื่อดำเนินการตามรูปแบบการดำเนินการที่กำหนดไว้ด้วยส่วนสร้างเหตุการณ์ข้อมูลเข้าแก่ระบบแต่กำเนิดของระบบ

นอกจากนี้ การส่งเหตุการณ์ข้อมูลเข้าแก่ระบบแต่กำเนิดของแต่ละการดำเนินงานหลักอาจจะมีการดำเนินการย่อยตัวอย่างเช่นการหาสถานที่ตั้งและขนาดขององค์ประกอบการคำนวณจุดการกด และการเรียกข้อความที่ถูกต้องตรงกับนิพจน์ปรกติที่ระบุให้ใช้ตรวจสอบองค์ประกอบนั้นๆ เป็นต้น

ตัวอย่างของการดำเนินการในหน้าเข้าสู่ระบบในกรณีที่การเข้าสู่ระบบประสบความสำเร็จประกอบด้วย

5. กรอกชื่อผู้ใช้ที่ถูกต้อง
    5.1) การเรียกชื่อผู้ใช้ที่ถูกต้องตรงกับนิพจน์ปรกติ (จัดเตรียมโดยการตั้งค่า)
    5.2) รับที่ตั้งและขนาดของเขตข้อมูลข้อความชื่อผู้ใช้ในการคำนวณจุดกด
    5.3) กดที่จุดกด
    5.4) ส่งค่าพิมพ์ชื่อผู้ใช้ที่ถูกต้องจาก5.1)
6. กรอกรหัสผ่านที่ถูกต้อง
    6.1) การเรียกรหัสผ่านที่ถูกต้อง
    6.2) รับที่ตั้งและขนาดของเขตข้อมูลรหัสผ่านในการคำนวณจุดกด
    6.3) กดที่จุดกด
    6.4) ส่งค่าพิมพ์ชื่อผู้ใช้ที่ถูกต้องจาก6.1)
7. กดที่ปุ่มเข้าสู่ระบบ
    7.1) รับที่ตั้งและขนาดของปุ่มเข้าสู่ระบบในการคำนวณจุดกด

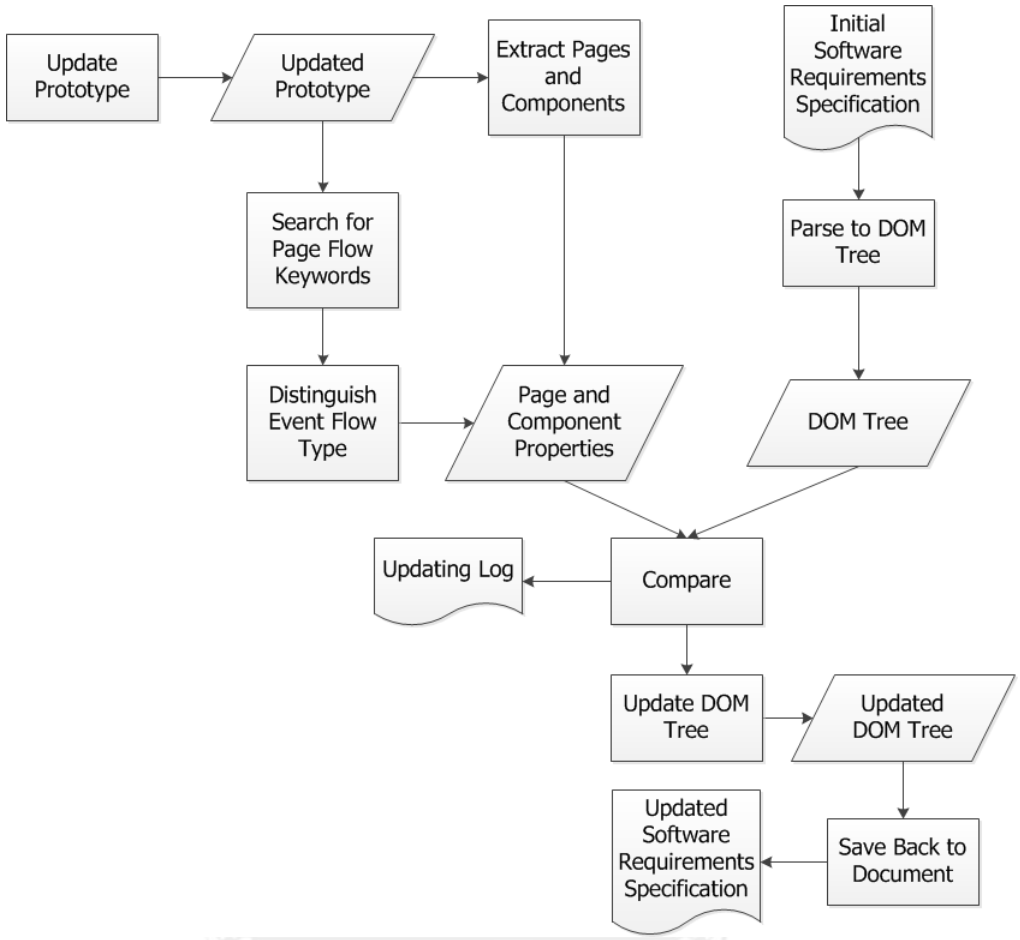
7.2) กดที่จุดกด

8. การเปลี่ยนผ่านไปยังหน้าหลัก
   8.1) ถ้าหน้าไม่ถูกเปลี่ยนไปยังหน้าคาดหวัง
      8.1.1) [เรียกการจัดการกระบวนการที่ไม่ถูกต้อง]
      8.1.2) ซ่อนหน้าปัจจุบัน
      8.1.3) แสดงหน้าคาดหวัง
   8.2) ถ้ารหัสค่าตรวจสอบของหน้าไม่ตรงกับรหัสค่าตรวจสอบเปลี่ยนแปลงที่คาดหวัง
      8.2.1) [เรียกการจัดการกระบวนการที่ไม่ถูกต้อง]
      8.2.2) เปลี่ยนรหัสของการเปลี่ยนแปลงเป็นรหัสค่าตรวจสอบที่คาดหวัง

รหัสเทียมแสดงขั้นตอนของการทดสอบด้วยตนเองของต้นแบบอธิบายได้ดังต่อไปนี้

```
all := รายการของรายการการดำเนินการ

วงวนสำหรับรายการการดำเนินการแต่ละรายการ ol ใน all

วงวนสำหรับการดำเนินการแต่ละการดำเนินการ op ในรายการ ol

        ทำงานตามการดำเนินการ op

        ถ้า op คือ การดำเนินการสุดท้ายใน ol

                ถ้าหน้าจอเปลี่ยนเป็นหน้าที่ไม่ตรงกับที่คาดหวัง

                        เรียกการจัดการกระบวนการที่ไม่ถูกต้อง

                จบเงื่อนไขถ้า

                แล้วถ้ารหัสค่าตรวจสอบของหน้าไม่ตรงกับที่คาดหวัง

                        เรียกการจัดการกระบวนการที่ไม่ถูกต้อง

                จบเงื่อนไขถ้า

        จบเงื่อนไขถ้า

    จบวงวน

จบวงวน
```

### 3.5 ส่วนการปรับปรุงเอกสาร




ภาพที่ 18 ขั้นตอนการปรับปรุงเอกสารข้อกำหนดความต้องการเพื่อความสอดคล้องกับข้อมูลที่เปลี่ยนแปลงไปเนื่องมาจากการแก้ไขต้นแบบ

การปรับปรุงเอกสารประกอบด้วยขั้นตอนต่างๆ ได้ดังนี้

1. หลังจากต้นแบบถูกปรับปรุง เราจะนำต้นแบบดังกล่าวไปสกัดเอาข้อมูลต่างๆ เพื่อได้ออกมาเป็นคุณสมบัติของหน้าจอและองค์ประกอบย่อย ซึ่งกระบวนการดังกล่าวประกอบด้วย
   1.1) สกัดองค์ประกอบของส่วนติดต่อผู้ใช้
   1.2) ค้นหาคำสำคัญที่แสดงการเปลี่ยนแปลงของหน้าจอในโค้ดต้นแบบที่ถูกปรับปรุงขึ้นใหม่
   1.3) แยกแยะการเปลี่ยนแปลงต่างๆ ของเหตุการณ์
2. การสกัดข้อมูลจากต้นไม้ DOM จากเอกสารตั้งต้น (เอกสารที่ต้องการปรับปรุง) กระบวนการนี้กระทำเพื่อจะได้ปรับปรุงเอกสารตั้งต้นได้ตรงตำแหน่ง จึงต้องค้นหาแต่ละส่วนของเอกสาร ว่าข้อมูลใดวางอยู่ตำแหน่งไหน จากนั้นจึงนำข้อมูลใหม่ไปปรับปรุงได้ตรงจุด วิธีการที่ใช้จะให้วิธีการเดียวกับการสกัดข้อมูลจากเอกสารเพื่อสร้างต้นแบบ

3. เปรียบเทียบข้อมูลที่ได้จากต้นแบบและข้อมูลที่ได้จากเอกสารตั้งต้น เพื่อค้นหาสิ่งที่ถูกเปลี่ยนแปลง เพิ่มเติมหรือลบออกไป
4. ทำการปรับปรุงบันทึก (Log) ของการเปลี่ยนแปลง
5. ทำการปรับปรุงต้นไม้ DOM ของเอกสาร จะได้ออกมาเป็นต้นไม้ที่ปรับปรุงแล้ว
6. บันทึกต้นไม้ใหม่เป็นรูปแบบของเอกสารที่ต้องการ เช่น “docx” เป็นต้น จะได้เอกสารข้อกำหนดความต้องการที่ถูกปรับปรุงจากเอกสารตั้งต้น

### 3.6 ส่วนการปรับปรุงต้นแบบ

การปรับปรุงต้นแบบอาศัยความรู้ในการประมวลโค้ดต้นทางของโปรแกรม ในงานวิจัยนี้ผู้วิจัยได้ใช้แพคเกจซอสของซัน (Sun's source package) และไลบราอาปาเช่แจกซ์มี (Apache JaxMe Library) ซึ่งสามารถประมวลโค้ดต้นทางแต่ละหน้าออกเป็นองค์ประกอบของภาษาจาวา จากนั้นเราจะสามารถเขียนหน่วยประมวลผลเพื่อรับรู้ถึงแต่ละองค์ประกอบของต้นแบบได้

ตัวอย่างเช่น ระบบทดสอบที่ผู้วิจัยสร้างขึ้นกำหนดให้แต่ละหน้าส่วนติดต่อผู้ใช้ แบ่งเป็นแต่ละคลาสแยกออกจากกัน โดยมีรหัสกำหนดแสดงถึงการเป็นหน้าของส่วนติดต่อผู้ใช้ หลังจากที่เราทำการอ่านโค้ดต้นแบบด้วยแพคเกจซอสของซัน เราจะได้ข้อมูลของโค้ดออกมาเป็นวัตถุต่างๆ และรายละเอียดของแต่ละวัตถุนั้นว่ามีความสัมพันธ์กันอย่างไรเราจึงนำข้อมูลเรานั้นมาระบุว่าต้นแบบนั้นๆ มีหน้าส่วนติดต่อผู้ใช้กี่หน้า อะไรบ้าง และมีองค์ประกอบใดบ้างในแต่ละหน้าได้ โดยคลาสในภาษาจาวาประกอบไปด้วยส่วนพื้นฐานต่างๆ 10 ส่วน ตามตารางที่ 10

ตารางที่ 10 ส่วนพื้นฐานต่างๆ ของคลาสในภาษาจาวา

| รายการ | คำอธิบาย |
|---|---|
| Magic Number | ตัวอย่างเช่น 0xCAFEBABE |
| Version of Class File Format | หมายเลขรุ่นหลักและย่อยของคลาส |
| Constant Pool | กลุ่มค่าคงที่สำหรับคลาส |
| Access Flags | ตัวอย่างเช่นสถานะคลาสเป็น abstract, static ฯลฯ |
| This Class | ชื่อของคลาสนี้ |
| Super Class | ชื่อของซุปเปอร์คลาส |
| Interfaces | อินเตอร์เฟสต่างๆ ภายในคลาส |
| Fields | เขตข้อมูลต่างๆ ภายในคลาส |
| Methods | เมธอดต่างๆ ภายในคลาส |
| Attributes | แอททริบิวท์ต่างๆ ของคลาส ตัวอย่างเช่น ชื่อเอกสารโค้ดต้นทาง ฯลฯ |

วิธีการปรับปรุงต้นแบบที่ผู้วิจัยนำเสนอจะเป็นวิธีการเดียวกันกับในหัวข้อที่ 3.5 เรื่องการปรับปรุงเอกสาร โดยเริ่มจากการเรียนรู้ต้นแบบตามวิธีที่ได้กล่าวมาข้างต้นการเรียนรู้เอกสารโดย

วิธีการสกัดข้อมูลจากเอกสารที่หัวข้อ 3.1 จากนั้นนำมาวิเคราะห์ข้อแตกต่างของทั้งสองแหล่ง จะต่างออกไปก็เพียงในครั้งนี้เราจะทำการปรับปรุงข้อแตกต่างมาที่ต้นแบบ แทนที่จะปรับปรุงไปที่เอกสารตามหัวข้อที่ 3.5

การปรับปรุงต้นแบบนั่นสามารถทำได้ในหลายกรณีด้วยกัน ตัวอย่างเช่น ตำแหน่งองค์ประกอบ ขนาดขององค์ประกอบ ชื่อของเขตข้อมูลต่างๆ ค่าตั้งต้นของเขตข้อมูลต่างๆ นิพจน์ปรกติสำหรับการทดสอบข้อมูลในเขตข้อมูลต่างๆ เป็นต้น



จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# บทที่ 4
# การพัฒนาระบบ

## 4.1 สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนา

สภาพแวดล้อมและเครื่องมือที่ใช้ในการพัฒนาระบบการสร้างต้นแบบลักษณะทดสอบได้โดยอัตโนมัติจากข้อกำหนดความต้องการที่ถูกเก็บเกี่ยว มีดังต่อไปนี้

### 4.1.1 ฮาร์ดแวร์

1. หน่วยประมวลผลกลาง (CPU) อินเทล คอร์ 2 ดูโอ 2.53 กิกะเฮิร์ต (CPU Intel Core 2 Duo 2.53 GHz)
2. หน่วยความจำ 4 กิกะไบต์ (4 GB RAM)
3. ฮาร์ดดิสก์ความจุ 500 กิกะไบต์ (500GB Hard Disk)

### 4.1.2 ซอฟต์แวร์

1. ระบบปฏิบัติการไมโครซอฟต์ วินโดวส์ 8 โปร (Microsoft Windows 8 Prol)
2. โปรแกรมสภาพแวดล้อมประมวลผลจาวา 7 (Java Runtime Environment 7)
3. โปรแกรมชุดเครื่องมือพัฒนาจาวา 7 (Java Development Kit 7)
4. โปรแกรมอีคลิปส์ ไอดีอี เจอีอี จูโน (Eclipse IDE JEE Juno)
5. โปรแกรมเน็ตบีนส์ ไอดีอี 7 (NetBeans IDE 7)
6. โปรแกรมกราฟวิซ 2(Graphviz 2)
7. โปรแกรมอโดบี โฟโตชอป ซีเอส 5 (Adobe Photoshop CS5)
8. แพคเกจซอสของซัน (Sun's source package)
9. ไลบรารี่อาปาเช่แจกซ์มี (Apache JaxMe Library)
10. ไลบรารี่กราปป้า (Grappa Library)

### 4.1.3 ขั้นตอนการติดตั้งซอฟต์แวร์

ซอฟต์แวร์และเครื่องมือสำหรับการพัฒนาจะต้องถูกติดตั้งให้เรียบร้อยก่อนการพัฒนาระบบโดยมีลำดับการติดตั้งดังต่อไปนี้

1. ติดตั้งระบบปฏิบัติการไมโครซอฟต์ วินโดวส์ 8 โปร
2. ติดตั้งโปรแกรมสภาพแวดล้อมประมวลผลจาวา 7
3. ติดตั้งโปรแกรมชุดเครื่องมือพัฒนาจาวา 7
4. ติดตั้งโปรแกรมอีคลิปส์ ไอดีอี เจอีอี จูโน
5. ติดตั้งโปรแกรมเน็ตบีนส์ ไอดีอี 7
6. ติดตั้งโปรแกรมกราฟวิซ 2
7. ติดตั้งโปรแกรมอโดบี โฟโตชอป ซีเอส 5
8. นำเข้าแพคเกจซอสของซันสู่ระบบที่พัฒนา

9. นำเข้าไลบรารี่อาปาเช่แจกซ์มีสู่ระบบที่พัฒนา
10. นำเข้าไลบรารี่กราปป้าสู่ระบบที่พัฒนา

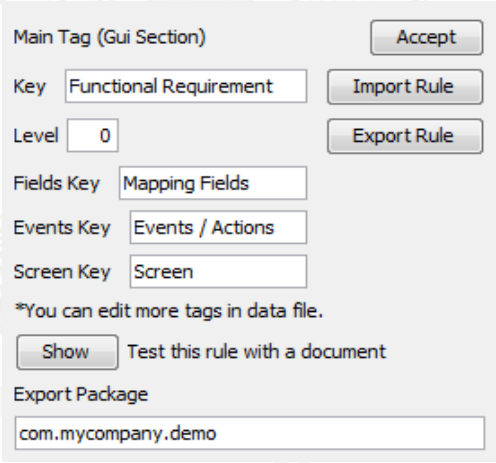
## 4.2 ขั้นตอนการพัฒนาระบบ

การพัฒนาระบบระบบในแต่ละส่วน ประกอบด้วยส่วนสำคัญ ดังนี้

### 4.2.1 การพัฒนาขั้นตอนวิธีการเตรียมข้อมูลเพื่อนำเข้า

รูปแบบของข้อมูลเข้าที่จะทำการสกัดจะสกัดจากต้นไม้ของ Document Object Model (DOM) ดังที่กล่าวไว้โดยละเอียดในเรื่องส่วนนำเข้าและสกัดข้อมูล โดยมีการใช้งานเอกสารย่อยดังที่แสดงไว้ในตารางที่ 5 อย่างไรก็ตาม ผู้ใช้งานไม่จำเป็นต้องมีความรู้ด้านต่างๆ เหล่านั้น ผู้ใช้งานสามารถกำหนดกฎการนำเข้าข้อมูลตามความรู้พื้นฐานด้านป้ายบอกตำแหน่ง จากระบบที่พัฒนาขึ้น ระบบตัวอย่างที่พัฒนาขึ้นจัดทำเพื่อรองรับการนำของเอกสารประเภท docx ของโปรแกรมไมโครซอฟต์เวิร์ด โดยผู้จัดทำแนะนำให้ใช้การสร้างรูปแบบ (Style) แบบลำดับชั้นของโปรแกรมไมโครซอฟต์เวิร์ด แทนการใช้ย่อหน้าปกติ ในส่วนของป้ายบอกตำแหน่งต่างๆ

ระบบเริ่มจากให้ผู้ใช้ระบุกฎการนำเข้าต่างๆ ในการนำเข้า (Import Spec) ดังภาพที่ 19




ภาพที่ 19 หน้าจอการระบุกฎการนำเข้า

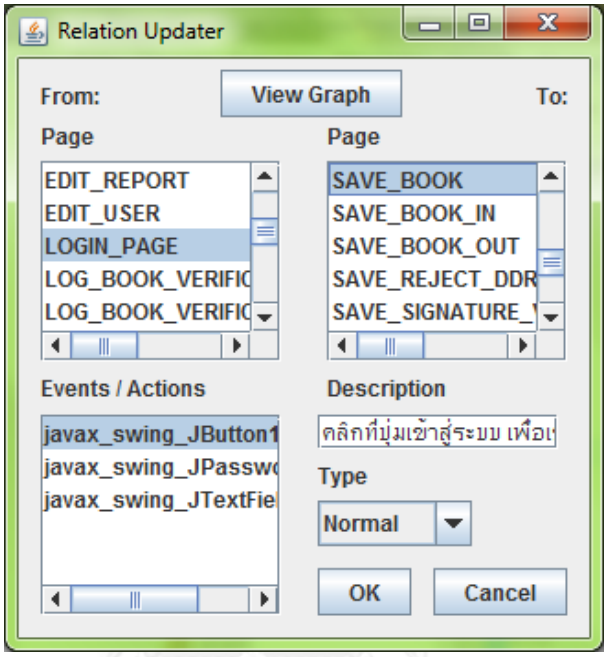
### 4.2.2 การแก้ไขปรับปรุงข้อมูลต้นแบบ

กระบวนการแก้ไขปรับปรุงข้อมูลต้นแบบจะเริ่มหลังจากผู้ใช้ระบุกฎการนำเข้า และเลือกเอกสารนำเข้าเรียบร้อยแล้ว เตรียมตัวเพื่อสร้างเป็นต้นแบบออกมา ในขั้นตอนนี้จะมีรายการแสดงรายละเอียดต่างๆ ที่สกัดได้จากเอกสารตามหัวข้อส่วนการสร้างต้นแบบ ผู้ใช้สามารถแก้ไข เพิ่มเติมข้อมูลต่างๆ ที่ไม่มีในเอกสาร แต่เป็นที่ต้องการในการสร้างต้นแบบขึ้นมาได้ ข้อมูลต่างๆ เหล่านี้จะถูกส่งออกเป็นไฟล์ไว้ด้วยเพื่อให้ผู้ใช้สามารถตรวจสอบการเปลี่ยนแปลงได้ ซึ่งภาพของระบบที่สร้างในส่วนของการรับข้อมูลต่างๆ นำเสนอไว้พร้อมคำอธิบายขั้นตอนการใช้งานในภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

### 4.2.3 การกำหนดความสัมพันธ์

ผู้ใช้สามารถกำหนดความสัมพันธ์ คือ การส่งผ่านของแต่ละหน้าจอได้ หากไม่ใช่ข้อมูลที่จะกำหนดในเอกสารหรือต้องการแก้ไขเพิ่มเติม ข้อมูลต่างๆ เหล่านี้จะถูกส่งออกเป็นไฟล์ไว้เพื่อให้ตรวจสอบภายหลังได้ ซึ่งระบบที่สร้างมีการรับข้อมูลต่างๆ ดังภาพที่ 20




ภาพที่ 20 หน้าจอการแก้ไขปรับปรุงความสัมพันธ์

### 4.2.4 การสร้างต้นแบบ

การสร้างต้นแบบจะเริ่มหลังจากผู้ใช้ระบุกฎการนำเข้า เลือกเอกสารนำเข้า แก้ไขปรับปรุงข้อมูลต่างๆ ที่ต้องการเรียบร้อยแล้ว ระบบจะทำการสร้างต้นแบบขึ้นมาโดยอัตโนมัติไปยังแพคเกจที่ผู้ใช้กำหนด โดยระบบที่สร้างขึ้นอ้างอิงชื่อเมธอดเบื้องต้นและมาตรฐานต่างๆ ตามโปรแกรมเน็ตบีนส์ไอดีอี และสร้างเมธอดหลักให้แต่ละคลาสในการแสดงหน้าจอต้นแบบได้เพื่ออำนวยความสะดวกแก่ผู้ใช้งาน กระบวนการต่างๆ ดำเนินการสร้างระบบตามวิธีการที่ได้นำเสนอ

### 4.2.5 การทดสอบต้นแบบ

ผู้ใช้งานสามารถสั่งทดสอบระบบปัจจุบันได้ทันที โดยการทดสอบระบบใช้วิธีการสร้างเหตุการณ์ข้อมูลเข้าแก่ระบบแต่กำเนิด คือ มีการใช้งานระบบขึ้นมาจริงๆ โดยอัตโนมัติให้ผู้ทดสอบได้ตรวจสอบทบทวน และเนื่องจากระบบตัวอย่างที่สร้างขึ้นต้องการสนับสนุนการปรับปรุงต้นแบบในภายหลังการสร้างคำสั่งทดสอบจะสร้างใหม่ได้โดยอัตโนมัติโดยอ้างอิงจากซอสโค้ด โดยระบบที่สร้างขึ้นกำหนดไว้ที่เมธอด “meetRequirements” เพื่อให้ผู้พัฒนาตัวระบบงานที่สร้างจากต้นแบบดังกล่าว สามารถเพิ่มเติมการตรวจสอบอื่นๆ เพิ่มเติมภายหลังได้ และจะถูกนำมาใช้จริงหากดำเนินการทดสอบต้นแบบในภายหลังด้วย โดยตัวระบบจะสร้างการตรวจสอบดังกล่าวไว้แล้วในขั้นการสร้างต้นแบบ อาศัยข้อมูลว่าเป็นเขตข้อมูลที่จำเป็น (ต้องมีข้อมูล) หรือไม่และประเภทข้อมูลที่

ถูกต้องจากประเภทข้อมูลหรือนิพจน์ปรกติที่กำหนดตามแต่ละเขตข้อมูลคืออะไรตามภาพที่ 21 แสดงให้เห็นถึงการตรวจสอบเขตข้อมูลประเภทข้อความ “TextField” ตามรูปแบบชื่อ “REGEX_NUMBER” ซึ่งกำหนดไว้กับนิพจน์ปรกติบรรยายตัวเลข และอำนวยความสะดวกในการตรวจสอบเขตข้อมูลรหัส “PasswordField” พร้อมคำแนะนำแก่ผู้ใช้

```
public boolean meetRequirements() {
  boolean passReqs = true;
  String javax_swing_JTextField1Text =
      javax_swing_JTextField1.getText();
  passReqs&=refUtils.UtilsM.isValid
      (javax_swing_JTextField1Text,"REGEX_NUMBER");
  char[] javax_swing_JPasswordField1Password =
      javax_swing_JPasswordField1.getPassword();
  //TODO compares by Arrays.equals
  //(javax_swing_JPasswordField1Password, correct) for safety.

  java.util.Arrays.fill
  (javax_swing_JPasswordField1Password, '\0');

  return passReqs;
}
```

ภาพที่ 21 ซอสโค้ดจากต้นแบบที่ระบบสร้างขึ้นในส่วนการตรวจสอบเขตข้อมูลต่างๆ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# บทที่ 5
# การทดสอบระบบ

การทดสอบระบบสร้างต้นแบบลักษณะทดสอบได้โดยอัตโนมัติจากข้อกำหนดความต้องการที่ถูกเก็บเกี่ยวมีจุดประสงค์เพื่อตรวจสอบความถูกต้องของผลลัพธ์ที่ได้จากการใช้งานในแต่ละฟังก์ชันในส่วนของการสร้างต้นแบบและส่วนของทดสอบต้นแบบรวมถึงทำการทดสอบการนำไปใช้เพื่อสนับสนุนกระบวนการพัฒนาซอฟต์แวร์ด้วยกรณีศึกษา นอกจากนี้ ยังมีส่วนช่วยในการตรวจทานข้อผิดพลาดที่เกิดขึ้นและค้นหาข้อจำกัดต่างๆ ของระบบ

## 5.1 การทดสอบความถูกต้องของฟังก์ชันการทำงานของระบบ

การทดสอบด้านประสิทธิผลเป็นการปฏิบัติการทดสอบการทำงานและผลลัพธ์ว่าถูกต้องตามฟังก์ชันการทำงาน (ความต้องการเชิงหน้าที่) ที่ได้กำหนดไว้ โดยวิธีการทดสอบหน้าที่การทำงานแบบกล่องดำ (Black Box Testing) ตามกรณีทดสอบที่ได้ออกแบบไว้ โดยมีขั้นตอนปฏิบัติและผลการทดสอบระบบด้วยกรณีทดสอบดังตารางที่ 11

ตารางที่ 11 ผลการทดสอบความถูกต้องของฟังก์ชันการทำงานของระบบ

| หมายเลขกรณีทดสอบ | การทดสอบ | คำอธิบาย | ผลการทดสอบคาดหวัง | ผลการทดสอบ |
|---|---|---|---|---|
| UAT1 | ทดสอบการนำเข้าเอกสาร | เลือกเมนู “File” และเลือกเมนู “Load Document” จากนั้นเลือกเอกสาร | ระบบโหลดข้อมูลจากเอกสารเข้าสู่โปรแกรม ระบบแสดงกล่องให้กำหนดค่าเกี่ยวกับเอกสารนำเข้า | ผ่าน |
| UAT2 | ทดสอบการกำหนดค่าเกี่ยวกับเอกสารนำเข้า | กำหนดค่าเกี่ยวกับเอกสารนำเข้าลงในกล่องกำหนดค่าหรือเลือกข้อกำหนดที่เคยบันทึกไว้เดิม | ระบบนำค่าดังกล่าวไปใช้ | ผ่าน |
| UAT3 | ทดสอบการแก้ไขข้อกำหนดเอกสารนำเข้า | เลือกเมนู “Option” และเลือกเมนู “Import Spec” จากนั้นเลือก Add/Edit/Delete ทำการแก้ไขเพิ่มเติมข้อกำหนดเอกสาร | ข้อกำหนดเอกสารนำเข้าใหม่ปรากฏให้เลือกใช้ มีการแก้ไขหรือถูกลบออกไปตามการทำงานที่เลือก | ผ่าน |

| | | | | |
|---|---|---|---|---|
| | | นำเข้า | | |
| UAT4 | ทดสอบการนำเข้าข้อกำหนดเอกสารนำเข้า | กดปุ่ม “Import Rule” | มีข้อกำหนดที่นำเข้าสู่ระบบ | ผ่าน |
| UAT5 | ทดสอบนำออกข้อกำหนดเอกสารนำเข้า | กดปุ่ม “Export Rule” | มีกล่องให้บันทึกข้อกำหนดเอกสารนำเข้า | ผ่าน |
| UAT6 | ทดสอบการแก้ไขค่าประเภทของข้อมูลในการตรวจสอบช่องใส่ค่าของต้นแบบ | เลือกเมนู “Option” และเลือกเมนู “Input Type” จากนั้นเลือก Add/Edit/Delete ทำการแก้ไขเพิ่มเติมชื่อประเภทข้อมูลและนิพจน์ปรกติในการตรวจสอบ | ประเภทข้อมูลใหม่ปรากฏให้เลือกใช้ มีการแก้ไขหรือถูกลบออกไปตามการทำงานที่เลือก | ผ่าน |
| UAT7 | ทดสอบการตรวจสอบประเภทของข้อมูล | เลือกใช้ประเภทของข้อมูลต่างๆ | ประเภทของข้อมูลใช้ตรวจสอบได้ถูกต้องตามที่กำหนดนิพจน์ปรกติไว้ | ผ่าน |
| UAT8 | ทดสอบการนำเข้าประเภทของข้อมูล | กดปุ่ม “Import Input Type” | มีประเภทของข้อมูลที่นำเข้าสู่ระบบ | ผ่าน |
| UAT9 | ทดสอบนำออกประเภทของข้อมูล | กดปุ่ม “Export Input Type” | มีกล่องให้บันทึกประเภทของข้อมูล | ผ่าน |
| UAT10 | ทดสอบการเพิ่มคำหลักและตำแหน่งของคำหลัก และรูปแบบคำหลักในข้อกำหนดเอกสารนำเข้า | กดข้อกำหนดเอกสารนำเข้าแล้วกรอกคำหลักและตำแหน่งของคำหลัก และรูปแบบคำหลัก | บันทึกทันทีที่ทำการกรอกข้อมูลคำหลักและตำแหน่งของคำหลัก และรูปแบบคำหลัก | ผ่าน |
| UAT11 | ทดสอบการดูหัวข้อขององค์ประกอบใน | เมื่อมีการกรอกคำหลักและตำแหน่งของคำหลัก และ | จะแสดงให้เห็นหัวข้อขององค์ประกอบต่างๆ แต่ละประเภทในเอกสาร และ | ผ่าน |

| | | | | |
|---|---|---|---|---|
| | ข้อกำหนดเอกสารนำเข้า | รูปแบบคำหลักและรูปแบบคำหลักเรียบร้อย (การกำหนดป้ายบอกข้อมูล) แล้วกดปุ่ม “Show” | ความสัมพันธ์ของแต่ละองประกอบตามจากการนำป้ายบอกข้อมูลที่กำหนดไปใช้และค้นหาองค์ประกอบต่างๆ จากในเอกสาร | |
| UAT12 | ทดสอบการนำเข้าข้อมูลตามข้อกำหนดเอกสารการนำเข้า | ระบุข้อกำหนดของป้ายบอกข้อมูลกดปุ่ม “Accept” | ระบบนำเข้าข้อมูลจากเอกสารความต้องการนำเข้าโดยอัตโนมัติด้วยข้อกำหนดเอกสารนำเข้าที่ตั้งเอาไว้ | ผ่าน |
| UAT13 | ทดสอบการดูข้อมูลในเอกสารเอ็กซ์เอ็มแอลของระบบ | กดดูข้อมูลเอกสารเอ็กซ์เอ็มแอลของระบบจากระบบ | แสดงเอกสารเอ็กซ์เอ็มแอลของระบบให้ดู | ผ่าน |
| UAT14 | ทดสอบการดูรายงานการเปลี่ยนแปลงความต้องการจากเอกสารหรือต้นแบบรุ่นก่อนหน้า | เลือกเมนู “Change Report” | แสดงกล่องรายงานการเปลี่ยนแปลงข้อมูลความต้องการจากรุ่นก่อนหน้า | ผ่าน |
| UAT15 | ทดสอบการสร้างความสัมพันธ์ของต้นแบบแต่ละหน้า | เมื่อนำเข้าข้อมูลเรียบร้อยแล้ว “Relation” | มีรายงานความสัมพันธ์ของต้นแบบ | ผ่าน |
| UAT16 | ทดสอบการแก้ไขปรับปรุงความสัมพันธ์ของต้นแบบแต่ละหน้า | เลือกเมนู“Relation” แล้วทำการแก้ไขปรับปรุงตามต้องการ | ระบบแสดงความสัมพันธ์ของต้นแบบใหม่ ตามที่ได้แก้ไขปรับปรุง | ผ่าน |

| | | | | |
|---|---|---|---|---|
| UAT17 | ทดสอบการสร้างต้นแบบใหม่ | เลือกเมนู“Generate Prototype” | ระบบทำการสร้างต้นแบบใหม่ตามข้อมูลปัจจุบัน | ผ่าน |
| UAT18 | ทดสอบการปรับปรุงต้นแบบ | เลือกเมนู“Update Prototype” | ระบบทำการปรับปรุงต้นแบบตามข้อมูลปัจจุบัน | ผ่าน |
| UAT19 | ทดสอบการปรับปรุงเอกสารความต้องการ | เลือก “Update RS” | ระบบทำการปรับปรุงข้อมูลในเอกสารข้อกำหนดความต้องการตามข้อมูลปัจจุบัน | ผ่าน |

**5.2 การทดสอบความถูกต้องของกระบวนการทำงานของระบบ**

การทดสอบความถูกต้องของกระบวนการทำงานของระบบเป็นการทดสอบกระบวนการทำงานตามขอบเขตการสร้างต้นแบบลักษณะทดสอบได้โดยอัตโนมัติจากข้อกำหนดความต้องการที่ถูกเก็บเกี่ยว ดังตารางที่ 12

ตารางที่ 12 ผลการทดสอบความถูกต้องของกระบวนการทำงานของระบบ

| **หมายเลขกรณีทดสอบ** | **การทดสอบ** | **คำอธิบาย** | **ผลการทดสอบที่คาดหวัง** | **ผลการทดสอบ** |
|---|---|---|---|---|
| UAT1 | ตรวจสอบความครบถ้วนของข้อมูลที่อยู่ในเอกสารกับที่ระบบนำเข้า | เปรียบเทียบความครบถ้วนของข้อมูลด้วยคน | ข้อมูลของส่วนประกอบต่างๆ แต่ละชนิดเข้าระบบมาได้อย่างครบถ้วน | ทำงานถูกต้อง |
| UAT2 | ตรวจสอบความถูกต้องของข้อมูลที่อยู่ในเอกสารกับที่ระบบนำเข้า | เปรียบเทียบความถูกต้องของข้อมูลด้วยคน | ข้อมูลของส่วนประกอบต่างๆ แต่ละชนิดเข้าระบบมาได้อย่างถูกต้อง ทั้งชนิดรายละเอียด และความสัมพันธ์ของแต่ละส่วนประกอบ | ทำงานถูกต้อง |
| UAT3 | ตรวจสอบรายงานการเปลี่ยนแปลงความต้องการ | เปรียบเทียบข้อมูลที่เปลี่ยนแปลงไปจากเอกสารรุ่นก่อนหน้าด้วยคน | รายงานการเปลี่ยนแปลงความต้องการจากเอกสารรุ่นก่อนหน้าได้อย่างถูกต้อง | ทำงานถูกต้อง |

| | | | | |
|---|---|---|---|---|
| | จากรุ่นก่อนหน้า | | | |
| UAT4 | การตรวจสอบต้นแบบที่สร้างถูกขึ้นหรือปรับปรุงขึ้น | ตรวจสอบต้นแบบที่สร้างถูกขึ้นหรือปรับปรุงขึ้นกับที่คาดหวังจะได้จากเอกสารและการเปลี่ยนแปลง | ต้นแบบที่ได้ตรงตามความคาดหวังจากข้อมูลที่ให้ในเอกสาร และข้อมูลที่ใส่เพิ่มลงไปเพื่อกำหนดเพิ่มเติม | ทำงานถูกต้อง |
| UAT5 | การตรวจสอบความสัมพันธ์ของต้นแบบที่สร้างถูกขึ้นหรือปรับปรุงขึ้น | ตรวจสอบความสัมพันธ์ของต้นแบบที่สร้างถูกขึ้นหรือปรับปรุงขึ้นขึ้นกับที่คาดหวังจะได้จากเอกสารและการเปลี่ยนแปลง | ความสัมพันธ์ของต้นแบบที่ได้ตรงตามความคาดหวังจากข้อมูลที่ให้ในเอกสารและข้อมูลที่ใส่เพิ่มลงไปเพื่อกำหนดเพิ่มเติม | ทำงานถูกต้อง |
| UAT6 | การตรวจสอบการปรับปรุงเอกสารความต้องการ | ตรวจสอบเอกสารความต้องการหลังถูกปรับปรุงกับที่คาดหวังจะได้จากหลังการเปลี่ยนแปลงข้อมูลต่างๆ | เอกสารความต้องการหลังถูกปรับปรุงตรงตามความคาดหวังจากข้อมูลที่เปลี่ยนแปลงไป | ทำงานถูกต้อง |
| UAT7 | ตรวจสอบการใช้นิพจน์ปรกติ | ตรวจสอบว่าผลการทดสอบด้วยนิพจน์ปรกติเป็นไปตามกฎ | ผลการทดสอบด้วยนิพจน์ปรกติเป็นไปตามกฎ | ทำงานถูกต้อง |
| UAT8 | ตรวจสอบความครบถ้วนของการดำเนินการทดสอบต้นแบบอัตโนมัติ | ตรวจสอบว่าระบบดำเนินการทดสอบต้นอัตโนมัติได้ครบถ้วนตามที่คาดหวังจากข้อมูลความสัมพันธ์ปัจจุบัน | ระบบดำเนินการทดสอบต้นอัตโนมัติได้ครบถ้วน | ทำงานถูกต้อง |
| UAT9 | ตรวจสอบความถูกต้องของการดำเนินการทดสอบต้นแบบอัตโนมัติ | ตรวจสอบว่าระบบดำเนินการทดสอบต้นอัตโนมัติได้ความถูกต้องตามที่คาดหวังจากข้อมูลปัจจุบัน | ระบบดำเนินการทดสอบต้นอัตโนมัติได้ความถูกต้อง | ทำงานถูกต้อง |

### 5.3 การประเมินผลระบบสร้างต้นแบบแบบอัตโนมัติ

การประเมินผลงานวิจัยนี้ ได้ทำการทดลองสร้างต้นแบบจากความต้องการด้วยมือเปรียบเทียบกับการนำเข้าความต้องการมาสร้างต้นแบบด้วยระบบที่พัฒนาขึ้น โดยทดสอบนำเข้าข้อมูลจำนวน 25, 50, 100, 150, 300 ความต้องการตามลำดับ และทำการจับเวลาที่ใช้ในการทำงานหน่วยเป็นนาทีโดยวัดตามสำหรับการทดลองตามรายการต่อไปนี้

1. การสร้างต้นแบบด้วยมือจากเอกสารครั้งแรก
2. ระบบสร้างต้นแบบอัตโนมัติจากเอกสารครั้งแรก
3. การปรับปรุงต้นแบบด้วยมือเมื่อเอกสารเกิดการเปลี่ยนแปลง
4. ระบบปรับปรุงต้นแบบอัตโนมัติเมื่อเอกสารเกิดการเปลี่ยนแปลง
5. การปรับปรุงต้นแบบด้วยมือเมื่อเอกสารเกิดการเปลี่ยนแปลงครั้งที่2
6. ระบบปรับปรุงต้นแบบอัตโนมัติเมื่อเอกสารเกิดการเปลี่ยนแปลงครั้งที่2

โดยวัดจากเหตุการณ์ที่เกิดขึ้นดังตารางที่ 13 เป็นเวลาตั้งต้นครั้งแรก แต่ถ้าเมื่อเอกสารเกิดการเปลี่ยนแปลงจะวัดรวมเวลาถึงเหตุการณ์ที่เกิดขึ้นใน0 ใช้วิธีจับเวลาเพื่อนำมาวิเคราะห์เปรียบเทียบได้มาซึ่งผลการทดลองในตารางที่ 15 แล้วสามารถนำมาสร้างกราฟเพื่อช่วยการวิเคราะห์ดังภาพที่ 22

ตารางที่ 13 เหตุการณ์ที่ใช้ในการเปรียบเทียบวัดจับเวลาระหว่างการสร้างต้นแบบด้วยมือเพียงอย่างเดียวกับด้วยระบบ จากเอกสารครั้งแรก

| ขั้นตอน | วัดเวลาจากการทำด้วยมือ | วัดเวลาจากการทำระบบอัตโนมัติ |
|---|---|---|
| สร้างเอกสารความต้องการ | วัด (คน) | วัด (คน) |
| ตรวจสอบเอกสาร | วัด (คน) | วัด (คน) |
| วิเคราะห์ข้อมูลจากเอกสาร | วัด (คน) | วัด (ระบบ) |
| นำเข้าเอกสาร | ไม่มีขั้นตอน | วัด (ระบบ) |
| ตั้งข้อกำหนดเอกสารนำเข้า | ไม่มีขั้นตอน | วัด (คนตั้งหรือใช้ค่าที่มีในระบบ) |
| นำเข้าข้อมูลสู่ระบบ | วัด (คน) | วัด (ระบบ) |
| กำหนดความสัมพันธ์ | วัด (คน) | วัด (เพิ่มเติมจากเดิมได้ด้วยคน) |
| สร้างต้นแบบ | วัด (คน) | วัด (ระบบ) |
| ตรวจสอบต้นแบบ(องค์ประกอบ) | วัด (คน) | วัด (คน) |

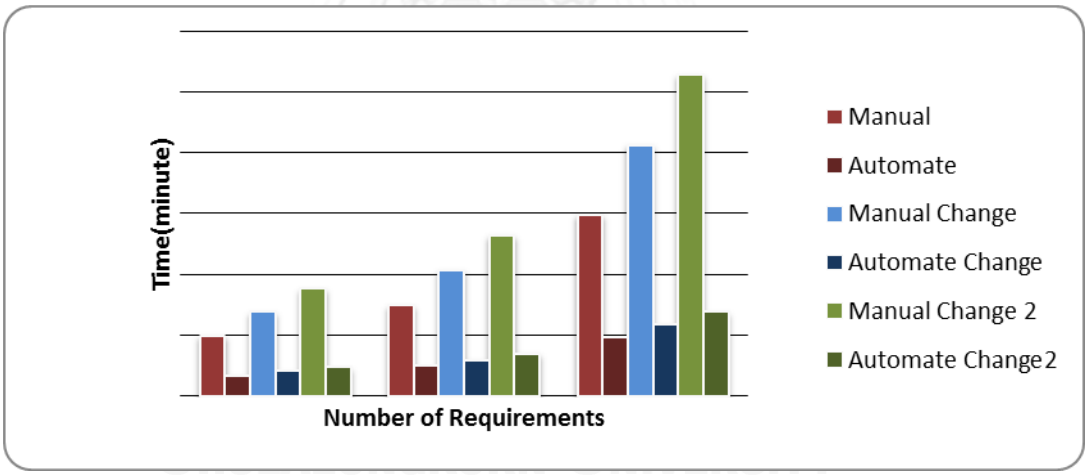
ตารางที่ 14 เหตุการณ์ที่ใช้ในการเปรียบเทียบวัดจับเวลาระหว่างการสร้างต้นแบบด้วยมือเพียงอย่างเดียวกับด้วยระบบ เมื่อเอกสารเกิดการเปลี่ยนแปลง

| ขั้นตอน | วัดเวลาจากการทำด้วยมือ | วัดเวลาจากการทำระบบอัตโนมัติ |
|---|---|---|
| ตรวจสอบเอกสาร | วัด (คน) | วัด (ระบบ) |
| วิเคราะห์ข้อมูลจากเอกสาร | วัด (คน) | วัด (ระบบ) |
| นำเข้าเอกสาร | ไม่มีขั้นตอน | วัด (ระบบ) |
| นำเข้าข้อมูลสู่ระบบ | วัด (คน) | วัด (ระบบ) |

| กำหนดความสัมพันธ์ | วัด (คน) | วัด (ระบบ) |
|---|---|---|
| ปรับปรุงต้นแบบ | วัด (คน) | วัด (ระบบ) |
| ตรวจสอบต้นแบบ(องค์ประกอบ) | วัด (คน) | วัด (คน) |

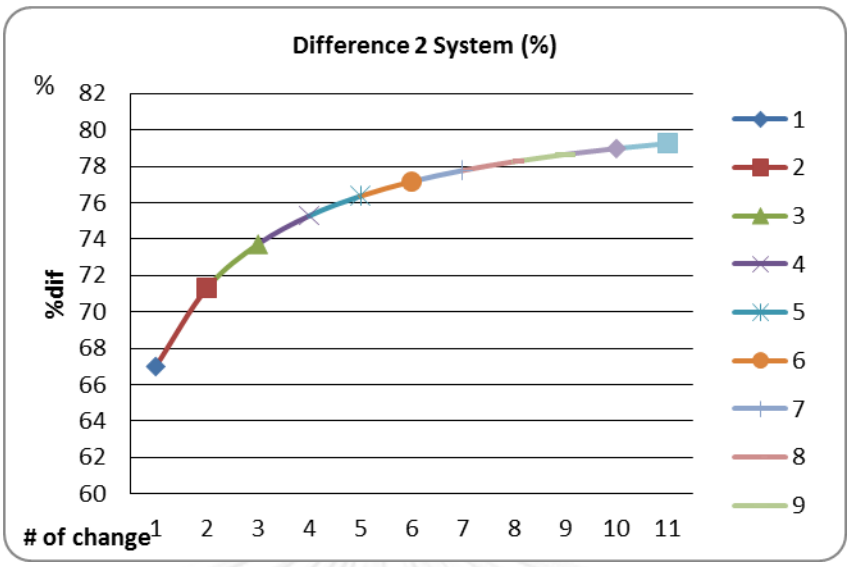
ตารางที่ 15 เปรียบเทียบเวลา(นาที)ที่ใช้การสร้างต้นแบบด้วยมือเพียงอย่างเดียวกับด้วยระบบในการเปลี่ยนแปลงแต่ละครั้ง

| Requirements | 25 Req | 50 Req | 100 Req | 150 Req | 300 Req |
|---|---|---|---|---|---|
| 1. Manual | 50 | 99 | 198 | 298 | 595 |
| 2. Automate | 18 | 34 | 66 | 98 | 194 |
| 3. Manual Change | 69 | 138 | 275 | 413 | 825 |
| 4. Automate Change | 22 | 41 | 80 | 118 | 234 |
| 5. Manual Change 2 | 88 | 176 | 352 | 527 | 1055 |
| 6. Automate Change2 | 25 | 48 | 93 | 139 | 275 |
| Diff Initial (%) | 63.43 | 65.57 | 66.65 | 67.00 | 67.36 |
| Diff Change (%) | 68.42 | 70.15 | 71.01 | 71.30 | 71.59 |
| Diff Change2 (%) | 71.24 | 72.73 | 73.48 | 73.72 | 73.97 |




ภาพที่ 22 กราฟความแตกต่างของเวลาที่ใช้ระหว่างการสร้างต้นแบบด้วยมือกับการสร้างต้นแบบอัตโนมัติ

เมื่อนำมาคำนวณผลที่ได้โดยเมื่อเปรียบเทียบข้อมูลจากการเปลี่ยนแปลงเอกสารจำนวน 10 ครั้ง รวมขั้นก่อนเปลี่ยนแปลงด้วยอีก 1 ครั้ง เทียบที่ความต้องการของลูกค้าจำนวน 100 ความต้องการ พบว่ายิ่งมีการเปลี่ยนแปลงเอกสารเก่าเข้าสู่ระบบมากครั้งเกิดเวลาที่แตกต่างกันของ 2 ระบบ โดยระบบอัตโนมัติใช้เวลาได้ดีกว่าระบบเดิมอย่างต่อเนื่อง ดังจะเห็นในภาพที่ 23




ภาพที่ 23 กราฟแสดงเปอร์เซนต์ความแตกต่างของเวลากับจำนวนครั้งของการสร้างและปรับปรุงต้นแบบระหว่างวิธีการทำด้วยมือกับวิธีการโดยอัตโนมัติ

สรุปได้ว่า การสร้างและปรับปรุงต้นแบบโดยอัตโนมัติใช้เวลาน้อยกว่าการทำด้วยมือประมาณ 65 ถึง 80 เปอร์เซ็นต์ และใช้เวลาลดลงอย่างต่อเนื่องถ้าเกิดการแก้ไขความต้องการ โดยทั้งหมดเทียบข้อมูลที่ความถูกต้องทั้งหมด คือ ผ่านการตรวจสอบแก้ไขสิ่งที่ผิดพลาดซึ่งเป็นหนึ่งในขั้นตอนที่ร่วมจับเวลา

### 5.4 การประเมินผลระบบทดสอบต้นแบบแบบอัตโนมัติ

การประเมินผลงานวิจัยนี้ ได้ทำการทดลองทดสอบต้นแบบจากความต้องการด้วยมือเปรียบเทียบกับการทดสอบอัตโนมัติด้วยระบบที่พัฒนาขึ้นกับต้นแบบลักษณะทดสอบได้โดยอัตโนมัติ โดยทดสอบระบบที่มีจำนวนสถานการณ์เปลี่ยนแปลง 25, 50, 100 สถานะตามลำดับ และทำการจับเวลาที่ใช้ในการทำงาน หน่วยเป็นนาทีโดยวัดตามสำหรับการทดลองตามรายการต่อไปนี้

1. การทดสอบต้นแบบด้วยมือจากเอกสารครั้งแรก
2. ระบบทดสอบต้นแบบอัตโนมัติจากเอกสารครั้งแรก
3. การทดสอบต้นแบบด้วยมือเมื่อเอกสารเกิดการเปลี่ยนแปลง
4. ระบบทดสอบต้นแบบอัตโนมัติเมื่อเอกสารเกิดการเปลี่ยนแปลง
5. การทดสอบต้นแบบด้วยมือเมื่อเอกสารเกิดการเปลี่ยนแปลงครั้งที่2
6. ระบบทดสอบต้นแบบอัตโนมัติเมื่อเอกสารเกิดการเปลี่ยนแปลงครั้งที่2

โดยวัดจากเหตุการณ์ที่เกิดขึ้นดังตารางที่ 16 เป็นเวลาตั้งต้นครั้งแรก แต่ถ้าเมื่อเอกสารเกิดการเปลี่ยนแปลงจะวัดรวมเวลาถึงเหตุการณ์ที่เกิดขึ้นในตารางที่ 17 ใช้วิธีจับเวลาเพื่อนำมาวิเคราะห์เปรียบเทียบได้มาซึ่งผลการทดลองในตารางที่ 18 แล้วสามารถนำมาสร้างกราฟเพื่อช่วยการวิเคราะห์ดังภาพที่ 24

ตารางที่ 16 เหตุการณ์ที่ใช้ในการเปรียบเทียบวัดจับเวลาระหว่างการทดสอบต้นแบบด้วยมือเพียงอย่างเดียวกับด้วยระบบ จากเอกสารครั้งแรก

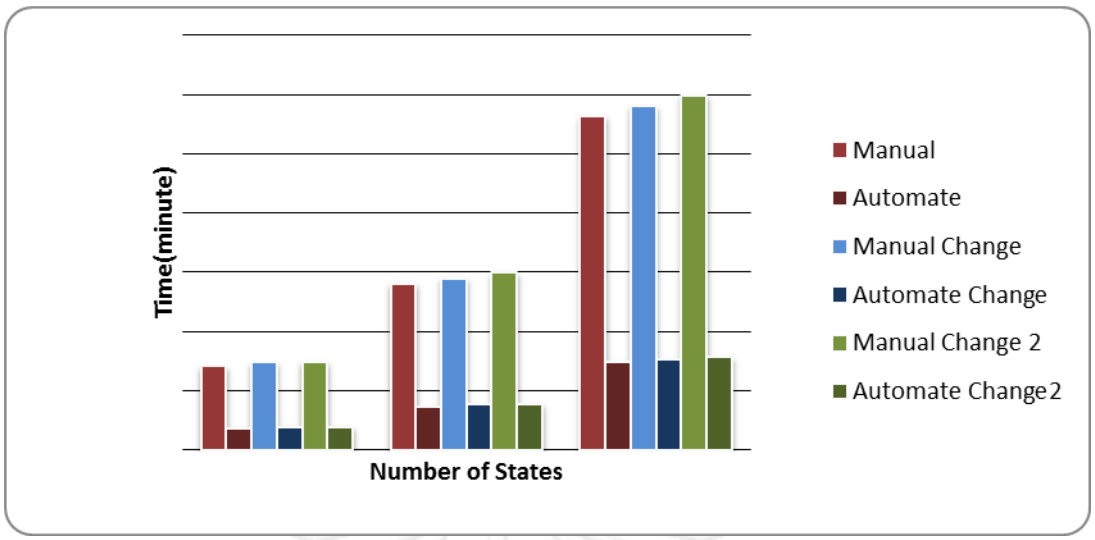
| ขั้นตอน | วัดเวลาจากการทำด้วยมือ | วัดเวลาจากการทำระบบอัตโนมัติ |
|---|---|---|
| สร้างคำสั่งทดสอบ | วัด (คน) | ไม่มีขั้นตอน (ระบบ - ได้มาในขั้นตอนการสร้างต้นแบบอัตโนมัติ) |
| ตรวจสอบและแก้ไขคำสั่งทดสอบหากมีข้อผิดพลาด | ไม่มีขั้นตอน (ถือเป็นส่วนหนึ่งของระยะเวลาการสร้างคำสั่งทดสอบ) | วัด (คน) |
| ทดสอบต้นแบบ | วัด (คน) | วัด (ระบบ - โดยผู้ตรวจสอบสามารถเข้าร่วมตรวจสอบระหว่างระบบทำการทดสอบเสมือนจริงได้) |

ตารางที่ 17 เหตุการณ์ที่ใช้ในการเปรียบเทียบวัดจับเวลาระหว่างการทดสอบต้นแบบด้วยมือเพียงอย่างเดียวกับด้วยระบบ เมื่อเอกสารเกิดการเปลี่ยนแปลง

| ขั้นตอน | วัดเวลาจากการทำด้วยมือ | วัดเวลาจากการทำระบบอัตโนมัติ |
|---|---|---|
| ปรับปรุงคำสั่งทดสอบ | วัด (คน) | วัด (ระบบ) |
| ตรวจสอบและแก้ไขคำสั่งทดสอบหากมีข้อผิดพลาด | ไม่มีขั้นตอน (ถือเป็นส่วนหนึ่งของระยะเวลาการสร้างคำสั่งทดสอบ) | วัด (คน) |
| ทดสอบต้นแบบ | วัด (คน) | วัด (ระบบ - โดยผู้ตรวจสอบสามารถเข้าร่วมตรวจสอบระหว่างระบบทำการทดสอบเสมือนจริงได้) |

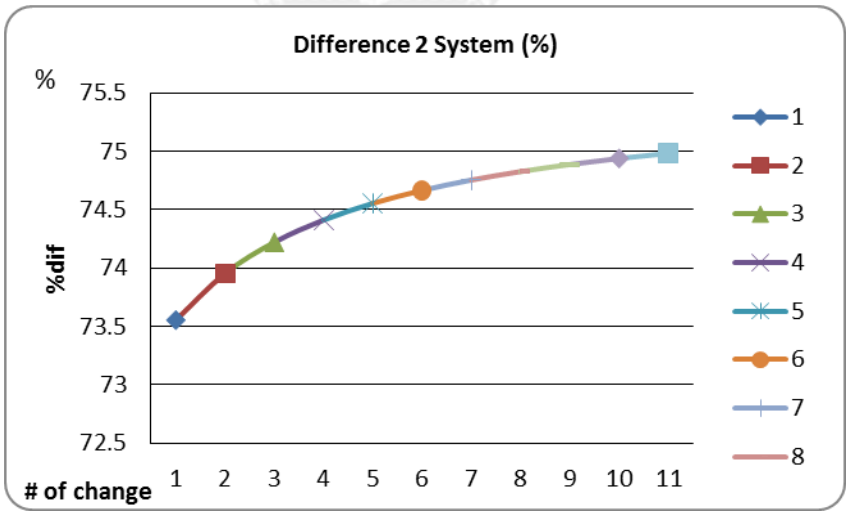
ตารางที่ 18 เปรียบเทียบเวลา(นาที)ที่ใช้การทดสอบต้นแบบด้วยมือเพียงอย่างเดียวกับด้วยระบบในการเปลี่ยนแปลงแต่ละครั้ง

| States | 25 States | 50 States | 100 States |
|---|---|---|---|
| 1. Manual | 141 | 281 | 563 |
| 2. Automate | 37 | 73 | 149 |
| 3. Manual Change | 148 | 290 | 580 |
| 4. Automate Change | 38 | 77 | 152 |
| 5. Manual Change 2 | 149 | 299 | 598 |
| 6. Automate Change2 | 39 | 78 | 157 |
| Diff Initial (%) | 73.56 | 74.04 | 73.56 |
| Diff Change (%) | 74.15 | 73.62 | 73.80 |
| Diff Change2 (%) | 73.67 | 73.90 | 73.67 |




ภาพที่ 24 กราฟความแตกต่างของเวลาที่ใช้ระหว่างการทดสอบต้นแบบด้วยมือกับการทดสอบต้นแบบอัตโนมัติ

เมื่อนำมาคำนวณผลที่ได้โดยเมื่อเปรียบเทียบข้อมูลจากการเปลี่ยนแปลงเอกสารจำนวน 10 ครั้ง รวมขั้นก่อนเปลี่ยนแปลงด้วยอีก 1 ครั้ง เทียบที่ความต้องการของลูกค้าจำนวน 25 สถานะ พบว่ายิ่งมีการเปลี่ยนแปลงเอกสารเก่าเข้าสู่ระบบมากครั้งเกิดเวลาที่แตกต่างกันของ 2 ระบบ โดยระบบอัตโนมัติใช้เวลาได้ดีกว่าระบบเดิมอย่างต่อเนื่อง ดังจะเห็นในภาพที่ 25




ภาพที่ 25 กราฟแสดงเปอร์เซนต์ความแตกต่างของเวลากับจำนวนครั้งของการทดสอบต้นแบบระหว่างวิธีการทำด้วยมือกับวิธีการโดยอัตโนมัติ

สรุปได้ว่า การทดสอบต้นแบบโดยอัตโนมัติใช้เวลาน้อยกว่าการทำด้วยมือประมาณ 70 ถึง 75 เปอร์เซ็นต์ และใช้เวลาลดลงอย่างต่อเนื่องถ้าเกิดการแก้ไขความต้องการ โดยทั้งหมดเทียบข้อมูล

ที่ความถูกต้องทั้งหมด คือ ผ่านการตรวจสอบแก้ไขคำสั่งทดสอบที่ผิดพลาด และมีการตรวจสอบว่าการทดสอบนั้นๆ ถูกต้อง จากผู้ตรวจสอบ ซึ่งเป็นหนึ่งในขั้นตอนที่ร่วมจับเวลา

**5.5 สรุปผลการทดลอง**

จากผลการทดลองที่กล่าวมาสรุปได้ว่า ระบบสามารถนำเข้าข้อมูลอัตโนมัติสำหรับสร้างต้นแบบส่วนติดต่อผู้ใช้อย่างถูกต้อง สามารถสร้างต้นแบบส่วนติดต่อผู้ใช้อัตโนมัติได้อย่างถูกต้องตามความต้องการด้านฟังก์ชันการทำงานและสามารถทดสอบต้นแบบนั้นได้อย่างอัตโนมัติตามขั้นตอนการใช้งานต้นแบบ โดยระบบช่วยลดการใช้ทรัพยากรด้านต่างๆ คือ เวลาและแรงงานที่ใช้ในการรับรู้ข้อมูลความต้องการ การเรียนรู้ประมวลผลข้อมูล การสร้างต้นแบบและการทดสอบต้นแบบได้อย่างดี โดยเฉพาะในกรณีที่มีการเปลี่ยนแปลงความต้องการดังกล่าว

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# บทที่ 6
# สรุปผลการวิจัย และข้อเสนอแนะ

## 6.1 สรุปผลการวิจัย

งานวิจัยนี้ได้นำเสนอวิธีการและได้พัฒนาระบบสำหรับการสร้างต้นแบบลักษณะทดสอบได้โดยอัตโนมัติจากข้อกำหนดความต้องการที่ถูกเก็บเกี่ยว คือ การรับข้อมูลจากเอกสารต้นทางที่เป็นเอกสารข้อความและแปลงให้อยู่ในรูปแบบเอ็กซ์เอ็มแอล ในกรณีนี้คือ เอกสารความต้องการทางด้านซอฟต์แวร์ อย่างไรก็ตาม กลไลดังกล่าวสามารถนำไปรับข้อมูลจากเอกสารประเภทอื่นๆ ได้ จากนั้นข้อมูลที่ได้จะถูกนำไปสกัดและนำไปใช้ในการสร้างต้นแบบลักษณะทดสอบได้ต่อไปโดยอัตโนมัติ รวมถึงสามารถสร้างต้นแบบแบบรูปภาพพร้อมใช้ให้ผู้ใช้ต้นแบบลักษณะทดสอบได้ดังกล่าวสามารถทดสอบตนเองได้โดยอัตโนมัติด้วยวิธีการที่นำเสนอไปข้างต้น โดยสามารถให้ผู้ทวนสอบเข้ามามีส่วนร่วมในการตรวจสอบขั้นตอนต่างๆ อย่างทันทีขณะที่ระบบกำลังทดสอบต้นแบบ ระบบสามารถแลกเปลี่ยนและปรับปรุงข้อมูลระหว่างเอกสารต่างๆ ระบบที่พัฒนาขึ้นได้สนับสนุนการนำเข้าองค์ประกอบของต้นแบบและความสัมพันธ์ของแต่ละองค์ประกอบโดยอัตโนมัติ และสามารถแสดงความสัมพันธ์ดังกล่าวออกมาให้ผู้ใช้ได้เห็นเป็นรูปธรรม ตามแนวคิดวิธีวิจัยที่ได้นำเสนอไว้ข้างต้น ประโยชน์ของงานวิจัยนี้ คือ สามารถได้ต้นแบบที่ต่อยอดได้อย่างทันทีหลังจากมีเอกสารข้อกำหนดความต้องการ และสามารถปรับปรุงต้นแบบนั้นๆ ได้ตามความต้องการที่เปลี่ยนแปลงไป ช่วยลดการใช้ทรัพยากรทางด้านแรงงานและเวลา ทั้งในการสร้างต้นแบบ การปรับปรุงต้นแบบและการทดสอบต้นแบบ นอกจากนี้ ยังเป็นการตรวจสอบเอกสารความต้องการขั้นต้นไปในตัว ก่อให้เกิดความสะดวกรวดเร็วต่อผู้พัฒนาซอฟต์แวร์ โดยลดระยะเวลาการทำงานจากเดิมลงอย่างมาก และมากยิ่งขึ้นในกรณีปรับปรุงข้อมูลจากเดิม รวมถึงคงสภาพความสอดคล้องต้องกันของข้อมูลได้ดียิ่งกว่าการทำด้วยตัวบุคคลซึ่งอาจจะเกิดความพลั้งพลาดได้สูง นอกจากนี้ ยังสามารถนำรูปแบบการเรียนรู้ การแลกเปลี่ยนและการปรับปรุงข้อมูลระหว่างเอกสารด้วยกันและต้นแบบซอฟต์แวร์ไปใช้ในด้านอื่นๆระบบจึงสามารถนำไปใช้ได้ในงานโดยทั่วไป และเหมาะอย่างยิ่งสำหรับกรณีที่มีการเปลี่ยนแปลงข้อกำหนดความต้องการหรือเอกสารประกอบบ่อยๆ และกรณีที่มีเอกสารหรือข้อมูลจำนวนมาก ซึ่งเป็นประโยชน์ต่อการปรับปรุงกระบวนการทางด้านซอฟต์แวร์ขององค์กรให้เป็นไปในแนวทางของการทำงานอย่างชาญฉลาด

## 6.2 ข้อจำกัด

จากการดำเนินงานวิจัย ปัญหาและข้อจำกัดที่พบคือ หากมีข้อผิดพลาดในการกำหนดค่าเกี่ยวกับเอกสารนำเข้าผิดพลาดเกินกว่าการทนต่อความผิดพร่องจะทำให้ระบบไม่สามารถได้ข้อมูลที่ต้องการได้ อย่างไรก็ตาม หากไม่มีการกำหนดค่าที่ต้องการ ระบบจะพยายามทำงานด้วยค่าหรือกระบวนการตั้งต้นที่กำหนดไว้ก่อน ตัวอย่างเช่น การใช้คำตั้งต้นเป็นป้ายบอกข้อมูล การใช้การจัดวางอย่างง่ายหากไม่มีการกำหนดตำแหน่งขององค์ประกอบ และการกำหนดขนาดมาตรฐานหากไม่มีการระบุขนาดขององค์ประกอบ เป็นต้น

### 6.3 แนวทางสำหรับการวิจัยต่อในอนาคต

สำหรับการวิจัยต่อในอนาคตจำเป็นต้องให้ความสนใจในการปรับปรุงการจัดวางองค์ประกอบในกรณีที่ผู้ใช้ไม่ได้กำหนดค่าไว้ให้สวยงามยิ่งขึ้น หรืออาจจะนำการประมวลผลภาพเข้ามาช่วยในการสกัดข้อมูลจากภาพร่างต้นแบบที่รับจากเอกสาร เพื่อให้ระบบตัดสินใจการจัดวางได้ โดยไม่ต้องอาศัยข้อมูลตำแหน่งโดยตรง

นอกจากนี้ ยังสามารถนำแนวคิดจากงานวิจัยนี้ไปใช้ในการสร้างองค์ประกอบอื่นๆ นอกจากต้นแบบซอฟต์แวร์ หรือนำส่วนหนึ่งของแนวคิดไปใช้ในการจัดการปรับปรุงข้อมูลในกลุ่มเอกสารหนึ่งๆ ที่ต้องมีความสอดคล้องกันให้เกิดการแก้ไขปรับปรุงข้อมูลต่างๆ อย่างถูกต้องได้โดยอัตโนมัติ



จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

**รายการอ้างอิง**


1. Repenning, N. P., and Terman, J. D. Nobody Ever Gets Credit for Fixing Problems that Never Happened: Creating and Sustaining Process Improvement. California Management Review(2001) : 64-88.
2. Srivastava, B., and Chang, Y. Business Insight from Collection of Unstructured Formatted Documents with IBM Content Harvester. ACM International Conference on Management of Data (COMAD), Mysore, India(2009).
3. Jackson, M. Software Requirements & Specifications. MA : Addison-Wesley, 1995.
4. Nielsen, J. Usability Engineering. CA : Morgan Kaufmann Publishers Inc, 1993.
5. Naumann, J. D., and Jenkins, M. A. Prototyping: The New Paradigm for Systems Development. MIS Quarterly6, 3 (Sep. 1982), 29-44.
6. Shehady, R. K., and Siewiorek, D. P. A Method to Automate User Interface Testing Using Variable Finite State Machines. Proceedings of the the 27th International Symposium on Fault-Tolerant Computing (FTCS), 80-88. Washington, DC : IEEE Computer Society,1997.
7. Elkoutbi, M., Khriss, I., and Keller, R. K. Automated Prototyping of User Interfaces Based on UML Scenarios. Automated Software Eng 13, 1 (Jan. 2006), 5-40.
8. สุรเชษฐ์ สูรย์ส่องธานี. การขยายความสามารถตามรอยความต้องการด้วยพฤติกรรมของความสัมพันธ์ระหว่างความต้องการ. วิทยานิพนธ์ปริญญามหาบัณฑิตสาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ภาควิศวกรรมคอมพิวเตอร์คณะวิศวกรรมศาสตร์จุฬาลงกรณ์มหาวิทยาลัย, 2553.
9. ยุรนันท์ โพธิ์ปั้น. ระบบนำเข้าข้อมูลอัตโนมัติสำหรับสร้างความสามารถตามรอยความต้องการ. วิทยานิพนธ์ปริญญามหาบัณฑิตสาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ภาควิศวกรรมคอมพิวเตอร์คณะวิศวกรรมศาสตร์จุฬาลงกรณ์มหาวิทยาลัย, 2554.
10. ชวลิต เอี่ยมสะอาด. แนวทางการสร้างกรณีทดสอบสำหรับการทดสอบการยอมรับของผู้ใช้. วิทยานิพนธ์ปริญญามหาบัณฑิตสาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ภาควิศวกรรมคอมพิวเตอร์คณะวิศวกรรมศาสตร์จุฬาลงกรณ์มหาวิทยาลัย, 2554.
11. Yourdon, E., and Constantine, L. L. Structured Design: Fundamentals of a Discipline of Computer Program and System Design. NJ : Prentice-Hall, 1979 (Facsimile edition 1986).

# ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

# ภาคผนวก ก
# ตัวอย่างแสดงการทำงานกรณีทดสอบ

ระบบที่พัฒนาขึ้นจากวิธีการที่นำเสนอออกแบบให้มีเมนูต่างๆ ดังต่อไปนี้

เมนูหลัก

1. File – เกี่ยวกับไฟล์

   ประกอบด้วย

   1.1) Load Document – นำเข้าเอกสารความต้องการ
2. Option – เกี่ยวกับการกำหนดค่าตัวเลือกต่างๆ

   2.1) Import Spec – กำหนดรายละเอียดหลักๆ ของการนำเข้าเอกสารความต้องการ เช่น ข้อความและลำดับชั้นของป้ายบอกตำแหน่งต่างๆ รวมถึงตัวเลือกหลักเกี่ยวกับการส่งออกต้นแบบ คือ ชื่อแพคเกจของต้นแบบที่ต้องการ

   2.2) Input Type – กำหนดประเภทการตรวจสอบข้อมูล โดยการตั้งชื่อประเภทและตั้งกฎด้วยนิพจน์ปรกติ
3. Misc – ฟังก์ชันความสามารถอื่นๆ ของระบบ

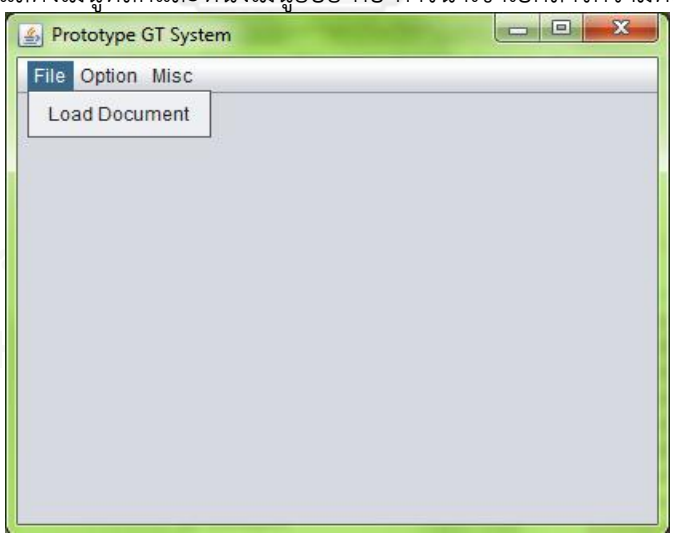
   3.1) Change Report – แสดงข้อมูลบันทึก (Log) ต่างๆ

   3.2) Relation – แสดงความสัมพันธ์ของการเปลี่ยนแปลงหน้าจอของต้นแบบ

   3.3) Update RS – ปรับปรุงเอกสารความต้องการตามที่ถูกแก้ไขภายหลัง

   3.4) Test Prototype – ทดสอบระบบปัจจุบันตามแพคเกตที่ตั้งค่าไว้

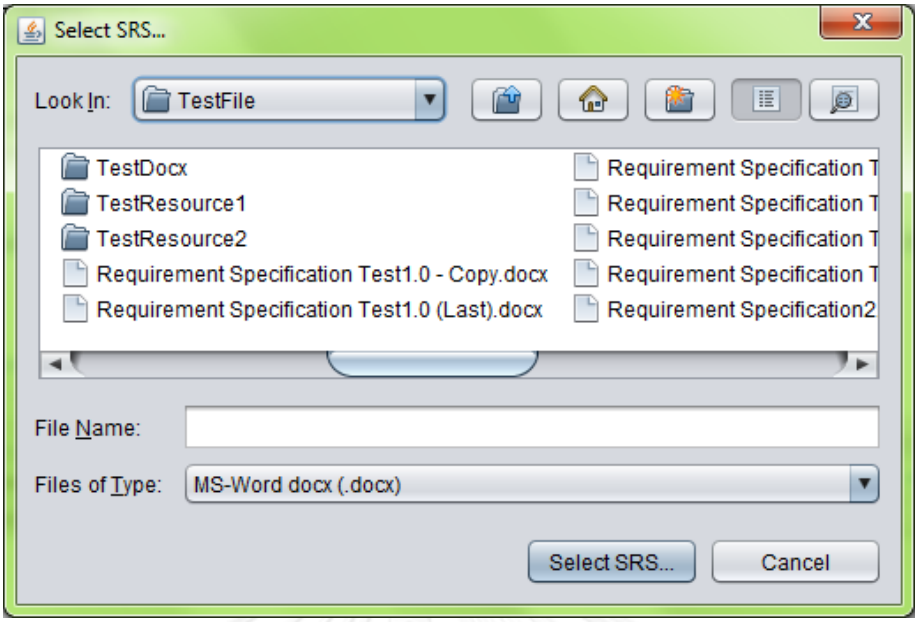
ภาพที่ 26 แสดงเมนูหลักและหนึ่งเมนูย่อย คือ การนำเข้าเอกสารความต้องการ




ภาพที่ 26 เมนูหลักและเมนูย่อยการนำเข้าเอกสารความต้องการ

รายละเอียดขั้นตอนการทำงานกรณีทดสอบสามารถทำตามขั้นตอนได้ดังนี้

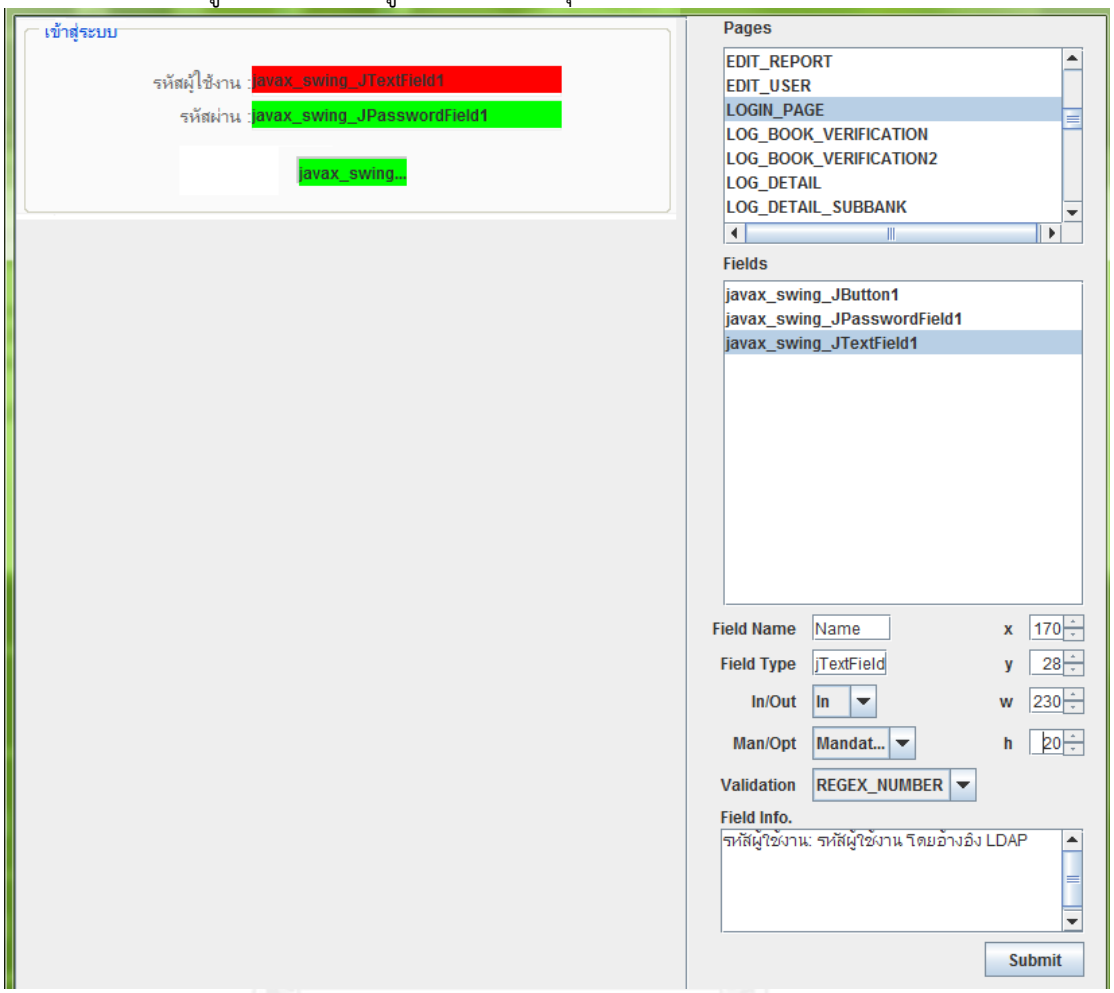
4. เริ่มการนำเข้าเอกสารความต้องการด้วยการเลือกเมนู File – Load Document
5. เลือกเอกสารความต้องการจากนั้นเลือกปุ่ม Select SRS... ตามภาพที่ 27




ภาพที่ 27 กล่องเลือกข้อกำหนดความต้องการที่จะนำเข้า

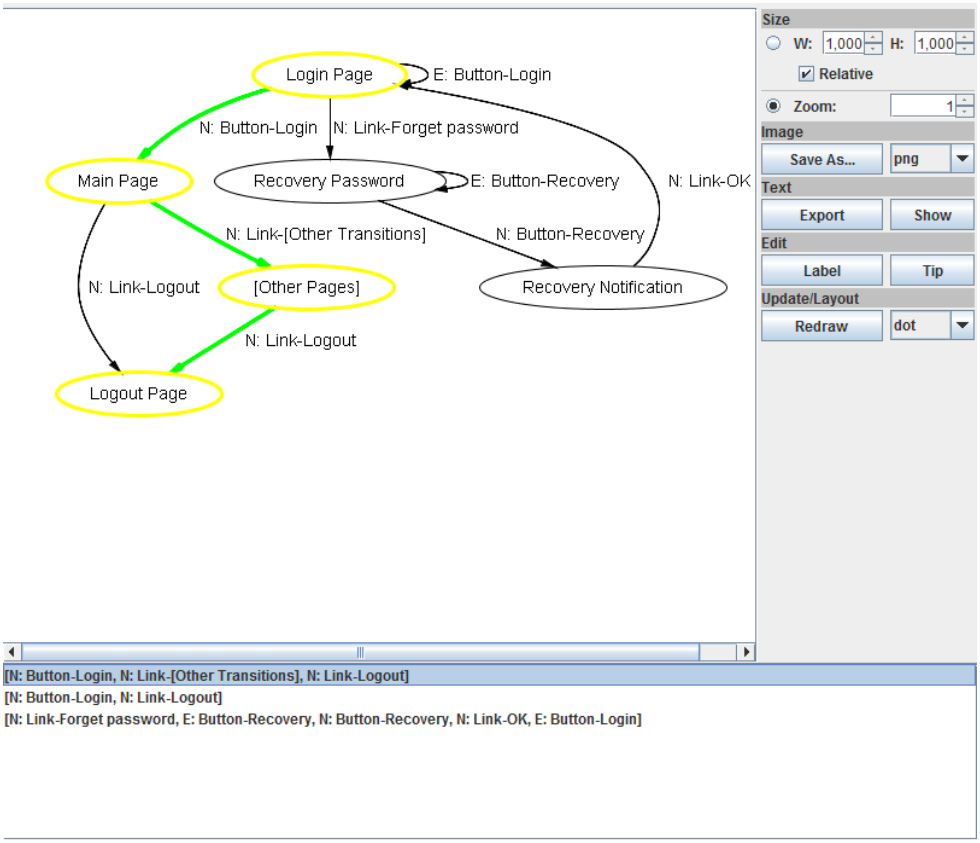
6. หากไม่มีการกำหนดข้อกำหนดการนำเข้าเอกสารความต้องการไว้ในไฟล์รวบรวมค่ากำหนดต่างๆ ระบบจะขึ้นหน้าต่างสอบถามรายละเอียดเกี่ยวกับป้ายบอกตำแหน่งและชื่อแพคเกจในการส่งออกต้นแบบตามภาพที่ 19
7. ระบบจะเริ่มประมวลผลโดยเก็บเกี่ยวข้อมูลจากข้อกำหนดความต้องการที่เลือกไว้ แล้วนำข้อมูลเหล่านั้นมาแสดงให้ผู้ใช้ได้แก้ไข เพิ่มเติมรายละเอียดด้านต่างๆ ดังภาพที่ 28
   7.1) ระบบแสดงรูปภาพอ้างอิงตามเอกสารดังที่เห็นบริเวณซ้ายของภาพ
   7.2) ระบบแสดงเขตข้อมูลทั้งหมดที่มีในหน้าจอที่กำลังเลือกด้วยบริเวณสีพร้อมชื่อกำกับ หากไม่ได้มีการตั้งชื่อไว้ในเอกสาร ระบบกำหนดชื่อมาตรฐานให้แก่ชื่อตัวแปรโดยอิงตามชื่อแพคเกจและคลาสขององค์ประกอบนั้นๆ โดยบริเวณสีของเขตข้อมูลทั่วไปจะเป็นสีเขียว ส่วนบริเวณสีของเขตข้อมูลที่กำลังเลือกจะเป็นสีแดง ผู้ใช้แก้ไขค่าต่างๆ ได้ดังนี้
      7.2.1) ตำแหน่งของเขตข้อมูล ผู้ใช้กำหนดจากกล่องทางด้านขวา (x,y) หรือคลิกเมาส์ซ้ายลงบนบริเวณหน้าจอตัวอย่าง
      7.2.2) ขนาดของเขตข้อมูล ผู้ใช้กำหนดจากกล่องทางด้านขวา (w,h) หรือคลิกเมาส์ขวาลงบนบริเวณหน้าจอตัวอย่างเพื่อกำหนดจุดล่างขวาของเขตข้อมูล
      7.2.3) กำหนดหรือแก้ไขชื่อเขตข้อมูล (Field Name), ชนิดเขตข้อมูล (Field Type), ประเภทการรับข้อมูลเข้า/ออก, ความสำคัญ (จำเป็น/เพิ่มเติม), ชื่อกฎการตรวจสอบเขตข้อมูลนี้ กรณีเป็นเขตข้อมูลเข้าที่รับจากผู้ใช้งานโปรแกรม
      7.2.4) Field Info. แสดงข้อมูลอธิบายเขตข้อมูลจากเอกสาร ซึ่งจะถูกนำไปเป็นส่วนหนึ่งของการคอมเมนต์ในโค้ดต้นแบบ เพื่อให้โปรแกรมเมอร์เข้าใจเขตข้อมูลนี้ได้โดยง่าย

7.3) ผู้ใช้กดยืนยันข้อมูลที่แก้ไขปรับปรุงแล้ว




ภาพที่ 28 ตัวอย่างการปรับปรุงค่าของต้นแบบในหน้าจอเข้าสู่ระบบ

8. ระบบจะแสดงความสัมพันธ์จากข้อมูลที่อยู่ในเอกสารในรูปแบบกล่องให้ผู้ใช้กำหนดความสัมพันธ์เพิ่มเติมได้ตามภาพที่ 20 และผู้ใช้สามารถมองความสัมพันธ์ในรูปแบบกราฟได้โดยการเลือกปุ่ม “View Graph” ซึ่งจะแสดงกราฟความสัมพันธ์ตามภาพที่ 29 ซึ่งมีรายละเอียดต่างๆ ดังนี้




ภาพที่ 29 กราฟตัวอย่างแสดงกราฟความสัมพันธ์

8.1) ภาพกราฟความสัมพันธ์ปัจจุบัน (ภาพด้านซ้าย)

8.2) เส้นทางทดสอบต่างๆ แบบค้นทางลึกก่อน (รายการด้านล่าง)

8.3) การกำหนดขนาดกราฟ (w,h) โดยเลือกกำหนดขนาดตามอัตราส่วนได้ด้วยตัวเลือก Relative

8.4) การกำหนดขนาดกราฟด้วยค่าการขยายด้วยการเลือกแบบ Zoom

8.5) ผู้ใช้สามารถเลือกชนิดไฟล์ส่งออกภาพกราฟแล้วบันทึกโดยการกดปุ่ม “Save As...”

8.6) ผู้ใช้สามารถบันทึกข้อมูลกราฟในรูปแบบข้อความที่ใช้ได้ร่วมกับโปรแกรมกราฟวิซ ด้วยปุ่ม Export หรือแสดงข้อมูลนั้นด้วยปุ่ม Show

8.7) ผู้ใช้สามารถแก้ไขป้ายกำกับและข้อมูลที่แสดงเมื่อชี้เพื่อนำเสนอไปยังกราฟได้

8.8) ผู้ใช้สามารถปรับเปลี่ยนอัลกอริทึมการจัดวางกราฟโดยเลือกอัลกอริทึมจากตัวเลือกตามภาพแสดงการใช้อัลกอรึทึมแบบ dot เหมาะกับการจัดวางกราฟทั่วไป ที่โหนดขนาดปานกลาง เราสามารถปรับอัลกอรึทึมตามความเหมาะสมแล้วสั่งวาดใหม่ด้วยการกดปุ่ม Redraw

9. ในรายการตัวเลือก Option จะมีเมนูย่อยต่างๆ โดยเมนู Import Spec เป็นการกำหนดค่าเบื้องต้นเกี่ยวกับการนำเข้า (และส่งออก) ดังที่ผ่านมาแล้วตามขั้นตอนข้างต้น

ส่วนเมนู Input Type เป็นการกำหนดประเภทข้อมูลเข้าเขตข้อมูลจากผู้ใช้ ซึ่งกฎเหล่านี้จะถูกนำไปแปลงเป็นโค้ตรวจสอบในต้นแบบที่ระบบสร้างขึ้นด้วยตามภาพที่ 21 โดยเมื่อเลือกเมนูดังกล่าวขะมีกล่องกำหนดประเภทข้อมูลดังภาพที่ 30

ภาพที่ 30 กล่องกำหนดประเภทข้อมูล

9.1) หากต้องการทำการเพิ่มกฎ ให้ผู้ใช้พิมพ์ชื่อกฎใหม่และกดปุ่ม Add New

9.2) หากต้องการแก้ไขกฎ ให้ผู้ใช้เลือกรายการชื่อกฎ จากตัวอย่างกำลังเลือกอยู่ที่ NUMBER2 แล้วแก้ไข RegEx Rule ให้เป็นนิพจน์ปรกติใหม่ที่ต้องการ แล้วกดปุ่ม Save Rule

9.3) หากต้องการลบกฎ ให้ผู้ใช้ทำเช่นเดียวกับขั้นตอนการแก้ไขโดยให้ช่องกฎว่างเปล่าแล้วบันทึก

10. เมนู Misc รวบรวมคุณสมบัติอื่นๆ ของระบบ อันได้แก่ การแสดงข้อมูลบันทึกเบื้องต้น ซึ่งระบบจะบันทึกไว้อัตโนมัติในแต่ละครั้ง การแสดงความสัมพันธ์ที่กล่าวไว้ข้างต้น การปรับปรุงเอกสารและการทดสอบต้นแบบ

10.1) การทดสอบต้นแบบจะเป็นไปอย่างอัตโนมัติ ผู้ใช้เลือกเมนู Test Prototype แล้วระบบจะขึ้นกล่องสอบถามเวลาดำเนินการในแต่ละขั้นตอนว่านานเท่าใด (Test Prototype) เช่น หากผู้ใช้กำหนดว่าขั้นตอนละ 2000 มิลลิวินาที (2 วินาที) ระบบจะจำลองการกรอกค่าเขตข้อมูลต่างๆ นานขั้นตอนละ 2 วินาที และผู้ใช้หรือผู้ตรวจสอบสามารถสังเกตตรวจสอบไปพร้อมๆ กับการทดสอบได้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

## ประวัติผู้เขียนวิทยานิพนธ์

นายนาวิน เฟื่องฟู เกิดเมื่อวันที่ 10 เมษายน พ.ศ. 2532 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (วศ.บ.) สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2553 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2554

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University