

การออกแบบ โครงสร้างแบบรูปความต้องการสำหรับระบบปลอดภัยเชิงวิกฤต



นายเอกชัย แซ่ตั้ง

สถาบันวิทยบริการ

วิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

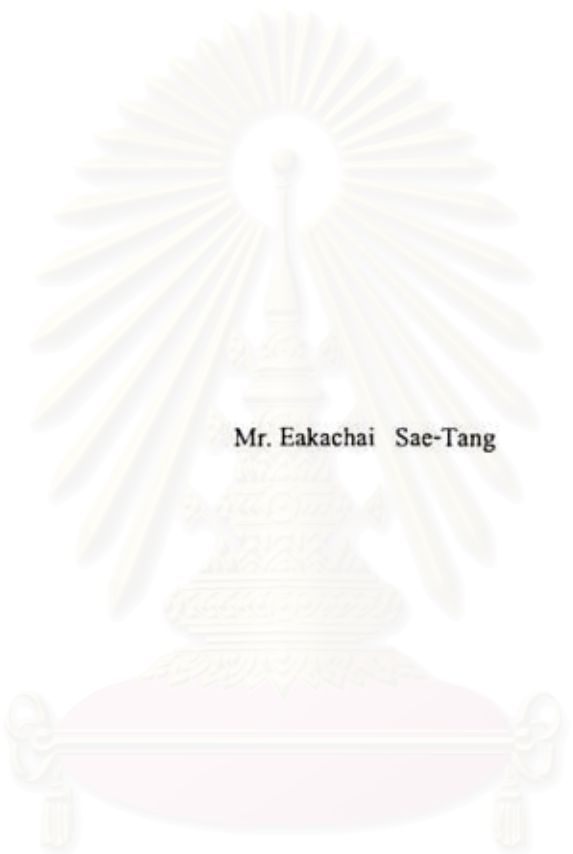
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2549

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

DESIGN OF REQUIREMENT PATTERN STRUCTURE
FOR SAFETY-CRITICAL SYSTEMS



Mr. Eakachai Sae-Tang

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

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

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2006

Copyright of Chulalongkorn University

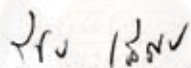
490626

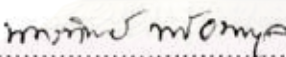
หัวข้อวิทยานิพนธ์	การออกแบบ โครงสร้างแบบรูปความต้องการสำหรับระบบปลอดภัยเชิงวิกฤต
โดย	นายเอกชัย แซ่ตั้ง
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษา	อาจารย์ นครทิพย์ พร้อมพูล

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็น
ส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาโทบัณฑิต

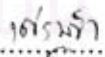

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร. ดิเรก ลาวัณยศิริ)

คณะกรรมการสอบวิทยานิพนธ์


..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ วิชาญ เลิศวิภาตระกูล)


..... อาจารย์ที่ปรึกษา
(อาจารย์ นครทิพย์ พร้อมพูล)


..... กรรมการ
(รองศาสตราจารย์ ดร. พรศิริ หมั่นไชยศรี)


..... กรรมการ
(อาจารย์ ดร. เศรษฐา ปานงาม)

เอกชัย แซ่ตั้ง : การออกแบบโครงสร้างแบบรูปความต้องการสำหรับระบบปลอดภัยเชิงวิกฤต
(DESIGN OF REQUIREMENT PATTERN STRUCTURE FOR SAFETY-CRITICAL
SYSTEMS) อ. ที่ปรึกษา : อ. นครทิพย์ พร้อมพูล, 217 หน้า

วิศวกรรมความต้องการซอฟต์แวร์ถือได้ว่าเป็นส่วนสำคัญในการพัฒนาซอฟต์แวร์ ข้อกำหนดความต้องการซอฟต์แวร์ที่ไม่สมบูรณ์จะส่งผลเสียต่อการพัฒนาซอฟต์แวร์อย่างมาก โดยเฉพาะวิศวกรรมความต้องการซอฟต์แวร์สำหรับระบบปลอดภัยเชิงวิกฤตซึ่งมีความสำคัญอย่างมากกับความถูกต้องในการทำงานของระบบ

แบบรูปเป็นการนำกลับมาใช้ใหม่ประเภทหนึ่งและนิยมนำมาใช้ในการพัฒนาซอฟต์แวร์ในหลายขั้นตอน ผู้วิจัยได้สังเกตเห็นคุณประโยชน์ของแบบรูปที่สร้างจากประสบการณ์ของผู้รู้เพื่อถ่ายทอดองค์ความรู้ และเห็นว่ามีเหมาะสมต่อการนำมาใช้ในงานวิศวกรรมความต้องการซอฟต์แวร์ โดยเฉพาะอย่างยิ่งการนำมาใช้ในงานวิศวกรรมความต้องการซอฟต์แวร์สำหรับระบบปลอดภัยเชิงวิกฤต ซึ่งจะทำให้ความต้องการที่ได้มามีความถูกต้องและสมบูรณ์ยิ่งขึ้น และยังช่วยลดต้นทุนและเวลาที่ต้องใช้ในงานวิศวกรรมความต้องการซอฟต์แวร์

ในวิทยานิพนธ์ชิ้นนี้ได้วิเคราะห์คุณลักษณะของวิศวกรรมความต้องการซอฟต์แวร์ โดยเฉพาะด้านที่เกี่ยวข้องกับความปลอดภัย เพื่อออกแบบโครงสร้างและองค์ประกอบของแบบรูปที่นำมาใช้กับวิศวกรรมความต้องการซอฟต์แวร์ โดยใช้ชื่อแบบรูปว่า “แบบรูปความต้องการ” ประเภทของแบบรูปที่ออกแบบมี 3 ประเภทคือ แบบรูปโดเมนความต้องการ แบบรูปกระบวนการความต้องการ และแบบรูปความต้องการส่วนความปลอดภัย วิทยานิพนธ์นี้ได้เสนอความหมายและรูปแบบขององค์ประกอบในแบบรูปความต้องการ แนวทางในการใช้งานร่วมกับกระบวนการพัฒนาซอฟต์แวร์ เครื่องมือสนับสนุนการใช้งานแบบรูปความต้องการ และได้ทำการทดลองเพื่อประเมินโครงสร้างแบบรูปความต้องการ

ผลลัพธ์ที่ได้จากการทดลองโดยหน่วยตัวอย่าง 18 หน่วยตัวอย่าง แสดงให้เห็นว่าแบบรูปความต้องการสามารถช่วยในงานวิศวกรรมความต้องการซอฟต์แวร์ได้อย่างมีนัยสำคัญที่ระดับความเชื่อมั่น 95 เปอร์เซ็นต์ และความเห็นจากผู้ประเมินแบบรูปความต้องการจำนวน 9 คน เห็นว่าโครงสร้าง ลักษณะการใช้งาน และความพึงพอใจในการใช้แบบรูปอยู่ในระดับที่ดี ที่คะแนนเฉลี่ย 3.95 จากช่วงการให้คะแนน 1 ถึง 5 จากน้อยมากถึงดีมาก

ภาควิชา.....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต..... เอกชัย แซ่ตั้ง
สาขาวิชา.....วิศวกรรมซอฟต์แวร์..... ลายมือชื่ออาจารย์ที่ปรึกษา..... ทศพร นพอมัย
ปีการศึกษา.....2549.....

##4670613921: MAJOR COMPUTER ENGINEERING

KEY WORDS: SOFTWARE REQUIREMENTS / SAFETY-CRITICAL SYSTEM / REQUIREMENT PATTERN

EAKACHAI SAE-TANG : DESIGN OF REQUIREMENT PATTERN STRUCTURE FOR SAFETY-CRITICAL SYSTEMS. THESIS ADVISOR: NAKORNTHIP PROMPON, 217 pp.

Software Requirements Engineering is one of the important in software development. An incomplete software requirement specification can cause a large damage in software development, especially in developing a safety-critical system concern with the correctness of system functional.

Pattern is a kind of reuse, which is popular in software development. The researcher sees the benefit of pattern created from expert experiences for knowledge transfer, and finds it is suitable to be used in software requirements engineering, especially for safety-critical systems. It would make the requirements gathered more accurate and complete, and reduce the amount of cost and time for software requirements engineering.

In this thesis, the characteristics of software requirements especially the safety-related requirements are analyzed to design the structure and element of pattern for requirements engineering. The pattern is called "Requirement Pattern" and is separated in 3 types, which are Requirement Domain Pattern, Requirement Process Pattern and Safety-Related Requirement Pattern. This thesis also presents the meaning and format of elements in requirement pattern, provides method to apply it to software development process, develops a supporting tool and performs the experiment to evaluate the pattern structure.

Results from an experiment using 18 sample units show that use of requirements pattern significantly help in the software requirements engineering with 95 percent of confidence level and 9 evaluator agree that structure, usability and satisfaction of propose pattern is in good level, which average score is 3.95 out of 5.

Department.....Computer Engineering..... Student's signature.....*เอกชัย แซ่แตง*.....
Field of study...Software Engineering..... Advisor's signature..... *Nakornthip Prompon*.....
Academic year.....2006.....

กิตติกรรมประกาศ

ขอกราบขอบพระคุณ อาจารย์ นครทิพย์ พร้อมพูล ในฐานะอาจารย์ที่ปรึกษาที่ได้เสียสละเวลาให้คำแนะนำ และแนวทางที่เป็นประโยชน์ ขอกราบขอบพระคุณอาจารย์ที่ดูแล ห่วงใย และสั่งสอนไม่เฉพาะในฐานะอาจารย์ที่ปรึกษา อาจารย์เป็นบุคคลสำคัญที่สุดที่ได้ช่วยให้วิทยานิพนธ์ชิ้นนี้สำเร็จลงได้

ขอกราบขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ วิชาญ เลิศวิภาตระกูล รองศาสตราจารย์ ดร.พรศิริ หมั่นไชยศรี และอาจารย์ ดร.เศรษฐา ปานงาม ที่ได้ให้คำแนะนำเกี่ยวกับงาน

ขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ กอบกุล เดชะวนิช รองศาสตราจารย์ มณฑนา ปราการสมุทร อาจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์ และอาจารย์ ดร.พินิจ คนองชัยยศ ที่สนับสนุนและให้กำลังใจ

ขอขอบพระคุณ บริษัท ระบบขนส่งมวลชนกรุงเทพ จำกัด มหาชน (BTS) และบริษัทรถไฟฟ้ากรุงเทพ จำกัด มหาชน (BMCL) ที่ให้ความอนุเคราะห์อนุญาตให้เข้าไปศึกษาระบบการทำงาน

ขอขอบคุณพี่ๆ เพื่อนๆ และน้องๆ ในห้องปฏิบัติการวิศวกรรมซอฟต์แวร์ ที่คอยให้กำลังใจ และแนะนำ โดยเฉพาะได้มาช่วยเป็นหน่วยตัวอย่างในการทดลองด้วย

ท้ายที่สุดนี้ขอกราบขอบพระคุณบิดา มารดา ผู้เป็นกำลังใจ แรงใจ ในการทำงาน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

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 ขอบเขตของการวิจัย.....	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.5 วิธีดำเนินการวิจัย.....	4
1.6 บทความวิชาการที่ได้รับการตีพิมพ์.....	9
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	10
2.1 แนวคิดและทฤษฎี.....	10
2.2 งานวิจัยที่เกี่ยวข้อง.....	23
บทที่ 3 การวิเคราะห์โครงสร้างแบบรูปความต้องการและโครงสร้างแบบรูปความต้องการสำหรับระบบปลอดภัยเชิงวิกฤต.....	28
3.1 ลักษณะของปัญหาในวิศวกรรมความต้องการซอฟต์แวร์.....	28
3.2 การนำกลับมาใช้ใหม่ในวิศวกรรมความต้องการซอฟต์แวร์.....	36
3.3 วิเคราะห์โครงสร้างแบบรูปความต้องการ.....	41
3.4 การวิเคราะห์องค์ประกอบและโครงสร้างความสัมพันธ์ในแบบรูปความต้องการ.....	44
บทที่ 4 สรุปโครงสร้างและองค์ประกอบของแบบรูปความต้องการ.....	56
4.1 โครงสร้างและความสัมพันธ์.....	56

บทที่	หน้า
4.2 องค์ประกอบพื้นฐานของแบบรูป.....	62
4.3 องค์ประกอบของแบบรูปโดเมนความต้องการ.....	65
4.4 องค์ประกอบของแบบรูปกระบวนการความต้องการ.....	78
4.5 องค์ประกอบของแบบรูปความต้องการส่วนความปลอดภัย.....	84
บทที่ 5 การสร้างแบบรูปความต้องการ.....	91
5.1 แบบรูปความต้องการที่สร้างขึ้น.....	91
5.2 แบบรูประบบงานจำหน่ายตั๋วโดยสารที่สถานีโดยสาร.....	92
5.3 แบบรูปการกำหนดสภาพแวดล้อม.....	100
5.4 แบบรูปการเว้นระยะห่างระหว่างรถไฟที่ปลอดภัย.....	103
5.5 แบบรูปการควบคุมการเปิดประตูรถไฟ.....	111
5.6 แบบรูปการกำหนดตำแหน่งและความเร็วของรถไฟ.....	119
บทที่ 6 การออกแบบกระบวนการในการใช้แบบรูปความต้องการ.....	126
6.1 วิสวกรรมความต้องการซอฟต์แวร์ในกระบวนการพัฒนาซอฟต์แวร์.....	126
6.2 การใช้แบบรูปความต้องการในวิสวกรรมความต้องการซอฟต์แวร์.....	127
6.3 การใช้แบบรูปความต้องการร่วมกับแบบรูปประเภทอื่นๆ.....	131
บทที่ 7 การพัฒนาเครื่องมือเพื่อช่วยในการสร้างและใช้งานแบบรูปความต้องการ.....	133
7.1 สภาพแวดล้อมและกระบวนการในการพัฒนาเครื่องมือ.....	133
7.2 ข้อกำหนดและความสามารถของเครื่องมือ.....	134
7.3 การวิเคราะห์และออกแบบเครื่องมือ.....	135
7.4 การทดสอบเครื่องมือ.....	144
7.5 ตัวอย่างของเครื่องมือที่พัฒนา.....	144
บทที่ 8 การทดสอบโครงสร้างแบบรูปความต้องการ.....	151
8.1 จุดประสงค์ในการทดสอบ.....	151
8.2 องค์ประกอบในการทดสอบ.....	151
8.3 ขั้นตอนวิธีในการทดสอบ.....	152

บทที่	หน้า
8.4 ผลที่ได้จากการทดลอง	153
8.5 วิเคราะห์และสรุปผลการทดสอบ	155
บทที่ 9 บทสรุป.....	158
9.1 อภิปรายผลการวิจัย.....	158
9.2 สรุปผลการวิจัย.....	159
9.3 ข้อเสนอแนะ.....	160
รายการอ้างอิง	161
ภาคผนวก	164
ภาคผนวก ก. บทความวิชาการ	165
ภาคผนวก ข. เอกสารแม่แบบและเอกสารตัวอย่างของแบบรูปความต้องการ	176
ภาคผนวก ค. คำถามประกอบการทดลอง.....	206
ประวัติผู้เขียนวิทยานิพนธ์	217

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 การประเมินค่าความเสี่ยงอุบัติเหตุ	21
ตารางที่ 2.2 หมวดหมู่การเสี่ยงอุบัติเหตุ	21
ตารางที่ 3.1 รายละเอียดและความสำคัญสำหรับองค์ความรู้ในมุมมองโดเมนความต้องการ	47
ตารางที่ 3.2 รายละเอียดและความสำคัญสำหรับองค์ความรู้ในมุมมองกระบวนการความต้องการ	51
ตารางที่ 3.3 รายละเอียดและความสำคัญสำหรับองค์ความรู้ในมุมมองกระบวนการความต้องการ	54
ตารางที่ 8.1 ผลการทดลองที่ได้จากการใช้และไม่ใช้แบบรูป	154
ตารางที่ 8.2 ผลการประเมินแบบรูปความต้องการ	155

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

ภาพประกอบ	หน้า
รูปที่ 1.1 วิธีและขั้นตอนในการวิจัย	4
รูปที่ 1.2 แนวคิดของโครงสร้างแบบรูปความต้องการในขั้นต้น.....	6
รูปที่ 1.3 หน้าที่การทำงานของเครื่องมือที่ช่วยในการสร้างและใช้งานแบบรูป	7
รูปที่ 2.1 ความสัมพันธ์ระหว่างองค์ประกอบของแบบรูป	12
รูปที่ 2.2 ความต้องการที่เกี่ยวข้องกับความปลอดภัย	18
รูปที่ 3.1 กระบวนการวิศวกรรมความต้องการซอฟต์แวร์และผู้มีส่วนเกี่ยวข้อง	30
รูปที่ 3.2 แบบจำลองแสดงปัญหาในวิศวกรรมวิศวกรรมความต้องการซอฟต์แวร์	32
รูปที่ 3.3 แบบจำลองที่แสดงกระบวนการวิศวกรรมความต้องการซอฟต์แวร์.....	34
รูปที่ 4.1 ความสัมพันธ์ของแบบรูปทั้ง 3 ประเภท.....	57
รูปที่ 4.2 โครงสร้างแบบรูปพื้นฐาน	58
รูปที่ 4.3 โครงสร้างของแบบรูปโดเมนความต้องการ	59
รูปที่ 4.4 โครงสร้างของแบบรูปกระบวนการความต้องการ	60
รูปที่ 4.5 โครงสร้างของแบบรูปความต้องการส่วนความปลอดภัย.....	61
รูปที่ 4.6 ลักษณะการอ้างอิงแบบรูปที่เกี่ยวข้อง.....	63
รูปที่ 4.7 รูปแบบขององค์ประกอบเป้าหมายของแบบรูปโดเมนความต้องการ	65
รูปที่ 4.8 รูปแบบองค์ประกอบปัญหาโดเมนของแบบรูปโดเมนความต้องการ	66
รูปที่ 4.9 รูปแบบองค์ประกอบปัญหาการพัฒนาของแบบรูปโดเมนความต้องการ	68
รูปที่ 4.10 รูปแบบองค์ประกอบลักษณะโดเมนของแบบรูปโดเมนความต้องการ	69
รูปที่ 4.11 รูปแบบองค์ประกอบสภาพแวดล้อมของแบบรูปโดเมนความต้องการ	71
รูปที่ 4.12 รูปแบบองค์ประกอบข้อจำกัดและกฎเกณฑ์ของแบบรูปโดเมนความต้องการ	72
รูปที่ 4.13 รูปแบบองค์ประกอบความต้องการของแบบรูปโดเมนความต้องการ	73
รูปที่ 4.14 รูปแบบองค์ประกอบความเชื่อมโยงและข้อจำกัดของความความต้องการของแบบรูปโดเมน ความต้องการ	74
รูปที่ 4.15 รูปแบบองค์ประกอบผู้เกี่ยวข้องและมุมมองต่างๆ ของแบบรูปโดเมนความต้องการ	75
รูปที่ 4.16 รูปแบบองค์ประกอบวิธีการเพื่อแก้ปัญหาโดเมนของแบบรูปโดเมนความต้องการ	76
รูปที่ 4.17 รูปแบบองค์ประกอบแบบรูปกระบวนการของแบบรูปโดเมนความต้องการ	77
รูปที่ 4.18 รูปแบบองค์ประกอบจุดประสงค์ของแบบรูปกระบวนการความต้องการ	78

รูปที่ 4.19	รูปแบบองค์ประกอบปัญหากระบวนการของแบบรูปกระบวนการความต้องการ	79
รูปที่ 4.20	รูปแบบองค์ประกอบกลุ่มประเภทของระบบของแบบรูปกระบวนการความต้องการ	80
รูปที่ 4.21	รูปแบบองค์ประกอบสถานการณ์ของแบบรูปกระบวนการความต้องการ	81
รูปที่ 4.22	รูปแบบองค์ประกอบเครื่องมือของแบบรูปกระบวนการความต้องการ	81
รูปที่ 4.23	รูปแบบองค์ประกอบวิธีการของแบบรูปกระบวนการความต้องการ	82
รูปที่ 4.24	รูปแบบองค์ประกอบประสบการณ์ของแบบรูปกระบวนการความต้องการ	83
รูปที่ 4.25	รูปแบบองค์ประกอบค่าใช้จ่ายของแบบรูปกระบวนการความต้องการ	83
รูปที่ 4.26	รูปแบบองค์ประกอบการยอมรับอันตรายของแบบรูปกระบวนการความต้องการ	85
รูปที่ 4.27	รูปแบบองค์ประกอบความต้องการความปลอดภัยของแบบรูปกระบวนการความต้องการ	86
รูปที่ 4.28	รูปแบบองค์ประกอบนัยความปลอดภัยของแบบรูปกระบวนการความต้องการ	87
รูปที่ 4.29	รูปแบบองค์ประกอบความเสี่ยงและอันตรายของแบบรูปกระบวนการความต้องการ	88
รูปที่ 4.30	รูปแบบองค์ประกอบอุบัติเหตุของแบบรูปกระบวนการความต้องการ	89
รูปที่ 5.1	ลักษณะของระยะห่างที่ปลอดภัย	105
รูปที่ 5.2	การส่งการหยุดไฟเมื่อถึงจุดเสี่ยง	105
รูปที่ 5.3	เงื่อนไขการเปิดประตู	113
รูปที่ 5.4	โครงสร้างการกำหนดเกี่ยวกับตำแหน่งและความเร็วของรถไฟ	120
รูปที่ 6.1	การใช้แบบรูปความต้องการในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์	128
รูปที่ 6.2	การใช้งานแบบรูปความต้องการในกระบวนการวิศวกรรมซอฟต์แวร์แบบน้ำตก	129
รูปที่ 6.3	การใช้งานแบบรูปความต้องการในกระบวนการวิศวกรรมซอฟต์แวร์แบบทำซ้ำ	130
รูปที่ 7.1	ส่วนต่างๆของเครื่องมือที่ช่วยในการสร้างและใช้งานแบบรูป	135
รูปที่ 7.2	หน้าที่การทำงานของเครื่องมือในส่วนการสร้างแบบรูป	136
รูปที่ 7.3	หน้าที่การทำงานของเครื่องมือในส่วนการใช้งานแบบรูป	137
รูปที่ 7.4	หน้าที่การใช้งานของเครื่องมือในส่วนการประเมินแบบรูป	138
รูปที่ 7.5	แผนภาพคอมโพเนนต์ของเครื่องมือที่พัฒนา	139
รูปที่ 7.6	ส่วนประกอบของเครื่องมือ	140
รูปที่ 7.7	แผนภาพแสดงชาร์ตแสดงวงจรชีวิตข้อมูลแบบรูป	141
รูปที่ 7.8	แผนภาพคลาสของเครื่องมือที่พัฒนา	141
รูปที่ 7.9	แผนภาพลำดับงานในการสร้างแบบรูปขึ้นมาใหม่	142
รูปที่ 7.10	แผนภาพลำดับงานในการดูหรือใช้งานแบบรูป	142

รูปที่ 7.11 แผนภาพลำดับงานในการแก้ไขข้อมูลในแบบรูป.....	143
รูปที่ 7.12 แผนภาพลำดับงานในการประเมินแบบรูป.....	143
รูปที่ 7.13 หน้าจอเครื่องมือเมื่อเริ่มทำงาน.....	145
รูปที่ 7.14 การใช้เครื่องมือเปิดอ่านแบบรูปในรูปแบบเอกสารเอ็กซ์เอ็มแอล	145
รูปที่ 7.15 หน้าจอแสดงลักษณะ โดเมนของแบบรูปโดเมนความต้องการ	146
รูปที่ 7.16 ตัวอย่างการแสดงรายละเอียดของแบบรูปโดยเครื่องมือ.....	146
รูปที่ 7.17 ตัวอย่างการแสดงรายละเอียดของแบบรูปโดยเครื่องมือ.....	147
รูปที่ 7.18 ตัวอย่างการแสดงรายละเอียดของแบบรูปโดยเครื่องมือ.....	147
รูปที่ 7.19 ตัวอย่างการแก้ไขหรือสร้างแบบรูปความต้องการ โดยเครื่องมือ	148
รูปที่ 7.20 ตัวอย่างการแก้ไขหรือสร้างแบบรูปความต้องการ โดยเครื่องมือ	148
รูปที่ 7.21 ตัวอย่างการแก้ไขหรือสร้างแบบรูปความต้องการ โดยเครื่องมือ	149
รูปที่ 7.22 ตัวอย่างการเปิดเอกสารที่สร้างขึ้นจากข้อมูลในแบบรูปที่เลือก	149
รูปที่ 7.23 ตัวอย่างเอกสารแบบรูปความต้องการที่เครื่องมือสร้างขึ้น	150

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

กระบวนการพัฒนาซอฟต์แวร์ในปัจจุบันมีการเปลี่ยนแปลงไปจากอดีตอย่างมาก ไม่ใช่เพียงเพื่อพัฒนาระบบให้สามารถเสร็จสมบูรณ์ และทันตามเวลาที่กำหนดเท่านั้น แต่การพัฒนาซอฟต์แวร์ในปัจจุบันยังให้ความสำคัญในเรื่องอื่นๆ อีก เช่น ประสิทธิภาพที่จะได้ (Performance) ความถูกต้องของระบบ (Correctness) การนำกลับมาใช้ใหม่ (Reuse) เป็นต้น

การนำกลับมาใช้ใหม่ในกระบวนการพัฒนาซอฟต์แวร์มักนำมาใช้ในขั้นตอนการเขียนโปรแกรม (Coding) เป็นส่วนใหญ่ หรือเป็นการนำกลับมาใช้ใหม่ทางด้านแนวคิด (Reuse Concept) ในกระบวนการวิเคราะห์และออกแบบระบบ (System Analysis and Design) แต่ยังมีอีกหลายส่วนในกระบวนการพัฒนาซอฟต์แวร์ที่ไม่สามารถนำกลับมาใช้ใหม่ได้อย่างมีประสิทธิภาพ

แบบรูป (Pattern) [1, 2] คือ การนำกลับมาใช้ใหม่ประเภทหนึ่ง แนวคิดของแบบรูป เป็นการอาศัยประสบการณ์จากการแก้ปัญหาที่เกิดขึ้นซ้ำๆ โดยนำองค์ความรู้ (Knowledge) การทำอย่างไร (How-to) มาแนะนำเสนอแนวทางในการแก้ปัญหา ซึ่งเป็นการนำกลับมาใช้ใหม่ในระดับความคิดรวบยอด โดยพิจารณาไปยังปัญหาที่เกิดขึ้นเพื่อพิจารณาหาผลการแก้ปัญหาที่เหมาะสม แรกเริ่มแบบรูปได้ถูกนำเสนอในงานด้านสถาปัตยกรรม[3] และถือว่าประสบความสำเร็จอย่างมากในการนำมาใช้กับงานวิศวกรรมซอฟต์แวร์ โดยเฉพาะในการนำไปใช้กับการออกแบบ [4]

แบบรูปความต้องการ (Requirement Pattern) [5-9] คือ แบบรูปที่นำมาใช้กับวิศวกรรมความต้องการซอฟต์แวร์ โดยใช้ในการแก้ปัญหาสำหรับกระบวนการทางวิศวกรรมความต้องการซอฟต์แวร์ตลอดจนเพื่อให้ได้ผลลัพธ์ที่เป็นข้อกำหนดของระบบที่เห็นพ้องร่วมกัน ส่วนโครงสร้างแบบรูปความต้องการ (Requirement Pattern Structure) คือ ความสัมพันธ์และองค์ประกอบของแบบรูปความต้องการ ซึ่งแสดงความเกี่ยวข้องของแบบรูปแต่ละชนิด ลักษณะการอ้างอิง และรายละเอียดขององค์ประกอบแต่ละส่วนในแบบรูป ดังนั้นแบบรูปความต้องการจะช่วยในการถ่ายทอดองค์ความรู้ที่ใช้แก้ปัญหาในงานวิศวกรรมความต้องการซอฟต์แวร์ ซึ่งจะช่วยลดเวลาในการพัฒนาซอฟต์แวร์ และเพิ่มประสิทธิภาพของตัวระบบที่พัฒนา แบบรูปจะเป็นวิธีการและเครื่องมือใหม่ที่จะช่วยให้กระบวนการพัฒนาซอฟต์แวร์ประสบความสำเร็จ โดยมีผลของการนำไปใช้แก้ปัญหาก่อนหน้านี้เป็นสิ่งยืนยัน

การสร้างแบบรูปความต้องการต้องอาศัยประสบการณ์และองค์ความรู้ที่มีอยู่ ซึ่งจะออกมาเป็นแบบรูปที่ดีได้นั้นต้องอาศัยโครงสร้างแบบรูปความต้องการที่เหมาะสม โดยงานวิจัยนี้ได้ให้ความสำคัญ

กับการออกแบบโครงสร้างแบบรูปความต้องการเป็นหลัก เพื่อหวังว่าจะได้โครงสร้างแบบรูปความต้องการที่เหมาะสมกับการนำไปใช้ โดยเฉพาะการนำไปใช้กับระบบปลอดภัยเชิงวิกฤติ

ระบบปลอดภัยเชิงวิกฤติ (Safety-Critical System) [10-12] เป็นระบบที่ต้องให้ความสำคัญกับความถูกต้องและผลการทำงานของระบบอย่างเข้มงวด เนื่องจากความผิดพลาดที่เกิดขึ้นกับระบบจะส่งผลให้เกิดความสูญเสียโดยตรงต่อชีวิต การพัฒนาซอฟต์แวร์สำหรับระบบประเภทนี้จึงมีกระบวนการที่ซับซ้อนและเจาะจงกว่าระบบทั่วไป และต้องอาศัยประสบการณ์ในการวิเคราะห์อันตราย (Hazard Analysis) ที่เกิดขึ้น ซึ่งจะเห็นว่าการพัฒนาซอฟต์แวร์สำหรับระบบปลอดภัยเชิงวิกฤติ จำเป็นต้องอาศัยความเข้าใจและประสบการณ์ในการพัฒนาระบบเป็นอย่างมาก ดังนั้นแบบรูปจึงมีความเหมาะสมที่จะนำมาช่วยในเรื่องของการขาดแคลนประสบการณ์ ทำให้สามารถมั่นใจในความถูกต้องที่จะได้

แบบรูปความต้องการสำหรับระบบปลอดภัยเชิงวิกฤติ จะช่วยในการแก้ปัญหาที่เคยเกิดขึ้นมาในอดีต โดยไม่จำเป็นต้องอาศัยประสบการณ์ของผู้พัฒนาซอฟต์แวร์มากนัก ทำให้การพัฒนาซอฟต์แวร์ที่ใช้แบบรูป มีความถูกต้อง ครอบคลุมปัญหา และทำได้รวดเร็ว โดยเฉพาะระบบปลอดภัยเชิงวิกฤติซึ่งเคร่งครัดกับความถูกต้องของการทำงาน และข้อกำหนดของระบบ

ในปัจจุบันยังไม่มีงานที่เสนอโครงสร้างของแบบรูปที่เหมาะสมกับงานวิศวกรรมความต้องการซอฟต์แวร์ที่ใช้กับระบบปลอดภัยเชิงวิกฤติ ดังนั้นในงานวิจัยนี้จึงเสนอ โครงสร้างแบบรูปความต้องการ ซึ่งจะนำไปใช้เป็นแม่แบบในการสร้างแบบรูปความต้องการ โดยเฉพาะสำหรับระบบปลอดภัยเชิงวิกฤติ ดังนั้น โครงสร้างแบบรูปที่เสนอมจะสามารถนำไปใช้ในการสร้างแบบรูปเพื่อถ่ายทอดการแก้ปัญหาและประสบการณ์ในการพัฒนาระบบปลอดภัยเชิงวิกฤติได้

ในงานวิจัยนี้ยังนำเสนอแนวทางในการนำแบบรูปความต้องการที่เสนอไปใช้งานด้วย เพื่อให้กระบวนการพัฒนาซอฟต์แวร์ที่ใช้แบบรูปทำได้ง่าย มีประสิทธิภาพมากขึ้น และเกิดการนำกลับมาใช้ใหม่ สำหรับกระบวนการพัฒนาซอฟต์แวร์โดยรวม

1.2 วัตถุประสงค์ของการวิจัย

- 1) วิเคราะห์และออกแบบโครงสร้างแบบรูปความต้องการ ที่สนับสนุนกระบวนการพัฒนาซอฟต์แวร์ของระบบปลอดภัยเชิงวิกฤติ
- 2) สร้างเครื่องมือเพื่อช่วยในการสร้างและบริหารแบบรูปความต้องการสำหรับระบบปลอดภัยเชิงวิกฤติ
- 3) สร้างวิธีการและกระบวนการ ในการนำแบบรูปความต้องการ ไปใช้ในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์

1.3 ขอบเขตของการวิจัย

1.3.1 นำเสนอโครงสร้างและองค์ประกอบของแบบรูป สำหรับกระบวนการวิศวกรรมความต้องการซอฟต์แวร์สำหรับระบบปลอดภัยเชิงวิกฤต

1.3.2 แบบรูปความต้องการที่ได้ต้องช่วยในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ดังนี้

- 1) สามารถช่วยในกระบวนการเก็บความต้องการซอฟต์แวร์ได้
- 2) สามารถช่วยในกระบวนการวิเคราะห์ความต้องการซอฟต์แวร์ได้
- 3) สามารถช่วยในกระบวนการจัดทำข้อกำหนดความต้องการซอฟต์แวร์ได้
- 4) สามารถช่วยในกระบวนการประเมินความต้องการซอฟต์แวร์ได้

1.3.3 สร้างเครื่องมือเพื่อใช้ในการสร้างและใช้งานแบบรูปความต้องการ โดยมีคุณสมบัติดังนี้

- 1) เครื่องมือสามารถช่วยในการสร้างแบบรูปความต้องการ
- 2) เครื่องมือช่วยในการนำแบบรูปไปใช้งาน
- 3) เครื่องมือช่วยในการประเมินแบบรูป

1.3.4 ออกแบบกระบวนการในการบริหารและใช้แบบรูปความต้องการ ที่จะสามารถช่วยในการนำแบบรูป และเครื่องมือ ไปใช้งานได้อย่างมีประสิทธิภาพ

1.3.5 ทำการวัดและการประเมินการนำแบบรูปความต้องการไปใช้ โดยวิธีการตอบคำถามเพื่อทดสอบการทำงานในงานวิศวกรรมความต้องการซอฟต์แวร์ โดยสร้างแบบรูปสำหรับสถานการณ์ที่ทดสอบและใช้คนทำการทดสอบเปรียบเทียบการใช้และไม่ใช้แบบรูป

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ช่วยแก้ปัญหาในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ โดยอาศัยหลักการของแบบรูป ซึ่งจะเป็นการถ่ายทอดและใช้ประสบการณ์ช่วยให้โครงการพัฒนาซอฟต์แวร์ประสบผลสำเร็จ
- 2) ช่วยให้การพัฒนาระบบโดยเฉพาะระบบปลอดภัยเชิงวิกฤตทำได้อย่างมีประสิทธิภาพมากขึ้น ลดความเสียหายที่เกิดจากกระบวนการวิศวกรรมความต้องการที่ผิดพลาดหรือไม่สมบูรณ์ โดยเฉพาะกับระบบปลอดภัยเชิงวิกฤต ซึ่งอาจเป็นอันตรายถึงชีวิตได้

- 3) ได้แนวทางใหม่เพื่อนำมาใช้เรือสนับสนุนกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ให้มีประสิทธิภาพยิ่งขึ้น ซึ่งมีจุดเด่นคือ การนำกลับมาใช้ใหม่ โดยเฉพาะในแง่ของประสิทธิภาพ และปัญหาที่เกิดขึ้นกับกระบวนการวิศวกรรมความต้องการซอฟต์แวร์

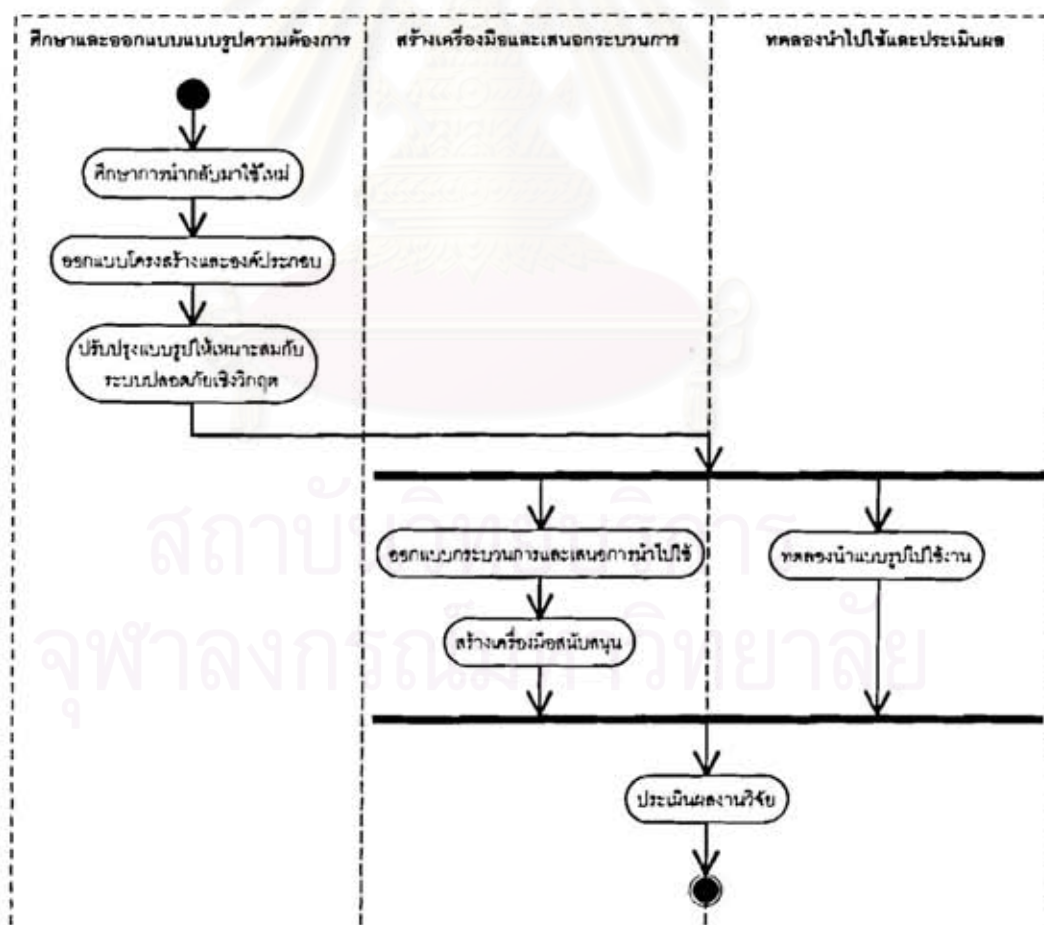
1.5 วิธีดำเนินการวิจัย

ในงานวิจัยนี้ ผู้วิจัยได้แบ่งลำดับของกระบวนการในการวิจัยเป็น 3 ส่วน คือ

ส่วนที่ 1 การศึกษาและออกแบบแบบรูปความต้องการ โดยเฉพาะในส่วนความต้องการซอฟต์แวร์สำหรับระบบปลอดภัยเชิงวิกฤต

ส่วนที่ 2 การสร้างเครื่องมือและกระบวนการในการนำแบบรูปความต้องการ ไปใช้

ส่วนที่ 3 ทำการทดลองและประเมินผลของการนำแบบรูปความต้องการ ไปใช้ โดยมีลำดับและขั้นตอนในงานวิจัยทั้งหมดแสดงในรูปที่ 1.1



รูปที่ 1.1 วิธีและขั้นตอนในการวิจัย

1.5.1 การศึกษาและออกแบบแบบรูปความต้องการซอฟต์แวร์

ในส่วนนี้เป็นการศึกษาและสรุปแนวทางเพื่อนำแบบรูปมาใช้กับงานวิศวกรรมความต้องการซอฟต์แวร์ และขยายแนวคิดไปสู่ระบบปลอดภัยเชิงวิกฤต เพื่อสามารถครอบคลุมความต้องการในส่วนของการปลอดภัย เช่น สามารถวิเคราะห์อันตรายในระบบได้ เป็นต้น

1.5.1.1 ศึกษาการนำกลับมาใช้ใหม่สำหรับงานวิศวกรรมความต้องการซอฟต์แวร์

ทำการศึกษารายละเอียดประกอบ คุณสมบัติ และงานวิศวกรรมความต้องการซอฟต์แวร์ เพื่อหาแนวทาง และวิธีการที่จะสามารถนำความต้องการซอฟต์แวร์กลับมาใช้ใหม่ได้ โดยพิจารณาใน 2 แนวทาง คือ

- 1) จากปัญหาที่เกิดขึ้นเป็นประจำจะสามารถแก้ปัญหาได้อย่างไร
- 2) จากแนวทางที่เป็นวิธีปฏิบัติ (Practice) มีการนำมาใช้อย่างไร

1.5.1.2 ออกแบบโครงสร้างแบบรูปสำหรับการนำไปใช้แก้ปัญหาในวิศวกรรมความต้องการซอฟต์แวร์

นำเอาลักษณะของแบบรูปมาใช้กับงานวิศวกรรมความต้องการซอฟต์แวร์ โดยอาศัยโครงสร้างตามแนวทางในการสร้างแบบรูป [13] และผนวกเข้ากับกระบวนการทางวิศวกรรมความต้องการซอฟต์แวร์ ซึ่งผู้วิจัยให้ความสนใจใน 2 ด้านด้วยกัน คือ ในมุมมองที่เป็นไปตามโดเมนของระบบที่จะพัฒนา และมุมมองของกระบวนการหรือเทคนิคขั้นตอนที่จะนำมาใช้ในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ ซึ่งทำให้สามารถสร้างแบบรูปขึ้นมาใหม่ได้เป็น 2 ประเภท ดังนี้

- 1) แบบรูปความต้องการเพื่อแก้ปัญหาที่ขึ้นกับโดเมน

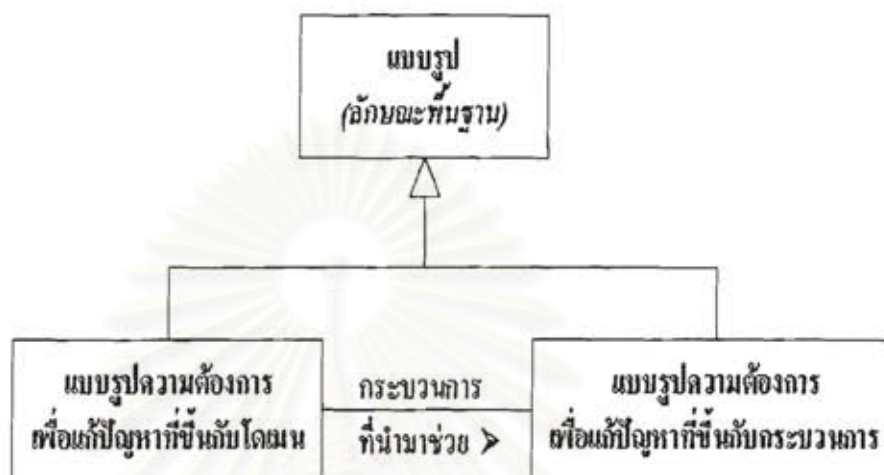
โดยนำเสนอลักษณะและองค์ประกอบของความต้องการซึ่งสอดคล้องกับลักษณะโดเมนของปัญหา ซึ่งเป็นข้อดีและมีลักษณะคล้ายกับงานของ S.Konrad และ Betty H.C.Cheng [6] และงานของ Toshihiko Tsumaki [8]

- 2) แบบรูปความต้องการเพื่อแก้ปัญหาที่ขึ้นกับกระบวนการ

นำเสนอแบบรูปสำหรับแก้ปัญหาในด้านเทคนิคและกระบวนการ โดยเป็นการกำหนดขอบเขตของแบบรูปกว้างๆ ที่ครอบคลุมลักษณะกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ ซึ่งจะคล้ายกับงานของ Lars Hagge และ Kathrin Lappe [9]

ถึงอย่างไรทั้งสองส่วนก็ถือได้ว่ามีลักษณะเป็นแบบรูปเหมือนกัน เพียงแต่เป็นแบบรูปที่ใช้ในการแก้ปัญหาคนละด้าน คือ ด้านกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ กับด้าน

โดเมนของระบบที่ทำการเก็บความต้องการซอฟต์แวร์ ซึ่งสามารถแสดงโครงสร้างความสัมพันธ์ได้ด้วยแผนภาพคลาส (Class Diagram) ดังรูปที่ 1.2



รูปที่ 1.2 แนวคิดของโครงสร้างแบบรูปความต้องการในขั้นต้น

1.5.1.3 ปรับปรุงโครงสร้างแบบรูปเพื่อให้เหมาะสมกับวิศวกรรมความต้องการซอฟต์แวร์สำหรับระบบปลอดภัยเชิงวิกฤต

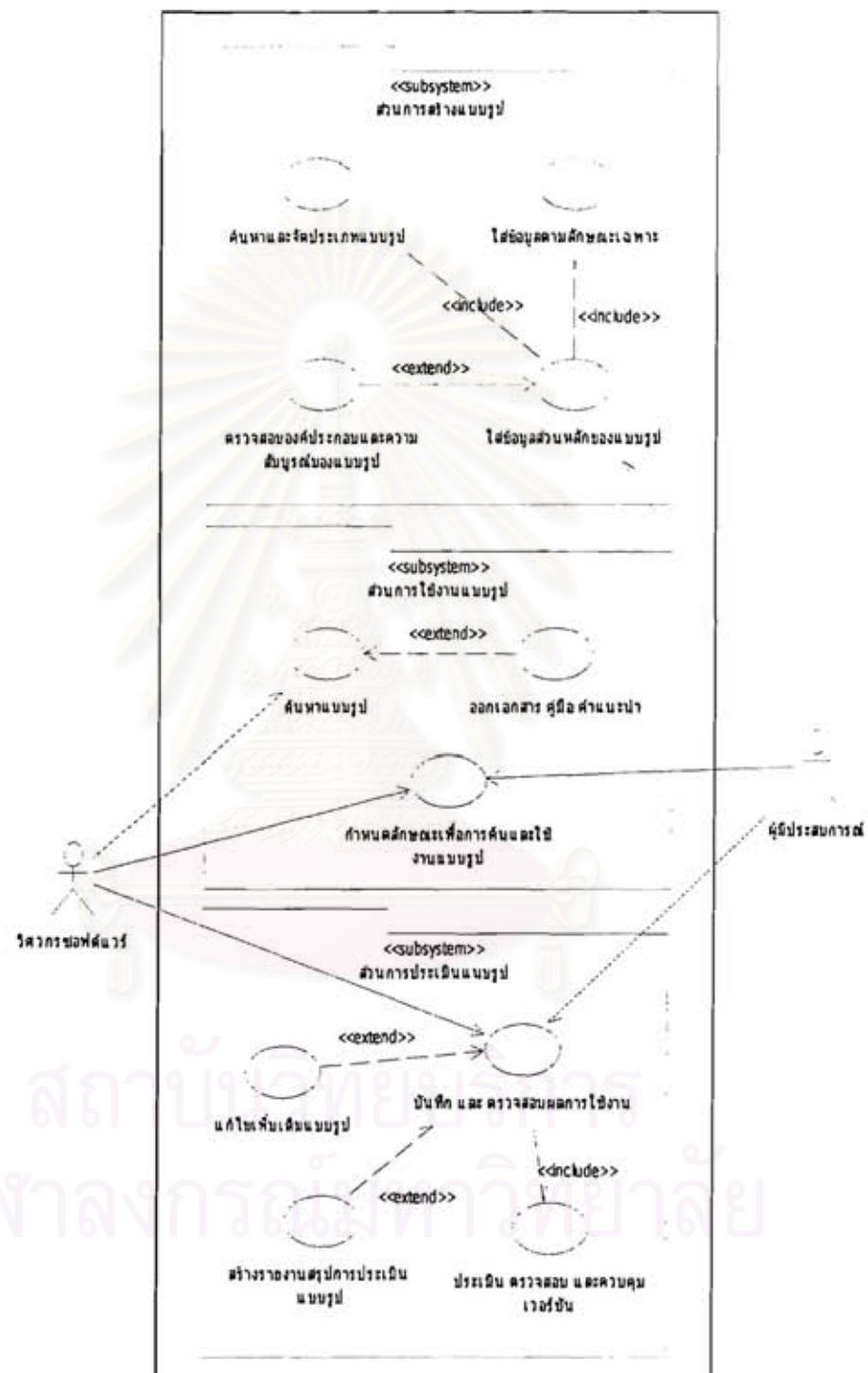
ศึกษาลักษณะของวิศวกรรมความต้องการซอฟต์แวร์ของระบบปลอดภัยเชิงวิกฤตรวมไปถึงคุณลักษณะของระบบปลอดภัยเชิงวิกฤตที่เข้ามาเกี่ยวข้องกับลักษณะความต้องการ เช่น อันตราย ความเสี่ยง (Risk) เป็นต้น ซึ่งรายละเอียดเหล่านี้จะไปขยายองค์ประกอบของแบบรูปโดเมนความต้องการ

1.5.2 สร้างเครื่องมือและกระบวนการในการนำแบบรูปไปใช้

1.5.2.1 สร้างเครื่องมือเพื่อช่วยในการสร้างและใช้งานแบบรูปความต้องการ

เพื่อให้การสร้างและใช้งานแบบรูปทำได้ง่ายและมีประสิทธิภาพยิ่งขึ้น เครื่องมือจึงเป็นสิ่งหนึ่งที่จะนำมาช่วยในส่วนนี้ โดยจะพัฒนาเครื่องมือสำหรับสร้างและใช้งานแบบรูปความต้องการที่สอดคล้องกับโครงสร้างที่ได้การออกแบบ ซึ่งจะมีการทำงานดังรูปที่ 1.3 โดยเครื่องมือจะช่วยแนะนำและอธิบายรายละเอียดองค์ประกอบของแบบรูป เครื่องมือจะช่วยในการสร้าง ใช้งาน และประเมินแบบรูป

เครื่องมือที่สร้างขึ้นมีหน้าที่ความสามารถเบื้องต้นแสดงในแผนภาพยูสเคส (Use Case Diagram) ดังรูปที่ 1.3



รูปที่ 1.3 หน้าทีการทำงานของเครื่องมือที่ช่วยในการสร้างและใช้งานแบบรูป

1.5.2.2 ออกแบบกระบวนการและเสนอการนำแบบรูปไปใช้งาน

ทำการออกแบบกระบวนการในการใช้งานแบบรูป เพื่อใช้เป็นแนวทางในการส่งเสริมกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ให้มีประสิทธิภาพยิ่งขึ้น โดยจะสามารถนำไปใช้ร่วมกับงานวิศวกรรมความต้องการซอฟต์แวร์หรือกระบวนการพัฒนาซอฟต์แวร์ที่ใช้อยู่ในปัจจุบันได้

1.5.3 ทดลองนำแบบรูปความต้องการไปใช้และทำการประเมินผล

1.5.3.1 สร้างสถานการณ์และแบบรูปสำหรับระบบปลอดภัยเชิงวิกฤต เพื่อทดสอบการนำไปใช้

ในขั้นตอนนี้จะทำการทดสอบการใช้แบบรูปความต้องการสำหรับระบบปลอดภัยเชิงวิกฤต โดยการสร้างสถานการณ์จำลองในการทำความต้องการซอฟต์แวร์สำหรับระบบปลอดภัยเชิงวิกฤต โดยเลือกใช้กรณีศึกษาจากระบบควบคุมรถไฟอัตโนมัติ ซึ่งมีขั้นตอนดังนี้

- 1) ศึกษาข้อมูลของระบบป้องกันการเดินรถไฟอัตโนมัติ (Automatic Train Protection: ATP) เพื่อนำมาใช้สร้างสถานการณ์จำลอง
- 2) สร้างแบบรูปที่ครอบคลุมผลลัพธ์ที่ควรจะได้ในขั้นตอนวิศวกรรมความต้องการซอฟต์แวร์
- 3) กำหนดเงื่อนไขที่จะนำมาใช้ประเมินความสำเร็จของงานวิศวกรรมความต้องการซอฟต์แวร์และแบบรูปความต้องการ
- 4) กำหนดและจัดหาหน่วยทดลอง
- 5) ใช้หน่วยทดลองสำหรับ 2 กรณี คือ แบบที่จะใช้แบบรูป และไม่ใช้แบบรูป โดยใช้ 18 หน่วยทดลอง ทำการทดลองกรณีละ 9 ตัวอย่างเพื่อเปรียบเทียบผลลัพธ์กัน
- 6) ให้ความรู้เบื้องต้นแก่ผู้ทดลองเกี่ยวกับกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ และระบบปลอดภัยเชิงวิกฤต โดยกลุ่มที่ใช้แบบรูปจะได้รับความรู้เพิ่มเติมเกี่ยวกับแบบรูป และการนำแบบรูปไปใช้งาน
- 7) ทำการทดลองภายใต้การควบคุม และเงื่อนไขที่กำหนด
- 8) เก็บข้อมูลและผลลัพธ์ที่ได้จากการทดลอง

1.5.3.2 ทำการประเมินโครงสร้างแบบรูป และกระบวนการนำไปใช้ที่ออกแบบ

นำข้อมูลที่ได้จากการทดลองมาประเมินผลตามที่กำหนดไว้ เพื่อสรุปว่าแบบรูปสามารถนำมาใช้ได้จริง มีประสิทธิภาพเป็นอย่างไร กระบวนการในการใช้งานเป็นอย่างไร เกิดปัญหาใน

การใช้ที่ได้นำ และสรุปผลการใช้แบบรูปความต้องการสำหรับระบบปลอดภัยเชิงวิกฤตโดยรวม โดยอาศัยวิธีการทางสถิติในการประเมินและสรุปผล

ในขั้นสุดท้ายจะนำผลการประเมินและสรุปมาปรับปรุงโครงสร้างแบบรูปความต้องการ ให้มีความเหมาะสมมากขึ้น

1.6 บทความวิชาการที่ได้รับการตีพิมพ์

ในงานวิทยานิพนธ์นี้ผู้วิจัยได้เสนอผลงานร่วมกับอาจารย์ที่ปรึกษา เพื่อตีพิมพ์เป็นบทความ วิชาการระดับชาติ ดังแสดงในภาคผนวก ก. คือ บทความวิชาการเรื่อง “การนำความต้องการซอฟต์แวร์ กลับมาใช้ใหม่ด้วยแบบรูปความต้องการ (Software Requirements Reuse by Requirements Patterns)” ซึ่ง ได้นำเสนอและตีพิมพ์ในงาน “การประชุมทางวิชาการทางวิทยาศาสตร์และวิศวกรรมคอมพิวเตอร์ ระดับชาติ ครั้งที่ 9 (The 9th National Computer Science and Engineering Conference: NCSEC 2005)” ระหว่างวันที่ 27-28 ตุลาคม 2548 ณ มหาวิทยาลัยหอการค้าไทย กรุงเทพฯ ประเทศไทย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 แนวคิดและทฤษฎี

2.1.1 แบบรูปและแบบรูปสำหรับซอฟต์แวร์ (Patterns and Software Patterns)

2.1.1.1 ความหมายของแบบรูป

แบบรูป คือ ปัญหา (Problem) และผลเฉลย (Solution) ซึ่งเกิดขึ้นซ้ำๆกัน [2] โดยแบบรูปเป็นการนำผลเฉลยที่เคยเกิดขึ้นจากปัญหาใดปัญหาหนึ่ง มาแก้ปัญหที่เกิดขึ้นใหม่ที่มีสภาพแวดล้อม (Context) คล้ายๆ กับปัญหาในแบบรูปนั้น ดังนั้นทุกๆ แบบรูปจึงตั้งอยู่บนกฎสามส่วน (Three-Path Rule) [1] ซึ่งอธิบายว่า แบบรูปเป็นความสัมพันธ์ที่ชัดเจนระหว่างสภาพแวดล้อมที่แน่นอน ปัญหา และ ผลเฉลย

2.1.1.2 ประวัติความเป็นมาของแบบรูป

แบบรูปถูกนำเสนอครั้งแรกเพื่อใช้แก้ปัญหาในงานสถาปัตยกรรม [3] หลังจากนั้น W.Cunningham และ K.Beck ได้นำแบบรูปมาใช้ในการออกแบบส่วนต่อประสานผู้ใช้โดยภาษาสมอลล์ทอล์ค (Smalltalk) เป็นครั้งแรก [14] และได้รับความนิยมอย่างมากจากหนังสือ Design Patterns: Elements of Reusable Object-Oriented Software ของ E. Gamma และคณะ [4] หลังจากนั้นแบบรูปจึงมีการนำไปใช้ในวงการซอฟต์แวร์อย่างแพร่หลาย เช่น แบบรูปการวิเคราะห์ระบบ (Analysis Patterns) [15] แบบรูปการเขียนโปรแกรม (Idioms) [16] แบบรูปสถาปัตยกรรม (Architecture Patterns) [17] เป็นต้น

2.1.1.3 การจัดแบ่งองค์ประกอบของแบบรูป

แบบรูปประกอบด้วยองค์ประกอบหลัก 3 ส่วน คือ ปัญหา สภาพแวดล้อม และผลเฉลย แต่การนำเอาแบบรูปไปใช้นั้นอาจมีความจำเป็นต้องมีองค์ประกอบอื่นๆ เพื่อช่วยให้แบบรูปมีความสมบูรณ์และง่ายต่อการนำไปใช้งาน เช่น ชื่อแบบรูป (Pattern Name) แรงชักจูง (Force) สถานะผลลัพธ์ (Result Context) แบบรูปที่เกี่ยวข้อง (Related Patterns) รวมทั้งองค์ประกอบสนับสนุนที่จะช่วยในการอธิบายผลเฉลยของแบบรูปนั้นๆ เช่น ตัวอย่าง (Example) แผนภาพคลาส (Class Diagram) ซึ่งใช้ในแบบรูปการออกแบบ เป็นต้น องค์ประกอบเหล่านี้จะขึ้นอยู่กับ การนำแบบรูปไปใช้งาน โดยมีจุดประสงค์เพื่อที่สามารถอธิบาย สนับสนุนการใช้ หรือช่วยในการทำ ความเข้าใจปัญหาหรือการแก้ปัญหาที่น่าเสนอได้ชัดเจนขึ้น

แบบรูปการออกแบบเป็นแบบรูปที่ได้รับความนิยมในการนำไปใช้เป็นอย่างกว้างขวาง และเป็นแบบรูปที่นำมาใช้กับกระบวนการพัฒนาซอฟต์แวร์อย่างได้ผล โดยแบบรูปการออกแบบ [4] มีองค์ประกอบดังต่อไปนี้

- 1) ชื่อแบบรูป (Pattern Name)
- 2) ชื่ออื่นที่เป็นที่รู้จักกัน (Also Known As)
- 3) แรงบันดาลใจ (Motivation)
- 4) เจตนา (Intent)
- 5) ผลที่ได้ (Consequences)
- 6) แบบรูปที่เกี่ยวข้อง (Related Patterns)
- 7) การนำไปใช้ที่ทราบ (Known Use)
- 8) ตัวอย่างโปรแกรม (Simple Code)
- 9) การนำไปปรับใช้ (Applicability)
- 10) การทำให้เกิดผล (Implementation)
- 11) ลักษณะทางโครงสร้าง (Structure)
- 12) ส่วนร่วมและสิ่งที่เข้ามาเกี่ยวข้อง (Participants)
- 13) คอลลาโบเรชัน (Collaboration)

องค์ประกอบของแบบรูปการออกแบบที่นำเสนอนี้เป็นของ E. Gamma และคณะ ซึ่งมีการนำไปใช้อย่างแพร่หลาย แต่ก็มีแบบรูปการออกแบบอื่นๆ อีก เช่น แบบรูปที่ใช้กับภาษาจาวาของ M. Grand [18] ซึ่งมีองค์ประกอบต่างออกไปอีกด้วย

สำหรับงานวิทยานิพนธ์นี้ได้นำเสนอโครงสร้างและองค์ประกอบของแบบรูปความต้องการ ดังนั้นจึงมีองค์ประกอบที่แตกต่างออกไปจากแบบรูปการออกแบบ โดยองค์ประกอบสำหรับแบบรูปความต้องการนั้นจะเกี่ยวข้องกับงานวิศวกรรมความต้องการซอฟต์แวร์เป็นหลัก โดยจะนำไปใช้ในกระบวนการพัฒนาซอฟต์แวร์ในช่วงการเก็บความต้องการ การวิเคราะห์ความต้องการ หรือขั้นตอนอื่นๆ ในวิศวกรรมความต้องการซอฟต์แวร์ดังกล่าวต่อไป

2.1.1.4 การพิจารณาองค์ประกอบพื้นฐานและตัวภาษาของแบบรูป

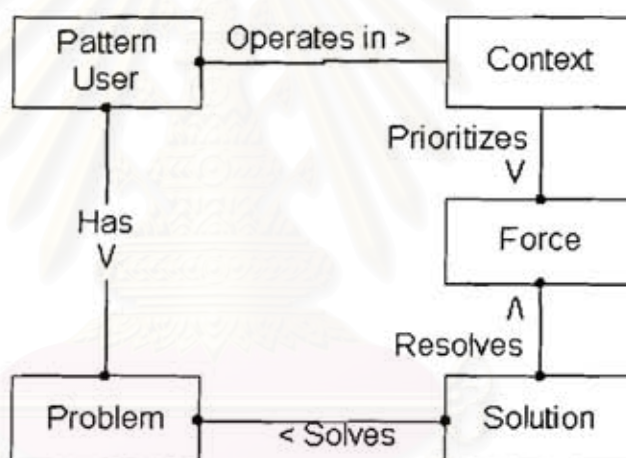
การพิจารณาองค์ประกอบพื้นฐาน และการพิจารณาตัวภาษาของแบบรูป คือ การพิจารณาในส่วนที่เป็นองค์ประกอบที่เป็นแกนหลักของแบบรูป ซึ่งประกอบด้วย ปัญหา สภาพแวดล้อม และผลเฉลย ควบคู่ไปกับภาษาแบบรูป

การนำเสนอผลเฉลยสำหรับปัญหานั้นจำเป็นต้องให้ความสำคัญกับมุมมองของการนำกลับมาใช้ใหม่ ซึ่งจะต้องพิจารณาเพื่อหาวิธีที่ง่ายในการเสนอผลเฉลยโดยเฉพาะที่มีความซับซ้อน

บนสภาพแวดล้อมที่แตกต่างกัน ซึ่งจะส่งผลให้แต่ละแบบรูปมีการแก้ปัญหาที่เฉพาะเจาะจงและชัดเจน โดยแบบรูปควรเป็นเอกเทศ มีสมบูรณ์ในตัวเอง และจำกัดความสัมพันธ์ในการอ้างอิงเพื่อเชื่อมโยงไปยังแบบรูปอื่นๆ สำหรับการใช้งานหนึ่งๆ ให้น้อยที่สุด

2.1.1.5 การพิจารณาโครงสร้างแบบรูปให้เหมาะสม

การพิจารณาแบบรูปให้เหมาะสมอีกนัยหนึ่งคือ การกำหนดองค์ประกอบสำหรับแบบรูปให้มีความสมบูรณ์สอดคล้องกับการใช้งาน ข้อสำคัญในการกำหนดองค์ประกอบแบบรูปคือ การมั่นใจว่ามีข้อมูลสำคัญครบถ้วนเพียงพอ โดยองค์ประกอบที่อยู่ในแบบรูปจะพิจารณาเพื่อการนำแบบรูปไปใช้แก้ปัญหา โดยจะต้องไม่มากเกินไปจนทำให้เกิดความยุ่งยากและซับซ้อนในการใช้ ซึ่งจะมียองค์ประกอบเป็นโครงสร้างหลักดังรูปที่ 2.1



รูปที่ 2.1 ความสัมพันธ์ระหว่างองค์ประกอบของแบบรูป [13]

องค์ประกอบแกนหลักของแบบรูปประกอบด้วย

- 1) สภาพแวดล้อม
- 2) ปัญหา
- 3) ผลเฉลย
- 4) แรงชักจูง

การใช้แรงชักจูงจะช่วยให้มั่นใจว่าทำการเลือกใช้แบบรูปได้อย่างถูกต้องตรงกับปัญหา และสภาพการณ์ที่เป็นอยู่ และยังช่วยให้สามารถละแบบรูปนี้ไปได้โดยง่ายหากเห็นว่าไม่ตรงกับการนำไปใช้

จากองค์ประกอบหลักของแบบรูป จะเห็นได้ว่ายังไม่เพียงพอในการนำไปใช้ จำเป็นต้องมีองค์ประกอบเสริมที่จะนำมาใช้ด้วย ดังนี้

- 1) สภาวะผลลัพธ์
- 2) แบบรูปที่เกี่ยวข้อง
- 3) ตัวอย่าง
- 4) การนำไปใช้ที่ทราบ

โดยองค์ประกอบเสริมที่ยกมาแสดงนั้นเป็นองค์ประกอบที่ใช้กันทั่วไป ซึ่งที่ยกมาเป็นองค์ประกอบที่มีความเหมาะสมและเพียงพอสำหรับการนำมาใช้งานในแบบรูปความต้องการ

2.1.1.6 การใช้ชื่อและการอ้างอิงแบบรูป

การเรียกชื่อองค์ประกอบนอกเหนือไปจากองค์ประกอบที่เป็นพื้นฐานในโครงสร้างแบบรูป ซึ่งเป็นองค์ประกอบที่มีลักษณะเฉพาะเจาะจงและมีจุดมุ่งหมายเพื่อใช้ในลักษณะพิเศษ การกำหนดชื่อองค์ประกอบเหล่านี้จึงจำเป็นต้องสัมพันธ์กับความหมายตามที่มักจะนำไปใช้จริง ชื่อที่กำหนดควรจะมี ความหมายชัดเจนและเป็นตัวแทนของสิ่งที่แบบรูปจะพยายามอธิบาย

แบบรูปควรมีองค์ประกอบภายในที่สัมพันธ์และอ้างอิงถึงกันได้ เพื่อสามารถใช้งานทดแทนหรือประกอบกัน และควรอ้างอิงไปยังแบบรูปอื่นที่มีความเกี่ยวข้องกันโดยพิจารณา ลักษณะความสัมพันธ์ภายนอก

การสร้างแบบรูปเพื่อใช้งานตามองค์ประกอบที่กำหนด จำเป็นต้องใช้ให้ตรงกับ ความหมาย การอ้างอิงและความสัมพันธ์

2.1.1.7 การทำแบบรูปให้เข้าใจง่าย

ในการออกแบบแบบรูปควรกำหนดองค์ประกอบให้ง่ายต่อการทำความเข้าใจ ซึ่งควรพิจารณาในด้านต่างๆ ควรจะพิจารณาในด้านต่อไปนี้

1) ความเข้าใจเฉพาะกลุ่ม

การใช้งานและสร้างแบบรูปควรระวังในเรื่องของความหมายหรือการกำหนดในสิ่งที่จำกัดเฉพาะในกลุ่มผู้เกี่ยวข้องหรือคุ้นเคย อย่างไรก็ตามไม่ควรที่จะใช้ความหมายที่เป็นกลางจนเกินไป จะทำให้ขาดความชัดเจนและความหมายกว้างเกินไป

2) การใช้ศัพท์

การใช้คำศัพท์จำเป็นต้องเป็นคำที่ง่ายต่อการเข้าใจ และควรเป็นคำศัพท์ที่รู้จักโดยทั่วไป

3) การใช้สัญลักษณ์

สัญลักษณ์เป็นสิ่งสำคัญที่ช่วยในการเข้าใจและจดจำ อาจอยู่ในรูปแบบสัญลักษณ์พิเศษ รูปภาพ หรือแผนภาพ ที่เป็นที่รู้จัก ไม่ควรใช้สัญลักษณ์เฉพาะกลุ่ม หรือนำมาจากงานทางด้านอื่นที่ไม่เกี่ยวข้องกัน

2.1.1.8 การกำหนดโครงสร้างแบบรูป

โครงสร้างแบบรูป คือ การสรุปภาพรวมของแบบรูปที่สร้างขึ้นในกรณีที่มีแบบรูปที่ออกแบบมีความเป็นระบบ ซึ่งมีการทำงานร่วมกันของแบบรูปมากกว่า 1 แบบรูป โดยมีแบบรูปที่มีความสัมพันธ์อ้างอิงถึงกัน ควรสร้างส่วนสรุปแบบรูปเพื่อช่วยผู้ใช้งานแบบรูปทำความเข้าใจ และเลือกใช้แบบรูปได้อย่างมีประสิทธิภาพ

การสรุปแบบรูปแบ่งออกเป็น 2 ส่วน คือ

1) สรุปภาษาแบบรูป

การสรุปภาษาแบบรูป คือ การสรุปลักษณะองค์ประกอบและการใช้งานของแต่ละแบบรูปเพื่ออธิบายความหมายขององค์ประกอบส่วนต่างๆ ซึ่งจะช่วยให้เข้าใจแบบรูป และสามารถใช้แบบรูปตามองค์ประกอบที่มีได้อย่างถูกต้อง

2) สรุปปัญหาและผลเฉลย

การสรุปปัญหาและผลเฉลย คือ การสรุปในภาพรวมของแบบรูป เมื่อแบบรูปในระบบมีจำนวนมากขึ้น จำเป็นต้องจัดทำสรุปและแจกแจงแบบรูปเพื่อให้เลือกใช้ง่ายขึ้น

2.1.2 วิศวกรรมความต้องการซอฟต์แวร์ (Software Requirements Engineering)

วิศวกรรมความต้องการซอฟต์แวร์เป็นส่วนหนึ่งของวิศวกรรมซอฟต์แวร์ ถูกกำหนดขึ้นในช่วงการเริ่มต้นของกระบวนการพัฒนาซอฟต์แวร์ โดยมีจุดมุ่งหมายในการให้ได้มาซึ่งความต้องการด้านซอฟต์แวร์ที่ถูกต้องและชัดเจนเพื่อนำไปใช้ในการกำหนดระบบที่จะทำการพัฒนา ซึ่งมีกระบวนการที่สำคัญดังนี้ [19]

2.1.2.1 การเก็บรวบรวมความต้องการ (Requirements Elicitation)

เป็นกระบวนการที่จะเข้าไปเกี่ยวข้องกับผู้ใช้ระบบโดยตรง โดยมีจุดประสงค์ในการเก็บข้อมูลที่จะนำมาใช้ในการกำหนดตัวระบบ

2.1.2.2 การวิเคราะห์ความต้องการ (Requirements Analysis)

เป็นกระบวนการวิเคราะห์ความต้องการที่ได้มาว่าได้ครอบคลุมความต้องการซอฟต์แวร์ที่มีทั้งหมดหรือไม่

2.1.2.3 การจัดทำข้อกำหนดความต้องการ (Requirements Specification)

เป็นกระบวนการในการจัดทำข้อกำหนดตัวระบบ ให้รายละเอียดของระบบที่จะพัฒนาขึ้น ซึ่งเป็นผลลัพธ์ที่สำคัญที่ได้จากกระบวนการวิศวกรรมความต้องการซอฟต์แวร์

2.1.2.4 การประเมินความต้องการ (Requirements Validation)

เป็นการตรวจสอบความถูกต้องของความต้องการที่เก็บมา เช่น ความต้องการที่เก็บมา มีความสอดคล้องกันหรือไม่ มีการจัดระดับความสำคัญของความต้องการอย่างไร เป็นต้น

2.1.2.5 การบริหารความต้องการ (Requirements Management)

เป็นการบริหารกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ คอยควบคุมและดูแลคุณภาพความถูกต้องของความต้องการ การผลิตความต้องการ ตลอดจนการบริหารความต้องการที่มีการเปลี่ยนแปลง (Requirements Change)

องค์ประกอบของความต้องการซอฟต์แวร์ประกอบด้วย ความต้องการที่เป็นหน้าที่การทำงาน (Functional Requirements) และความต้องการที่ไม่ใช่หน้าที่การทำงาน (Non-Functional Requirements) ซึ่งเป็นตัวกำหนดและบ่งบอกคุณภาพของระบบ

กระบวนการวิศวกรรมความต้องการซอฟต์แวร์เป็นกระบวนการหนึ่งที่มีความสำคัญในการพัฒนาซอฟต์แวร์ ความผิดพลาดหรือไม่สมบูรณ์ที่เกิดขึ้นในกระบวนการนี้ มักก่อให้เกิดค่าใช้จ่ายที่สูงกว่าผลของความผิดพลาดที่เกิดจากกระบวนการพัฒนาซอฟต์แวร์ในส่วนอื่นๆ และมีความเกี่ยวข้องสัมพันธ์กับผู้ใช้หรือผู้เป็นเจ้าของระบบอย่างมาก โดยท้ายที่สุดแล้ววัตถุประสงค์ของวิศวกรรมความต้องการซอฟต์แวร์คือการให้ได้มาซึ่ง 3 สิ่งด้วยกัน [20] คือ

- 1) การยอมรับในความต้องการ (Agreed Requirements)
- 2) ข้อกำหนดของระบบ (System Specification)
- 3) แบบจำลองของระบบ (System Models)

2.1.3 ระบบปลอดภัยเชิงวิกฤต (Safety-Critical System)

ระบบปลอดภัยเชิงวิกฤต คือ ระบบซึ่งความผิดพลาดจะส่งผลให้สูญเสียชีวิตและก่อให้เกิดอันตรายร้ายแรง [10] เช่น ระบบนิวเคลียร์ โปรแกรมประยุกต์ในอุปกรณ์การแพทย์ ระบบควบคุมการจราจรทางอากาศ เป็นต้น

ระบบปลอดภัยเชิงวิกฤตแตกต่างกับระบบทั่วไป ที่ไม่อาจยอมรับความผิดพลาดในตัวระบบได้ ตลอดจนให้ความสำคัญกับทุกเหตุการณ์ที่อาจเกิดขึ้น

กระบวนการพัฒนาระบบปลอดภัยเชิงวิกฤต [21] มีความแตกต่างจากกระบวนการพัฒนาระบบทั่วไปค่อนข้างมาก นอกจากความสามารถของระบบที่เป็นไปตามความต้องการที่มีมาแล้ว ความปลอดภัยเป็นอีกส่วนหนึ่งที่ต้องให้ความสำคัญอย่างมาก ซึ่งต้องพิจารณาในเรื่อง ความเสี่ยง (Risk) อุบัติเหตุ (Accident) เหตุการณ์ (Incident/Near Miss) อันตราย (Hazard) ที่เกิดขึ้นด้วย

วิศวกรรมซอฟต์แวร์ สำหรับระบบปลอดภัยเชิงวิกฤตมีประเด็นสำคัญที่เป็นหัวใจของการพัฒนาระบบ 6 อย่าง [11] ดังนี้

- 1) การวิเคราะห์อันตราย
- 2) การวางข้อกำหนดและวิเคราะห์ความต้องการด้านความปลอดภัย (Safety Requirements Specification and Analysis)
- 3) การออกแบบสำหรับความปลอดภัย (Designing for Safety)
- 4) การทดสอบ (Testing)
- 5) การรับรองและมาตรฐาน (Certification and Standard)
- 6) ทรัพยากร (Resource)

ความหมายของระบบปลอดภัยเชิงวิกฤตนั้นไม่ได้มีความหมายเพียงแค่ซอฟต์แวร์ที่ควบคุมการทำงานของระบบ แต่ยังรวมไปถึงส่วนของฮาร์ดแวร์ และการดำเนินการด้วย และในหลายๆ ลักษณะได้มีการพัฒนาระบบปลอดภัยเชิงวิกฤตในรูปแบบของระบบฝังตัว ดังนั้นปัญหาความผิดพลาดในระบบปลอดภัยเชิงวิกฤตจึงไม่ได้พิจารณาที่ความผิดพลาดของซอฟต์แวร์อย่างเดียว แต่รวมไปถึงความผิดพลาดในตัวอุปกรณ์ฮาร์ดแวร์ ลักษณะการใช้งานที่ผิดๆ ความผิดพลาดของมนุษย์ เช่น ความประมาท การหลงลืม รวมทั้งภัยธรรมชาติที่เกิดขึ้นด้วย

อย่างไรก็ดีในการพัฒนาระบบปลอดภัยเชิงวิกฤต โดยเฉพาะในส่วนที่เกี่ยวข้องกับวิศวกรรมความต้องการซอฟต์แวร์ จะทำการพิจารณาสิ่งที่จะเป็นอันตรายต่อชีวิตโดยละเอียด ซึ่งมักจะใช้การวิเคราะห์อันตรายเป็นหลัก ระหว่างการพัฒนาซอฟต์แวร์จะทำการระมัดระวังผลที่เกิดตามความเสี่ยงเหล่านี้ด้วย โดยในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์สำหรับระบบปลอดภัยเชิงวิกฤตยังมีจุดประสงค์ที่สำคัญอีกอย่าง คือ การให้ได้มาซึ่งข้อกำหนดซอฟต์แวร์ที่สามารถป้องกัน ตรวจสอบ หรือตอบสนองต่อเหตุการณ์ที่เป็นอันตรายได้

การวิเคราะห์อันตรายเป็นขั้นตอนที่ต้องทำตลอดการพัฒนาระบบปลอดภัยเชิงวิกฤต และเป็นส่วนสำคัญในการวางแผนและกำหนดลักษณะของระบบในการเริ่มต้นการพัฒนา ซึ่งเกี่ยวข้องและใกล้ชิดกับกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ ดังนั้นการวิเคราะห์อันตรายจึงเป็นส่วนหนึ่งที่เกี่ยวข้องโดยตรงกับงานวิจัยนี้

การวิเคราะห์อันตรายมีด้วยกันหลายวิธีขึ้นกับจุดประสงค์ในการนำไปใช้ และอาจมีการใช้หลายๆวิธีร่วมกัน ตัวอย่างของวิธีการในการวิเคราะห์อันตราย ได้แก่

- 1) การวิเคราะห์ต้นไม้ความผิดพลาด (Fault Tree Analysis)
- 2) การวิเคราะห์ต้นไม้เหตุการณ์ (Event Tree Analysis)
- 3) การวิเคราะห์อันตรายและการดำเนินการ (Hazards and Operability Analysis)
- 4) การวิเคราะห์หนทางขัดข้องและผลกระทบ (Failure Modes and Effects Analysis)
- 5) การวิเคราะห์อันตรายด้วยสเตตแมชชีน (State Machine Hazard Analysis)

ในส่วนข้อกำหนดซอฟต์แวร์โดยมากมักนำวิธีเชิงรูปนัย (Formal Method) [22] มาใช้เพื่อกำหนดตัวระบบ และใช้ในการตรวจสอบ ในปัจจุบันมีผู้พยายามนำวิธีการหรือเทคนิคอื่นๆ มาช่วยในการเขียนข้อกำหนดซอฟต์แวร์นอกจากวิธีเชิงรูปนัยมากยิ่งขึ้น เช่น งานของ F. Modugno และคณะ [23] ซึ่งนำเสนอข้อกำหนดความต้องการซอฟต์แวร์ที่รวมเอาการวิเคราะห์อันตรายไว้

โดยสรุปแล้ววิศวกรรมความต้องการซอฟต์แวร์ที่ใช้กับระบบปลอดภัยเชิงวิกฤต มีความแตกต่างจากระบบประเภทอื่นๆ อย่างเห็นได้ชัด เช่น งานของ Donald Firesmith ที่นำเสนอเกี่ยวกับประเภทความต้องการในระบบปลอดภัยเชิงวิกฤต [24] ซึ่งทำให้เห็นลักษณะที่แตกต่างและความจำเพาะอย่างชัดเจนของตัวความต้องการ เช่น ตัวความต้องการในส่วนข้อจำกัดความปลอดภัย (Safety Constraints) เป็นต้น

2.1.4 ความต้องการที่เกี่ยวข้องกับความปลอดภัย (Safety-Related Requirements)

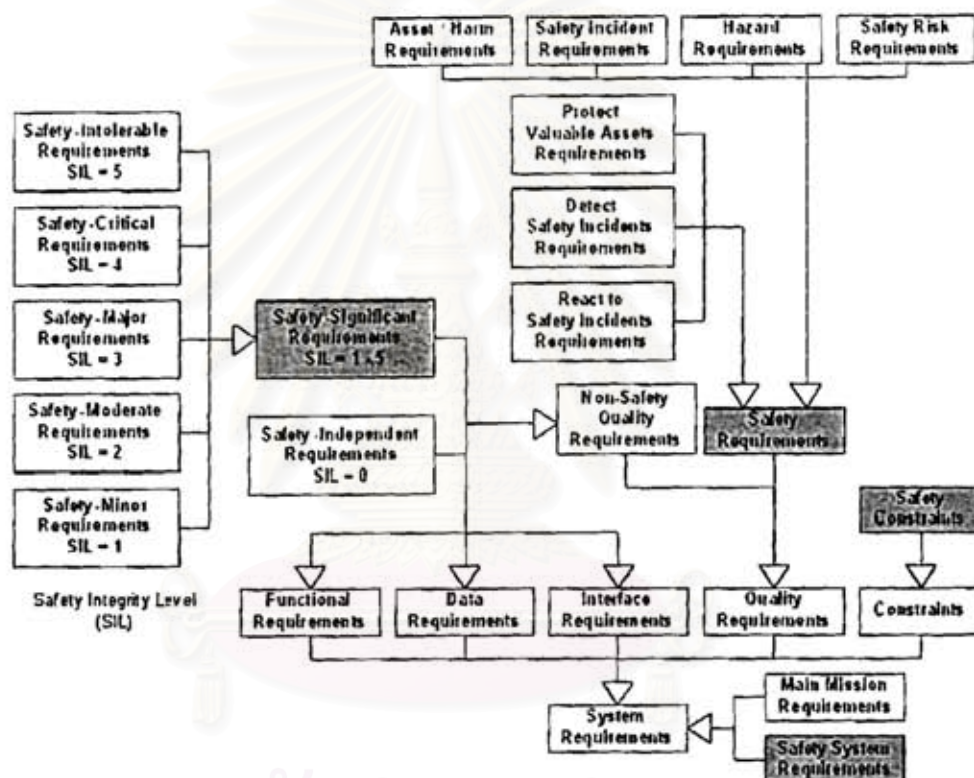
ความต้องการที่เกี่ยวข้องกับความปลอดภัย คือ ความต้องการในระบบปลอดภัยเชิงวิกฤต ซึ่งครอบคลุมทั้งส่วนของฮาร์ดแวร์ ซอฟต์แวร์ และการปฏิบัติงาน โดยเป็นความต้องการที่เกี่ยวข้องกับลักษณะความปลอดภัย ซึ่งเป็นตัวกำหนดระบบให้ปลอดภัยจากอันตรายและกำหนดหน้าที่และคุณภาพของซอฟต์แวร์ และได้ให้ความสำคัญกับอันตรายหรืออุบัติเหตุที่อาจเกิดขึ้นเป็นพิเศษ

นอกจากความต้องการทางผลิตภัณฑ์ (Product Requirements) ที่ครอบคลุมลักษณะซึ่งถือเป็นการกำหนดคุณสมบัติของระบบที่ได้ ความต้องการของระบบปลอดภัยเชิงวิกฤตยังให้ความสำคัญกับความต้องการทางกระบวนการ (Process Requirements) เช่น การมีเงื่อนไขบังคับให้ต้องทำการวิเคราะห์อันตราย เป็นต้น จะเห็นว่าความต้องการสำหรับระบบปลอดภัยเชิงวิกฤตมีขอบเขตที่กว้างและเจาะจงกว่าระบบทั่วไป ซึ่งมาตรฐาน [21] และคู่มือ [11] ในการพัฒนาระบบปลอดภัยเชิงวิกฤตก็มีการกำหนดรายละเอียดที่เกี่ยวข้องกับความต้องการที่ควรมีความชัดเจนและครอบคลุมทั้งความต้องการทางผลิตภัณฑ์และความต้องการทางกระบวนการ

ตามมาตรฐานไออีซีหมายเลข 61508 (IEC61508) [21] ซึ่งเป็นมาตรฐานอุตสาหกรรมในการพัฒนาระบบปลอดภัยเชิงวิกฤต จะให้ความสำคัญทั้งส่วนความต้องการในระดับซอฟต์แวร์และความต้องการในระดับระบบ และมองว่าทั้ง 2 ส่วนเกี่ยวข้องสัมพันธ์กันอย่างใกล้ชิด และการเก็บความต้องการ

สำหรับระบบปลอดภัยเชิงวิกฤตมักจะเริ่มต้นด้วยการทำการวิเคราะห์อันตรายเบื้องต้น (Preliminary Hazard Analysis: PHA) [11, 12] ซึ่งเป็นการวิเคราะห์ในระดับระบบ เพื่อใช้เป็นส่วนในการวิเคราะห์ความต้องการและตั้งเป็นข้อกำหนดสำหรับตัวซอฟต์แวร์ที่พัฒนา

การแบ่งประเภทของความต้องการที่เกี่ยวข้องกับความปลอดภัย ได้อาศัยการแบ่งประเภทตามแนวทางของ Donald Firesmith [24] ตามรูปที่ 2.2 ซึ่งเป็นการแบ่งประเภทความต้องการที่เกี่ยวข้องกับความปลอดภัย ประเภทของความต้องการสามารถแบ่งได้เป็น 4 ประเภท ดังนี้



รูปที่ 2.2 ความต้องการที่เกี่ยวข้องกับความปลอดภัย

2.1.4.1 ความต้องการความปลอดภัย (Safety Requirements)

ความต้องการความปลอดภัย คือ ความต้องการที่เจาะจงไปที่คุณสมบัติของระบบ โดยเป็นความต้องการที่กำหนดว่าระบบจะต้องปลอดภัยอย่างไร มีการป้องกันอันตรายที่เกิดขึ้นได้อย่างไร เช่น ระบบรถไฟต้องป้องกันไม่ให้รถไฟวิ่งในระดับความเร็วที่เป็นอันตราย เป็นต้น โดยความต้องการนี้พิจารณาไปยังผลลัพธ์ที่ได้ ถือได้ว่าเป็นความต้องการที่ไม่ใช้หน้าที่การทำงาน เนื่องจากไม่ได้เป็นการระบุหน้าที่ของระบบโดยตรง และจัดได้ว่าเป็นความต้องการเพื่อความวางใจ (Dependability Requirements) [25]

ความต้องการความปลอดภัยสามารถแบ่งตามลักษณะของความต้องการได้เป็น 3 ประเภท ดังนี้

- 1) การป้องกัน (Protection) คือ ความต้องการให้ระบบสามารถป้องกันความผิดพลาดได้ เช่น ระบบรถไฟต้องป้องกันไม่ให้เกิดไฟฟ้าดับเกิน 30 นาที เป็นต้น
- 2) การตรวจหา (Detection) คือ ความต้องการในการตรวจพบความผิดพลาดที่เป็นความผิดพลาดที่คาดการณ์หรือจดจำได้ เช่น ระบบรถไฟมีความเป็นไปได้ที่จะเกิดปัญหาไฟฟ้าดับ และเมื่อเกิดเหตุการณ์นี้ขึ้นระบบจะต้องตรวจพบและสามารถดำเนินแก้ไขได้ทันเวลาที่ เป็นต้น
- 3) การตอบสนอง (Reaction) คือ การรายงานความผิดพลาดที่เกิดขึ้น เป็นการเตือนหรือส่งผลออกมา เช่น เมื่อไฟฟ้าดับรถไฟจะมีการแจ้งให้ผู้โดยสารทราบ และระบบช่วยเหลือจะดำเนินการโดยอัตโนมัติ เป็นต้น

จะเห็นว่าลักษณะการแบ่งความต้องการข้างต้น จะพิจารณาไปที่วิธีการที่จะทำให้เกิดความเชื่อมั่นในความปลอดภัยของระบบ และจากรูปที่ 2.2 นอกจากการแบ่งตามลักษณะการทำให้เกิดความเชื่อมั่น ยังสามารถแบ่งประเภทตามลักษณะมูลเหตุที่จำเป็นต้องทำการกำหนดความต้องการด้วย โดยความต้องการในส่วนนี้จัดได้ว่าเป็นความต้องการที่ไม่ใช่หน้าที่

2.1.4.2 ความต้องการนัยปลอดภัย (Safety-Significant Requirements)

ความต้องการนัยปลอดภัย คือ การประเมินความต้องการจากอันตรายที่เกิดขึ้น ซึ่งได้จากการวิเคราะห์อันตราย ซึ่งจะทำการวิเคราะห์อันตรายที่เกิดขึ้นทั้งในส่วนของ ซอฟต์แวร์ ฮาร์ดแวร์ และการปฏิบัติงาน โดยอีกแง่หนึ่งจะมีการพิจารณาตามระดับความปลอดภัยการรวม (Safety Integrity Level: SIL) [26] ที่เกี่ยวข้องกับระบบงานย่อยที่รวมกันเป็นระบบ

ความต้องการนัยปลอดภัยจะเน้นไปที่การวิเคราะห์ความต้องการ สำหรับระบบปลอดภัยเชิงวิกฤตเป็นหลัก หรือที่เรียกว่า การวิเคราะห์ความต้องการความปลอดภัย (Safety Requirements Analysis) โดยจะพิจารณาไปยังกลุ่มความต้องการอื่นๆ ที่ไม่ใช่กลุ่มความต้องการความปลอดภัย เช่น ความต้องการที่เป็นหน้าที่ทั่วไปของระบบ ความต้องการด้านข้อมูล ความต้องการในส่วนต่อประสาน เป็นต้น โดยความต้องการแต่ละตัวอาจมีเหตุการณ์ที่ทำให้เกิดความผิดพลาดของการทำงาน ส่งผลที่เป็นความเสี่ยง หรือแม้แต่อุบัติเหตุ ทำให้ต้องมีการวิเคราะห์ปัจจัยเหล่านี้ประกอบ โดยสามารถแจกแจงนัยปลอดภัยได้ ดังนี้

1) นัยปลอดภัยในซอฟต์แวร์

นัยปลอดภัยในซอฟต์แวร์จะวิเคราะห์อันตรายหรือผลที่ตามมา จากความน่าจะเป็นของเหตุการณ์ที่เกิดขึ้น ในกรณีที่ซอฟต์แวร์เกิดการทำงานผิดพลาดหรือหยุดทำงาน เช่น นัยที่ซอฟต์แวร์ที่ใช้ในการควบคุมความเร็วของรถไฟที่ศูนย์ควบคุมกลางหยุดทำงาน ทำให้ไม่สามารถควบคุมความเร็วรถไฟที่อยู่ในระบบได้ เป็นต้น

2) นัยปลอดภัยในฮาร์ดแวร์

นัยปลอดภัยในฮาร์ดแวร์จะวิเคราะห์อันตรายหรือผลที่ตามมา จากความน่าจะเป็นของเหตุการณ์ที่เกิดขึ้น ในกรณีที่อุปกรณ์ฮาร์ดแวร์ในระบบทำงานผิดพลาดหรือเสีย เช่น นัยที่อุปกรณ์ที่ใช้ในการควบคุมความเร็วของรถไฟ ไม่สามารถลดหรือเพิ่มความเร็วของรถไฟได้อย่างถูกต้องตามที่กำหนด เป็นต้น

3) นัยปลอดภัยในการดำเนินงาน

นัยปลอดภัยในการดำเนินงานจะวิเคราะห์อันตรายหรือผลที่ตามมา จากความน่าจะเป็นของเหตุการณ์ที่เกิดขึ้น ในกรณีที่ผู้ใช้งานระบบใช้งานผิดหรือไม่ระมัดระวัง เพียงพอจนเกิดความผิดพลาดในการทำงานขึ้น เช่น นัยที่พนักงานตั้งค่าความเร็วรถไฟผิดไปจากที่กำหนด เป็นต้น

4) นัยปลอดภัยการรวม

นัยปลอดภัยการรวมจะวิเคราะห์อันตรายหรือผลที่ตามมา จากความน่าจะเป็นของเหตุการณ์ที่เกิดขึ้น ในการนำระบบมาทำงานร่วมกันทั้งส่วนที่เป็นฮาร์ดแวร์ ซอฟต์แวร์ และการดำเนินงาน ที่ส่งผลจากความผิดพลาดไปยังอันตรายหรือเหตุการณ์เดียวกันหรือคล้ายกัน เช่น นัยที่ระบบควบคุมความเร็วของรถไฟไม่เป็นไปตามกำหนด โดยอาจเกิดจากซอฟต์แวร์ ฮาร์ดแวร์ หรือการดำเนินงาน มีความผิดพลาดเกิดขึ้น

ความต้องการนัยปลอดภัยคือ ความต้องการที่ต้องทำการวิเคราะห์อันตราย จากเหตุการณ์ที่อาจเกิดขึ้นได้ โดยความต้องการทุกตัวสามารถเป็นความต้องการความปลอดภัยได้ขึ้นอยู่กับความร้ายแรงและความน่าจะเป็นที่จะทำให้เกิดอันตรายขึ้น ซึ่งสามารถประเมินอันตรายหรือความเสี่ยงได้ตามตารางที่ 2.1 โดยแบ่งการประเมินออกเป็น 2 ด้าน คือ

ประเมินความรุนแรง โดยแบ่งเป็น 4 ระดับ ดังนี้

- 1) ภัยพิบัติ (Catastrophic)
- 2) ช่วงอันตราย (Critical)
- 3) เล็กน้อย (Marginal)
- 4) ไม่สำคัญ (Negligible)

ประเมินความน่าจะเป็น โดยแบ่งเป็น 5 ระดับ ดังนี้

- 1) เป็นประจำ (Frequent)
- 2) มีความเป็นไปได้ (Probable)
- 3) เป็นบางโอกาส (Occasional)
- 4) นานๆครั้ง (Remote)

5) ไม่น่าเป็นไปได้ (Improbable)

ระดับความเสี่ยงที่เกิดขึ้นนั้นเป็นไปตามความน่าจะเป็นที่จะเกิด และระดับความรุนแรงเมื่อเกิดเหตุการณ์นั้น เมื่อความรุนแรงของเหตุการณ์มีความรุนแรงมาก แต่โอกาสที่เกิดขึ้นมีน้อย จะทำให้ประเมินว่าความเสี่ยงที่จะประสบเหตุอยู่ในเกณฑ์ ที่ปลอดภัย ซึ่งระดับความเสี่ยงจากตารางที่ 2.1 สามารถที่จะกำหนดและจัดระดับของความเสี่ยงเพื่อสามารถจัดการได้อย่างถูกต้องตามตารางที่ 2.2

ตารางที่ 2.1 การประเมินค่าความเสี่ยงอุบัติเหตุ [26]

SEVERITY PROBABILITY	Catastrophic	Critical	Marginal	Negligible
Frequent	1	3	7	13
Probable	5	5	9	16
Occasional	4	6	11	18
Remote	8	10	14	19
Improbable	12	15	17	20

ตารางที่ 2.2 หมวดหมู่การเสี่ยงอุบัติเหตุ [26]

Mishap Risk Assessment Value	Mishap Risk Category	Mishap Risk Acceptance Level
1 – 5	High	Component Acquisition Executive
6 – 9	Serious	Program Executive Officer
10 – 17	Medium	Program Manager
18 – 20	Low	As directed

2.1.4.3 ความต้องการระบบปลอดภัย (Safety System Requirements)

ความต้องการระบบปลอดภัย คือ ความต้องการในระดับระบบ ที่ครอบคลุมไปถึงองค์ประกอบต่างๆที่ทำให้ระบบนั้นสามารถดำเนินการได้ ทั้งในส่วนของฮาร์ดแวร์ ซอฟต์แวร์ และการดำเนินงาน ซึ่งความปลอดภัยในส่วนต่างๆ เหล่านี้ถือเป็นองค์ประกอบสำคัญในการกำหนดระบบ และถือเป็นความจำเป็นขั้นพื้นฐานของตัวระบบปลอดภัยเชิงวิกฤต

ความต้องการระบบปลอดภัยถือได้ว่าเป็นเอกลักษณ์ของระบบปลอดภัยเชิงวิกฤต และมีความสำคัญสูงสุดในการพัฒนาระบบ ความต้องการระบบปลอดภัยไม่ได้กำหนดลักษณะในระดับผลผลิตที่ได้ออกมาเท่านั้น แต่ความต้องการระบบปลอดภัยจะทำการพิจารณาไปถึงกระบวนการที่จะได้ผลผลิตออกมาด้วย ไม่ว่าจะเป็นเรื่องของการกำหนดและควบคุมกระบวนการในการพัฒนา การรับประกันและดูแลระบบหลังจากการพัฒนาระบบเสร็จสิ้น เช่น การพัฒนาซอฟต์แวร์จะต้องทำการประเมินความปลอดภัย ต้องทำการอบรมพนักงานให้สามารถใช้งานระบบได้อย่างปลอดภัยหลังการส่งมอบ เป็นต้น

ขอบเขตของความต้องการระบบปลอดภัยให้ข้อจำกัดในวงกว้าง ตัวอย่างเช่น ความต้องการระบบปลอดภัยของระบบขึ้นบินและลงจอดเครื่องบิน ซึ่งบางส่วนมีการกำหนดว่าเครื่องบินต้องขึ้นหรือลงไม่ออกนอกทางวิ่ง ซึ่งเป็นความต้องการที่เกี่ยวข้องกับส่วนต่างๆ ของระบบ ทั้งซอฟต์แวร์ที่ควบคุมการลงจอด อุปกรณ์ลงจอดของเครื่องบิน ซอฟต์แวร์และระบบนำร่อง สัญญาณไฟที่ทางวิ่ง แม้กระทั่งลักษณะรายละเอียดของทางวิ่งและสภาพแวดล้อมขณะลงจอด

จะเห็นว่าความต้องการหลักที่กำหนดรายละเอียดของระบบ จะครอบคลุมลักษณะการทำงานขององค์ประกอบย่อยๆ โดยไม่ได้ขึ้นกับส่วนใดส่วนหนึ่งหรือส่วนที่สำคัญที่สุดเพียงส่วนเดียว เมื่อพัฒนาหรือสร้างระบบความต้องการจะต้องมีส่วนที่กำหนดรายละเอียดส่วนย่อยเหล่านั้นของแต่ละองค์ประกอบ ให้เป็นไปตามความต้องการหลัก

2.1.4.4 ข้อบังคับความปลอดภัย (Safety Constraints)

ข้อบังคับความปลอดภัย คือ กฎเกณฑ์หรือข้อจำกัดต่างๆ ของระบบที่มีจุดประสงค์เพื่อให้เกิดความปลอดภัย โดยมาจากมาตรฐานสำหรับระบบกลุ่มนั้น หรือเป็นลักษณะพื้นฐานของระบบที่ต้องนำมาพิจารณา หรือข้อบังคับที่ถูกวางเอาไว้ตามแนวทางที่ถือปฏิบัติกัน

ข้อบังคับความปลอดภัยถือได้ว่าเป็นลักษณะหนึ่งของความต้องการ เนื่องจากข้อบังคับความปลอดภัยเป็นความต้องการพื้นฐานที่กำหนดขึ้นเพื่อให้มีการทำตามอย่างเคร่งครัด ซึ่งอาจมาจากความต้องการความปลอดภัยหรือความต้องการระบบปลอดภัยที่กำหนดไว้ชัดเจนและมีความสำคัญจนต้องถือเป็นข้อกำหนดที่ต้องทำตาม

ลักษณะของข้อบังคับความปลอดภัยสามารถแจกแจงจากที่มาได้ ดังนี้

1) ข้อบังคับความปลอดภัยจากมาตรฐานความปลอดภัย

ข้อบังคับกลุ่มนี้คือข้อบังคับที่กำหนดขึ้นมาในลักษณะของมาตรฐาน หรือกฎเกณฑ์ควบคุมให้ปฏิบัติตาม เช่น การกำหนดระยะเบรกของรถไฟในระบบอาณัติสัญญาณและการสื่อสารของระบบรถไฟ ข้อกำหนดเหล่านี้มักจะมาจากกลุ่มองค์กรกลาง

หรือหน่วยงานจัดทำและกำหนดมาตรฐาน ซึ่งในหลายกรณีข้อกำหนดเหล่านี้จะถูกนำมาใช้ในการพัฒนาระบบอย่างเคร่งครัด และเป็นเงื่อนไขหลักในการพัฒนาด้วย

2) ข้อบังคับความปลอดภัยจากการกำหนดขึ้นมาในระบบหรือโดยองค์กร

เป็นข้อกำหนดที่เกิดขึ้นในการพัฒนาระบบจากทีมพัฒนาหรือองค์กร ซึ่งกำหนดขึ้นมาเพื่อใช้ในการทำงานในกลุ่ม อาจเป็นกรณีเฉพาะ หรือเป็นมาตรฐานขององค์กร ซึ่งจะมีความแตกต่างกันขึ้นอยู่กับการกำหนดของแต่ละโครงการ โดยอาจสร้างหรือพัฒนาข้อบังคับมาจากมาตรฐานอีกทีหนึ่ง อาจอยู่ในลักษณะแนวทางที่ควรปฏิบัติหรือข้อบังคับที่ให้ทีมพัฒนาต้องปฏิบัติก็ได้ เช่น ในองค์กรมีการกำหนดมาตรฐานซอฟต์แวร์ของตนว่า ต้องมีความผิดพลาดในโค้ดน้อยกว่า 95 เปอร์เซ็นต์

3) ข้อบังคับความปลอดภัยที่เป็นวิธีการปฏิบัติที่ใช้กันทั่วไป

เป็นข้อกำหนดที่เกิดขึ้นโดยไม่มีหลักการหรือเป็นพฤติกรรมขององค์กรเหมือนใน 2 กรณีแรก โดยข้อบังคับลักษณะนี้เกิดจากความเข้าใจของทีมพัฒนาระบบที่ได้จากการพิจารณาหลักการหรือสิ่งที่พึงปฏิบัติโดยทั่วไปที่มักนำมาใช้ในการกำหนดระบบ โดยอาจไม่มีความจำเป็นโดยตรงหรือมีความสำคัญสูงสุดในการพัฒนา แต่จะเป็นตัวช่วยให้เกิดผลลัพธ์ที่ดีขึ้นหากกำหนดเป็นข้อจำกัดในการพัฒนาระบบ เช่น ในระบบควบคุมจราจรทางอากาศโดยส่วนใหญ่จะกำหนดให้มีการทำแบบจำลอง(Simulation) ในการขึ้นบินและลงจอดเครื่องบิน

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 Requirements Patterns for Embedded Systems โดย S. Konrad และ Betty H.C. Cheng [6]

งานวิจัยนี้นำเสนอแบบรูปความต้องการสำหรับระบบฝังตัว โดยอาศัยโครงสร้างตามงานของ E.Gamma โดยจัดองค์ประกอบของแบบรูปในส่วนของการทำให้เกิดผล และ ตัวอย่างโปรแกรมออกไป และเพิ่มข้อจำกัด (Constraints) พฤติกรรม (Behavior) และ แบบรูปการออกแบบ (Design Patterns) เข้ามาซึ่งนำเสนอโดยใช้ยูเอ็มแอล (Unified Modeling Language: UML) พร้อมกับข้อความอธิบาย

ในงานวิจัยได้ทำการแบ่งประเภทของตัวแบบรูปความต้องการสำหรับระบบฝังตัวออกเป็นกลุ่มระบบทำงานย่อยส่วนต่างๆ และนำเสนอตัวอย่างแบบรูปความต้องการของระบบฝังตัว 2 ตัวอย่างด้วยกันคือ

- 1) ตัวกระตุ้น-ตัวตรวจจับ (Actuator-Sensor) ซึ่งเป็นแบบรูปประเภทโครงสร้าง
- 2) การจัดการผลของความผิดพลาด (Fault Handler) ซึ่งเป็นแบบรูปประเภทพฤติกรรม

ตัวอย่างทั้ง 2 ที่แสดงในงานวิจัยนี้ทำให้เห็นภาพรวมของการนำแบบรูปไปใช้งาน ซึ่งเป็นแนวทางที่นำความต้องการที่ผ่านการวิเคราะห์หรือออกแบบมาใช้ใหม่

จากงานวิจัยนี้ทำให้เห็นแนวทางในการนำเอาองค์ความรู้ที่ได้จากกระบวนการวิศวกรรมความต้องการซอฟต์แวร์สำหรับระบบฝังตัวมาใช้ โดยผ่านรูปแบบของแบบรูป ซึ่งการแบ่งแบบรูปออกเป็นส่วนๆ ทำให้มีความชัดเจนและง่ายในการนำไปใช้ โดยผู้วิจัยเห็นว่าลักษณะการใช้แบบรูปความต้องการที่นำเสนอในงานวิจัยนี้มีแนวคิดที่จะนำมาใช้กับงานวิจัยดังนี้

- 1) แนวทางในการใช้แบบรูปตามประเภทของระบบซึ่งในงานวิจัยนี้เป็นระบบฝังตัวแต่ในงานที่ผู้วิจัยนำเสนอจะเป็นระบบปลอดภัยเชิงวิกฤต
- 2) การจัดกลุ่มแบบรูปตามระบบงานย่อย ช่วยให้การนำแบบรูปไปใช้ทำได้ง่ายขึ้น และเห็นภาพความสัมพันธ์ระหว่างแบบรูป

อย่างไรก็ดีในงานวิจัยนี้นำเสนอแบบรูปที่ปรับปรุงมาจากแบบรูปการออกแบบ และยังขาดส่วนที่จะครอบคลุมกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ทั้งหมด เช่น กระบวนการการเก็บรวบรวมความต้องการ เป็นต้น

2.2.2 Requirements Engineering Pattern Structure โดย Toshihiko Tsumaki [8]

งานวิจัยชิ้นนี้มองว่าแบบรูปเป็นการบันทึกและนำความรู้ประสบการณ์กลับมาใช้ใหม่ โดยพยายามนำเอาแบบรูปไปใช้ในการแก้ปัญหาเกี่ยวกับลักษณะของกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ โดยเฉพาะกับกระบวนการเก็บรวบรวมความต้องการ

งานวิจัยนี้สร้างแบบรูปโดยอาศัยมุมมองใน 2 ส่วน คือ

- 1) โครงสร้างจากด้านใน (Inside Structure) ซึ่งให้ความสำคัญกับปัญหา และมองจากเป้าหมาย (Goal) ไปสู่ผลลัพธ์ โดยการชี้แจงและให้เหตุผล ซึ่งนำเสนอออกมาในรูปแบบของแบบรูป
- 2) โครงสร้างจากด้านนอก (Outside Structure) จะมองไปยังสถานการณ์ที่กำลังเกิดขึ้น ซึ่งเป็นส่วนของพลวัต เป็นการให้คำอธิบายในตัวสถานการณ์เงื่อนไขในการเกิดและเงื่อนไขหลังเกิดเหตุการณ์ขึ้น

โดยงานวิจัยนี้พยายามชี้ให้เห็นว่าสถานการณ์จะนำไปสู่การแก้ปัญหาที่เกิดขึ้นด้วยแบบรูป หรือนำไปสู่สถานการณ์อื่นๆ ที่จะเป็ผลเฉลยให้ได้ ซึ่งอาจมองได้ว่างานวิจัยนี้เป็นการนำเสนอแบบรูปจากการพิจารณาโครงสร้างภายใน และนำเสนอวิธีการและกระบวนการใช้งานการพิจารณาโครงสร้างภายนอก

แนวความคิดในการใช้แบบรูปจากงานวิจัยนี้ ได้ให้แนวทางในการจะประยุกต์ใช้แบบรูปให้ได้ประสิทธิภาพสูงขึ้น โดยนำเสนอวิธีการในการนำไปใช้ที่เป็นระบบและแนวทางที่ยืดหยุ่นอย่างพลวัต ซึ่งมีลักษณะที่จะนำไปประยุกต์ใช้ในงานของผู้วิจัยดังต่อไปนี้

- 1) การหาแบบรูปเพื่อจะนำไปใช้โดยมองจากสถานการณ์ ซึ่งแนวทางนี้ผู้วิจัยเห็นว่าเป็นแนวทางที่เหมาะสมในการนำไปใช้ในการกำหนดแนวทางการใช้งานแบบรูป
- 2) การแยกส่วนของแบบรูปเป็นสองส่วน ซึ่งทำให้การใช้งานยืดหยุ่นและนำกลับมาใช้ใหม่ได้อย่างมีประสิทธิภาพ เพราะเป็นการแยกย่อยองค์ประกอบลงไป โดยอาศัยการอ้างอิงและใช้งานผ่านความสัมพันธ์ขององค์ประกอบ ซึ่งผู้วิจัยได้นำแนวทางนี้มาใช้ในการแบ่งประเภทของแบบรูป ซึ่งนอกจากที่แบ่งแบบรูปตามกลุ่มระบบงานแล้ว ยังแบ่งตามองค์ประกอบและแบบรูปเป็น 2 ประเภท ตามลักษณะการใช้ด้วย คือ แบบรูปโดเมนความต้องการ และแบบรูปกระบวนการความต้องการ

ผู้วิจัยเห็นว่างานวิจัยนี้ยังมีข้อบกพร่องที่สำคัญอยู่ คือ

- 1) มีความยากในการสร้างแบบรูป แม้ว่าจะง่ายต่อการนำไปใช้แต่การสร้างแบบรูปตามที่นำเสนอในงานวิจัยนี้ การสร้างตามโครงสร้างที่กำหนดขึ้นทำได้ยาก
- 2) ตัวแบบรูปได้ให้ความสำคัญกับกระบวนการในการใช้ และความสัมพันธ์ระหว่างโครงสร้างภายในและโครงสร้างภายนอกมากเกินไป ทำให้องค์ประกอบของแบบรูปไม่ชัดเจน และขาดความสมบูรณ์ในการจะบันทึกประสบการณ์และปัญหาที่เกิดขึ้นเพื่อการนำกลับมาใช้ใหม่ที่มีประสิทธิภาพ

2.2.3 Pattern for the RE Process โดย Lars Hagge และ Kathrin Lappe [9]

งานวิจัยนี้นำเสนอแบบรูปความต้องการ ซึ่งสนับสนุนกระบวนการวิศวกรรมความต้องการในโครงการพัฒนาซอฟต์แวร์อื่นได้แก่ การเก็บรวบรวมความต้องการ ข้อกำหนดความต้องการ การประเมินความต้องการ

แบบรูปที่นำเสนอในงานวิจัยนี้พยายามที่จะครอบคลุมกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ โดยแต่ละแบบรูปจะนำเสนอปัญหาและผลลัพธ์ในกระบวนการวิศวกรรมความต้องการ ซึ่งเป็นกระบวนการที่ถูกนำมาใช้หรือเป็นเทคนิคที่ประสบความสำเร็จในการใช้มาแล้ว โดยเป็นผลที่เกิดจากการนำไปใช้กับโครงการพัฒนาซอฟต์แวร์

ในงานวิจัยนี้ได้แสดงแบบรูปพื้นฐานสำหรับกระบวนการวิศวกรรมซอฟต์แวร์ไว้ 4 แบบรูป ดังนี้

- 1) ข้อกำหนดจากภาพสะท้อนโครงสร้างของโครงการ (Specification Mirrors Project Structure)

- 2) การจัดเตรียมแนวทางสำหรับข้อกำหนดจากมุมมองย้อนกลับ (Provide Specification Guideline by Reverse Envisioning)
- 3) บัตรดัชนีความต้องการ (Requirements Index Cards)
- 4) การสร้างหนังสือรับรองการยอมรับ (Generated Approval Certificate)

ในงานวิจัยนี้ไม่ได้แสดงให้เห็นตัวอย่างของการนำแบบรูปไปใช้ แต่ก็ได้ให้แนวทางในการนำมาใช้ในโครงการพัฒนาซอฟต์แวร์ ซึ่งมีดังต่อไปนี้

- 1) องค์ประกอบบางตัวของแบบรูปในงานวิจัยนี้เป็นองค์ประกอบที่เหมาะสมในการนำมาใช้ในแบบรูปความต้องการ ซึ่งได้แก่ จุดประสงค์ (Objective) ขอบเขตของโปรแกรมประยุกต์ (Application Area) และ ประสบการณ์ (Experience)
- 2) แบบรูปที่นำเสนอในงานวิจัยนี้นำเสนอสำหรับกระบวนการและเทคนิคที่ใช้ในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ ซึ่งเป็นแนวทางที่จะนำมาประยุกต์ใช้กับการสร้างแบบรูปกระบวนการความต้องการ

แนวคิดของงานวิจัยนี้พิจารณาไปที่กระบวนการวิศวกรรมความต้องการซอฟต์แวร์เป็นหลัก ไม่ได้ให้ข้อสังเกตในการนำไปใช้กับโดเมนของระบบที่ต่างกันออกไป ตัวแบบรูปจึงเหมือนการนำเสนอแนวทาง เทคนิค และกระบวนการ จากตัวปัญหาของการปฏิบัติเป็นหลัก ไม่ได้ครอบคลุมในองค์ประกอบหรือวัตถุที่จะใช้หรือได้รับจากกระบวนการวิศวกรรมความต้องการซอฟต์แวร์

2.2.4 Pattern-based Reuse of Successful Designs: Usability of Safety-Critical Systems

โดย M. Mahemoff A. Hussey และ L. Johnston [27]

งานวิจัยนี้ให้ความสำคัญกับคุณสมบัติการนำไปใช้ (Usability) ของระบบปลอดภัยเชิงวิกฤต ในการนำกลับมาใช้ใหม่ ซึ่งวิธีการที่งานวิจัยนี้นำเสนอคือการใช้แบบรูป โดยมองไปที่ความปลอดภัยและการใช้งาน (Safety-Usability) ซึ่งพิจารณาในองค์ประกอบที่สำคัญ 5 อย่าง คือ

- 1) ความคงทน (Robustness)
- 2) ประสิทธิภาพในงาน (Task Efficiency)
- 3) การนำกลับมาใช้ใหม่ (Reuse)
- 4) การสื่อสารระหว่างผู้ใช้กับคอมพิวเตอร์ (User-Computer Communication)
- 5) ความยืดหยุ่น (Flexibility)

งานวิจัยนี้แบ่งกลุ่มของแบบรูปออกเป็นกลุ่มต่างซึ่งได้แก่ งานดำเนินการ (Task Execution) งานการจัดการ (Task Management) การควบคุมเครื่องจักร (Machine Control) และ

ข้อมูล (Information) และยังนำเสนอตัวอย่างของแบบรูปและสร้างกรณีศึกษาในการนำแบบรูปไปใช้ในการพัฒนาระบบที่เป็นระบบปลอดภัยเชิงวิกฤต

โดยรวมแล้วงานวิจัยนี้ได้นำเสนอการนำกลับมาใช้ใหม่บนพื้นฐานของแบบรูป และได้เป็นแบบรูปความปลอดภัยและการใช้งาน ซึ่งเป็นแนวทางในการนำแบบรูปมาประยุกต์ใช้และเห็นผลจากการใช้จากกรณีศึกษา โดยสิ่งที่ได้จากงานวิจัยนี้ คือ

- 1) แนวทางการนำแบบรูปมาประยุกต์ใช้กับระบบปลอดภัยเชิงวิกฤต
- 2) องค์ความรู้ในกระบวนการพัฒนาระบบปลอดภัยเชิงวิกฤตที่สามารถนำมาใช้ในกระบวนการนำกลับมาใช้ใหม่ได้

สำหรับแบบรูปที่ได้จากงานวิจัยนี้การนำไปใช้ยังค่อนข้างยาก ตัวแบบรูปนำเสนอในลักษณะของสิ่งที่ต้องทำเมื่อเกิดปัญหานั้นๆ การแก้ปัญหาที่แสดงเป็นลักษณะกล่าวโดยรวมยังขาดรายละเอียดที่ควรจะมีในแต่ละขั้นตอนของการพัฒนาระบบ สิ่งที่แบบรูปนำเสนอจึงเป็นเพียงแนวทางในการพัฒนาระบบเท่านั้น ซึ่งหากนำไปใช้จริงก็จะมีรายละเอียดที่ต้องพยายามแก้ปัญหาเองนอกเหนือที่ระบุไว้ ได้แก่ ปัญหาในการวิเคราะห์ระบบ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

การวิเคราะห์โครงสร้างแบบรูปความต้องการและโครงสร้างแบบรูปความต้องการ สำหรับระบบปลอดภัยเชิงวิกฤต

ในบทนี้จะทำการวิเคราะห์เพื่อหาโครงสร้างแบบรูปความต้องการและโครงสร้างแบบรูปความต้องการสำหรับระบบปลอดภัยเชิงวิกฤตที่เหมาะสม โดยเริ่มจากการศึกษาและทำความเข้าใจการนำกลับมาใช้ใหม่ผ่านปัญหาและแนวทางปฏิบัติในวิศวกรรมความต้องการซอฟต์แวร์ ในบทนี้จะมีการวิเคราะห์ความต้องการที่เกี่ยวข้องกับความปลอดภัยประกอบด้วย ซึ่งท้ายที่สุดจะได้ลักษณะของโครงสร้างแบบรูปความต้องการที่เสนอในวิทยานิพนธ์นี้ โดยจะสรุปโครงสร้างและองค์ประกอบในบทถัดไป

3.1 ลักษณะของปัญหาในวิศวกรรมความต้องการซอฟต์แวร์

3.1.1 กระบวนการและองค์ประกอบในวิศวกรรมความต้องการซอฟต์แวร์

วิศวกรรมความต้องการซอฟต์แวร์ คือ กระบวนการที่สำคัญในการพัฒนาซอฟต์แวร์ ซึ่งเกิดขึ้นในช่วงต้นของการพัฒนาซอฟต์แวร์ โดยทำการเก็บความต้องการของผู้ใช้ (User Requirements) และนำไปใช้เป็นข้อกำหนดความต้องการซอฟต์แวร์ (Software Requirements Specification: SRS) ของซอฟต์แวร์ที่จะพัฒนาขึ้น ซึ่งเกี่ยวข้องกับกลุ่มผู้มีส่วนเกี่ยวข้องกับซอฟต์แวร์ (Stakeholder) 2 กลุ่มหลัก คือ ผู้ใช้ (User) และ ผู้พัฒนา (Developer) โดยทั้ง 2 กลุ่มนั้นมีจุดประสงค์ในด้านวิศวกรรมความต้องการซอฟต์แวร์ที่แตกต่างกัน ดังนี้

3.1.1.1 ผู้ใช้

บุคคลกลุ่มนี้คือกลุ่มที่เป็นผู้ใช้ซอฟต์แวร์ที่จะพัฒนาขึ้น ดังนั้นข้อมูลจากคนในกลุ่มนี้จึงกลายมาเป็นคุณลักษณะและหน้าที่ของซอฟต์แวร์ที่จะพัฒนา โดยในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ได้ให้ความสำคัญกับความต้องการของคนกลุ่มนี้เป็นอย่างมาก ซึ่งจะนำไปใช้สร้างเป็นข้อกำหนดความต้องการซอฟต์แวร์และข้อกำหนดความต้องการระบบ (System Requirements Specification: SyRS)

3.1.1.2 ผู้พัฒนา

กลุ่มผู้พัฒนาจำเป็นต้องเข้าใจลักษณะของซอฟต์แวร์ ความจำเพาะของระบบ และข้อจำกัดต่างๆ ของซอฟต์แวร์ที่จะพัฒนาขึ้น ซึ่งข้อมูลเหล่านี้จะได้มาจากกลุ่มผู้ใช้ผ่านทางกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ โดยอาจอยู่ในรูปของข้อกำหนดซอฟต์แวร์ หรือ

แบบจำลองระบบ (System Model) ซึ่งกลุ่มผู้พัฒนาจะนำข้อมูล และ ความเข้าใจในซอฟต์แวร์ที่จะพัฒนา ไปทำการวิเคราะห์ และออกแบบระบบในกระบวนการพัฒนาซอฟต์แวร์เป็นลำดับต่อไป

จะเห็นได้ว่ากลุ่มผู้มีส่วนเกี่ยวข้องกับซอฟต์แวร์ทั้ง 2 กลุ่มจะมีบทบาทและหน้าที่ที่แตกต่างกัน แต่จะขึ้นอยู่กับโดเมนของระบบ (System Domain) ที่พัฒนาเสมอ เช่น ระบบซื้อขายสินค้าสินค้าผ่านทางอินเทอร์เน็ต กลุ่มผู้ใช้เป็นคนคนละกลุ่มกับระบบการยืมคืนหนังสือของห้องสมุด เป็นต้น ความต้องการของผู้ใช้จะมีลักษณะแตกต่างกัน หรือแม้แต่ในระบบประเภทเดียวกัน นโยบาย (Policy) ขององค์กรก็ยังเป็นตัวกำหนดลักษณะของซอฟต์แวร์ให้แตกต่างกันอีกด้วย ดังนั้นจึงสรุปได้ว่าข้อกำหนดความต้องการจะขึ้นอยู่กับโดเมนของระบบที่พัฒนาเป็นหลัก

สำหรับวิศวกรรมความต้องการซอฟต์แวร์มีกระบวนการที่สำคัญด้วยกัน 5 กระบวนการ คือ

- 1) การเก็บรวบรวมความต้องการ
- 2) การวิเคราะห์ความต้องการ
- 3) การจัดทำเอกสารความต้องการ
- 4) การประเมินความต้องการ
- 5) การบริหารความต้องการ

โดยกระบวนการหลักทั้ง 5 ที่กล่าวถึงนั้น ไม่ได้ขึ้นอยู่กับลักษณะหรือโดเมนของระบบ เนื่องจากกระบวนการอยู่ในลักษณะของวิธีการ และขั้นตอนที่จะปฏิบัติเพื่อให้ได้มาซึ่งความต้องการซอฟต์แวร์ ดังนั้นกระบวนการเหล่านี้จึงเป็นส่วนของวิธีการและเทคนิค รวมไปถึงเครื่องมือที่จะนำมาช่วยในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ ไม่ว่าจะเป็นการเก็บความต้องการ โดเมนประเภทใดก็ตาม จะมีแนวทางหลักๆ ที่เหมือนกัน โดยตัวโดเมนจะเป็นตัวตัดสินใจกระบวนการหรือเทคนิคที่เหมาะสมที่จะนำมาใช้ในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ โดยขึ้นอยู่กับประสบการณ์ของผู้พัฒนาเป็นหลัก

ในรูปที่ 3.1 ได้แสดงกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ที่สำคัญทั้ง 5 กระบวนการ โดยมีกระบวนการบริหารความต้องการ เป็นกระบวนการที่เกี่ยวข้องและควบคุมจัดการกระบวนการที่เหลืออยู่ และแสดงให้เห็นว่ากระบวนการวิศวกรรมความต้องการซอฟต์แวร์นั้น เป็นกระบวนการที่มุ่งจะเปลี่ยนความต้องการของผู้ใช้ไปสู่ความเข้าใจของผู้พัฒนา

จะเห็นได้ว่าผู้ที่จะทำงานเกี่ยวกับวิศวกรรมความต้องการซอฟต์แวร์นั้นต้องมีความรู้และประสบการณ์ใน 2 สิ่งด้วยกัน จึงจะสามารถทำงานวิศวกรรมความต้องการซอฟต์แวร์ได้อย่างมีประสิทธิภาพ สิ่งแรกคือ ความรู้เกี่ยวกับตัวระบบที่จะพัฒนาโดยต้องมีความรู้ความเข้าใจในการทำงานระบบงาน ข้อจำกัด กฎเกณฑ์ต่างๆ ซึ่งประสบการณ์จะเข้ามามีส่วนเกี่ยวข้องและมีความจำเป็นอย่างมาก

และยังต้องเข้าใจในแง่ของการพัฒนา เทคโนโลยี และทีมพัฒนาด้วย ซึ่งคนที่ทำงานนี้จะเป็นตัวกลางระหว่างผู้ใช้และผู้พัฒนาซอฟต์แวร์

อีกสิ่งหนึ่งที่ต้องอาศัยความรู้และประสบการณ์ในงานคือ การใช้กระบวนการและเทคนิค ซึ่งขึ้นอยู่กับลักษณะการเลือกและนำไปใช้ว่าจะมีประสิทธิภาพมากน้อยเพียงใด ทำการตัดสินใจใช้เทคนิคหรือกระบวนการใดอย่างไรจึงเหมาะสมกับระบบที่จะพัฒนา



รูปที่ 3.1 กระบวนการวิศวกรรมความต้องการซอฟต์แวร์และผู้มีส่วนเกี่ยวข้อง

3.1.2 ปัญหาด้านโดเมนในวิศวกรรมความต้องการซอฟต์แวร์

ผู้มีส่วนเกี่ยวข้องกับซอฟต์แวร์โดยตรงทั้ง 2 กลุ่ม มีมุมมองสำหรับซอฟต์แวร์ที่พัฒนาแตกต่างกัน โดยเฉพาะโดเมนของระบบที่พัฒนาทั้ง 2 กลุ่มจะมีความคิดและความเข้าใจเป็นของตนเอง ซึ่งปัญหาที่เกิดขึ้นกับวิศวกรรมความต้องการซอฟต์แวร์ในส่วนของโดเมนสามารถศึกษาได้จากปัญหาที่เกิดขึ้นกับกลุ่มบุคคลทั้ง 2 กลุ่ม ตลอดจนส่วนขัดแย้งในมุมมองของทั้ง 2 กลุ่มด้วย โดยสามารถสรุปปัญหาเป็นประเด็นต่างๆ ได้ดังนี้

3.1.2.1 ปัญหาของผู้ใช้

- 1) ผู้ใช้ไม่เปิดเผยความต้องการที่แท้จริง
- 2) ผู้ใช้ยังไม่ทราบสิ่งที่ต้องการหรือไม่ทราบทั้งหมด
- 3) ผู้ใช้ให้ความต้องการที่ไม่ชัดเจน

- 4) ผู้ใช้ตัดสินใจแต่เพียงเฉพาะหน้าไม่ได้คิดอย่างถี่ถ้วน
- 5) ผู้ใช้พยายามเสนอความต้องการ โดยไม่สนใจว่าทำได้จริง
- 6) กลุ่มผู้ใช้ให้ความต้องการที่ไม่ตรงกัน

3.1.2.2 ปัญหาของผู้พัฒนา

- 1) ผู้พัฒนาเข้าใจในความต้องการผิดไป
- 2) ผู้พัฒนาทราบความต้องการ ไม่หมด
- 3) ผู้พัฒนาไม่เข้าใจจุดประสงค์หลักของความต้องการ
- 4) ผู้พัฒนาได้รับความต้องการที่ไม่ถูกต้อง
- 5) ผู้พัฒนาได้รับความต้องการที่ขัดแย้งกัน

3.1.2.3 ปัญหาระหว่างผู้ใช้และผู้พัฒนา

- 1) ผู้ใช้และผู้พัฒนาไม่ได้มีการปรึกษาความต้องการร่วมกัน
- 2) ผู้ใช้และผู้พัฒนามีความเข้าใจไม่ตรงกัน
- 3) ผู้พัฒนาพยายามเปลี่ยนความต้องการของผู้ใช้
- 4) ผู้พัฒนาพยายามเสนอสิ่งที่ไม่ตรงกับความต้องการจริงของผู้ใช้

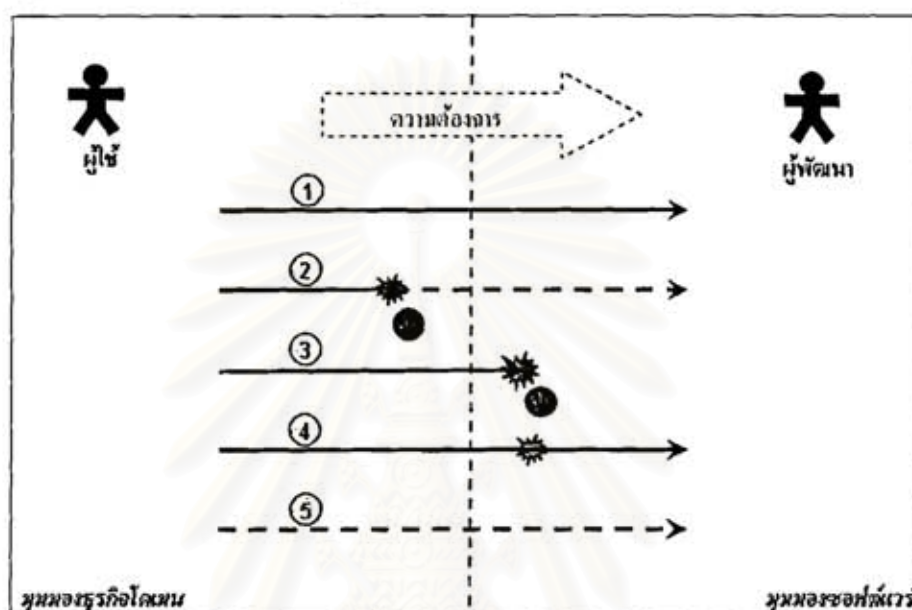
3.1.3 ปัญหาด้านกระบวนการ

ปัญหาที่จะพบในด้านกระบวนการสำหรับวิศวกรรมความต้องการซอฟต์แวร์เสมอ คือ ปัญหาในการนำกระบวนการไปปฏิบัติ ซึ่งจะเห็นว่าปัญหามีการแบ่งแยกชัดเจนในแต่ละกระบวนการ แต่ก็ยังคงมีความสัมพันธ์และส่งผลถึงอีกกระบวนการหนึ่ง เนื่องจากการแก้ปัญหาในกระบวนการใดกระบวนการหนึ่งจะส่งผลในทางบวกกับอีกกระบวนการหนึ่งด้วย แต่ปัญหาที่พบเหล่านี้จะมีลักษณะเป็นไปตามสถานการณ์ที่ได้พบ แม้ว่าจะมีลักษณะของระบบในการพัฒนาที่เหมือนกัน แต่ปัญหาที่เกิดขึ้นก็อาจแตกต่างกันออกไป โดยสามารถสรุปมูลเหตุของปัญหาในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ได้ดังต่อไปนี้

- 1) ไม่มีการนำวิธีการต่าง ๆ มาใช้
- 2) การไม่รู้หรือทราบวิธีการที่จะนำมาใช้ได้
- 3) การใช้วิธีการไม่ถูกต้อง
- 4) การเลือกวิธีการที่ไม่เหมาะสมกับงานหรือจุดประสงค์หลัก
- 5) การเลือกวิธีการที่มีต้นทุนสูงเกินไป
- 6) การเลือกวิธีการที่ส่งผลกระทบต่อส่วนอื่นๆ
- 7) ปัญหาในจุดด้อยของวิธีการหรือส่วนที่วิธีการไม่ครอบคลุม

3.1.4 แบบจำลองของปัญหา

ปัญหาในวิศวกรรมความต้องการซอฟต์แวร์ เมื่อมองในภาพรวมของกลุ่มผู้เกี่ยวข้อง สามารถสรุปการเกิดปัญหาและลักษณะที่เกิดขึ้นได้ดังรูปที่ 3.2 ดังนี้



รูปที่ 3.2 แบบจำลองแสดงปัญหาในวิศวกรรมวิศวกรรมความต้องการซอฟต์แวร์

จากรูปที่ 3.2 แสดงการให้ความต้องการจากผู้ใช้ซึ่งอยู่ในมุมมองโดเมนธุรกิจ (Business Domain View) ที่ตนอยู่ และส่งความต้องการผ่านไปยังผู้พัฒนาที่อยู่ในมุมมองของซอฟต์แวร์ (Software View) โดยมีลักษณะปัญหาที่เกิดขึ้นเป็นไปตามมุมมอง 2 ด้านที่แตกต่างกันดังรูป ซึ่งแบ่งแยกได้เป็นปัญหากลุ่ม ก และ ปัญหากลุ่ม ข ซึ่งมีรายละเอียดดังนี้

1) ปัญหาจากมุมมองของโดเมนธุรกิจ

เป็นปัญหาที่มาจากข้อจำกัดขององค์กร ธุรกิจ ความต้องการเบื้องต้น ซึ่งปัญหานี้เกิดขึ้นกับบุคคลหรือองค์กรที่เป็นผู้ให้ความต้องการ มักจะเป็นข้อกำหนด ความต้องการที่ไม่ชัดเจน หรือความขัดแย้งกันในการความต้องการที่เป็นต้นเหตุของปัญหา

2) ปัญหาจากมุมมองของซอฟต์แวร์

เป็นปัญหาในมุมมองของการพัฒนาซอฟต์แวร์ ซึ่งเป็นปัญหาที่เกี่ยวข้องกับทางผู้พัฒนา และกระบวนการพัฒนาซอฟต์แวร์ เช่น ข้อจำกัดของเทคโนโลยีที่นำมาพัฒนา ความเข้าใจของผู้พัฒนา การแปลความของตัวความต้องการ เป็นต้น

นอกจากปัญหาที่นำเสนอข้างต้น ในรูปที่ 3.2 ได้แสดงความต้องการที่ส่งจากผู้ใช้ไปยังผู้พัฒนาซึ่งมีรูปแบบต่างๆที่สำคัญ และแสดงความเกี่ยวข้องกับปัญหาที่เกิดขึ้นได้ ซึ่งถูกสรุปในรูปแบบแสดงลักษณะการ

ส่งผ่านความต้องการจากผู้ใช้ซึ่งเป็นบุคคลในมุมมองธุรกิจโดเมนไปสู่ผู้พัฒนาซึ่งมีมุมมองของซอฟต์แวร์ โดยเส้นประแสดงให้เห็นความต้องการที่ไม่ถูกต้องหรือเบี่ยงเบนไปจากความต้องการที่แท้จริง ซึ่งอธิบายถูกสรุปต่างๆ ที่ปรากฏในรูปได้ดังนี้

เส้นที่ 1 ความต้องการที่ส่งไปยังผู้พัฒนาโดยไม่มีปัญหาเกิดขึ้น

เป็นความต้องการที่ออกมาจากผู้ใช้ที่มีความชัดเจนถูกต้อง ผ่านกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ไปสู่ผู้พัฒนาโดยไม่มีปัญหาเกิดขึ้นกับตัวความต้องการ และผู้พัฒนาก็มีความเข้าใจและนำไปใช้ได้อย่างถูกต้องโดยไม่มีปัญหาของการพัฒนาที่จะทำให้ความต้องการนั้นเป็นไปไม่ได้หรือต้องมีการเปลี่ยนแปลง

เส้นที่ 2 ความต้องการจากผู้ใช้มีการเบี่ยงเบนหรือผิดพลาดไปเนื่องจากปัญหาที่เกิดขึ้น

ความต้องการที่ผู้ใช้ส่งไปยังผู้พัฒนาผ่านกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ อาจมีความผิดพลาดเกิดขึ้น และส่งผลให้ความต้องการนั้นมีการเปลี่ยนแปลงไปจากเมื่อเริ่มต้น ซึ่งสุดท้ายการพัฒนาซอฟต์แวร์จากการนำความต้องการส่วนนี้ไปใช้ ก็จะทำให้ได้ผลลัพธ์ที่ไม่เป็นไปตามที่ผู้ใช้ต้องการ ปัญหาที่เกิดขึ้นอาจเป็นความผิดพลาดที่เกิดขึ้นเพียงเล็กน้อย แต่ก็ทำให้ระบบเปลี่ยนแปลงไปจากที่ควรจะเป็น และอาจร้ายแรงถึงกับไม่สามารถยอมรับการทำงานที่ไม่ตรงกับความต้องการเมื่อเริ่มแรกได้

เส้นที่ 3 ความต้องการที่เกิดขึ้นไม่ได้เนื่องจากไม่สามารถหลีกเลี่ยงปัญหาได้

ในบางปัญหาที่เกิดขึ้นในวิศวกรรมความต้องการซอฟต์แวร์ อาจส่งผลให้ต้องยุติความต้องการในลักษณะนั้นไป เช่น เป็นความต้องการที่ไม่สามารถนำไปพัฒนาให้เกิดขึ้นได้จริง เป็นความต้องการที่มีความขัดแย้งกันอย่างรุนแรง เป็นต้น ซึ่งปัญหาเหล่านี้เมื่อพบก็จะทำให้ต้องคัดความต้องการส่วนนั้นออกไปซึ่งส่วนใหญ่ปัญหาเหล่านี้เกิดจาก ข้อกำหนดและกฎเกณฑ์ขององค์กร ขีดจำกัดทางด้านเทคโนโลยีในการพัฒนา และความคุ้มค่าในการพัฒนาตามความต้องการนั้น

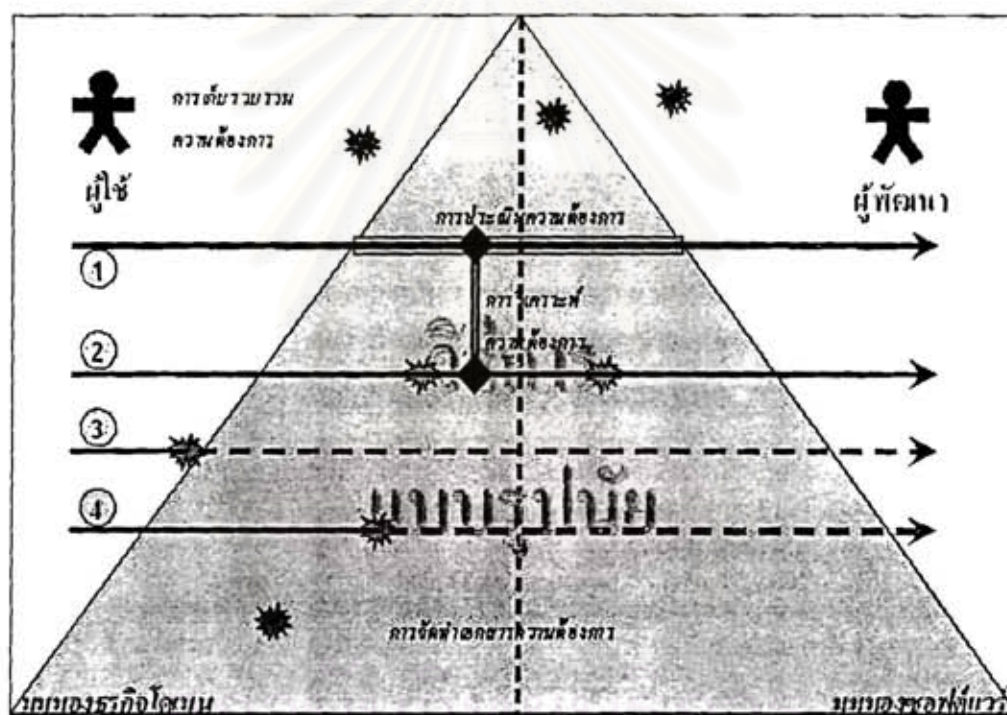
เส้นที่ 4 ความต้องการที่นำไปพัฒนาต่อได้โดยปกติแม้จะต้องพบกับปัญหา

สำหรับบางปัญหาที่เกิดขึ้นอาจไม่ส่งผลร้ายแรงต่อความต้องการจากผู้ใช้ หรืออาจมีผลลัพธ์อื่นๆ ที่ยอมรับได้เท่าเทียมกับความต้องการที่กำหนดมาแต่แรก ซึ่งปัญหาเหล่านี้แก้ไขให้ผ่านไปได้ การพัฒนาตามความต้องการนี้อาจเปลี่ยนไปแต่ให้ผลที่ตอบสนองความต้องการเมื่อแรกเริ่มของผู้ใช้ได้อย่างสมบูรณ์

เส้นที่ 5 ความต้องการที่ผิดพลาดที่เริ่มแต่ต้นจากทางผู้ใช้อยู่แล้ว

ลักษณะความต้องการประเภทนี้มีความผิดพลาดของการให้ความต้องการตั้งแต่เมื่อเริ่มต้นการพัฒนาตามความต้องการแม้จะไม่มีปัญหาเกิดขึ้น แต่สุดท้ายก็ไม่สามารถยอมรับผลลัพธ์ในการพัฒนาจากความต้องการเหล่านี้ได้

สำหรับปัญหาในมุมมองของกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ สามารถนำเสนอได้ดังรูปที่ 3.3 ซึ่งจะเห็นความสำคัญและความเกี่ยวข้องของกระบวนการต่างๆของวิศวกรรมความต้องการซอฟต์แวร์กับตัวความต้องการ



รูปที่ 3.3 แบบจำลองที่แสดงกระบวนการวิศวกรรมความต้องการซอฟต์แวร์

ในรูปที่ 3.3 ได้แสดงกระบวนการหลักในวิศวกรรมความต้องการซอฟต์แวร์ 4 อย่าง ซึ่งแต่ละส่วนจะมีหน้าที่และลักษณะปัญหาที่พบแตกต่างกัน โดยสามารถสรุปได้ดังนี้

1) การเก็บรวบรวมความต้องการ

เป็นกระบวนการที่เกี่ยวข้องกับผู้ใช้โดยตรง โดยจัดการกับความต้องการที่เกิดจากมุมมองธุรกิจโดเมนในการสืบค้นความต้องการที่แท้จริงจากผู้ใช้ ดังนั้นปัญหาที่พบในส่วนนี้จึงเป็นปัญหาของการปฏิบัติและเทคนิควิธีเพื่อให้ได้ความต้องการจากผู้ใช้ ซึ่งถือได้ว่าเป็นขั้นตอนที่ไม่มีลักษณะตายตัวในการปฏิบัติ กระบวนการเก็บรวบรวมความต้องการที่เหมาะสมจะขึ้นอยู่กับลักษณะของผู้ใช้ การเลือกเทคนิคมาปฏิบัติ และความเชี่ยวชาญของผู้เก็บข้อมูล

2) การจัดทำเอกสารความต้องการ

การจัดทำเอกสารความต้องการถือได้ว่าเป็นการรวบรวมความต้องการให้อยู่ในรูปแบบรูปนัย (Formal) มากขึ้น เพื่อให้ง่ายต่อการนำไปใช้ การจัดการ หรือการตรวจสอบ ซึ่งจะเห็นได้ว่าในรูปที่ 3.3 มีส่วนสามเหลี่ยมที่เป็นบริเวณของวิธีการแบบรูปนัย ซึ่งนำมาใช้ในวิศวกรรมความต้องการซอฟต์แวร์ โดยการจัดทำเอกสารความต้องการเกิดขึ้นในส่วนนี้เอง โดยการทำเอกสารความต้องการจะครอบคลุมทั้งมุมมองธุรกิจโดเมนและมุมมองซอฟต์แวร์

3) การวิเคราะห์ความต้องการ

ความต้องการที่ผู้ใช้ส่งไปยังผู้พัฒนามีหลายๆส่วนที่ทำให้เกิดปัญหาขึ้น ไม่ว่าจะเป็นความขัดแย้งในตัวความต้องการ การให้ความสำคัญ ความเกี่ยวข้องกับปัจจัยอื่นๆทั้งภายในและภายนอกของการพัฒนา ซึ่งมองได้ว่าเป็นกระบวนการในการตรวจสอบความต้องการผ่านเอกสารความต้องการซอฟต์แวร์อาศัยวิธีการแบบรูปนัย หรือกึ่งรูปนัย เป็นเครื่องมือในการวิเคราะห์ ดังรูปที่ 3.3 กระบวนการวิเคราะห์ความต้องการเป็นการมุ่งประเด็นไปที่การตรวจสอบในแต่ละส่วนของความต้องการ ซึ่งอาจเกิดขึ้นในช่วงใดช่วงหนึ่ง และเกิดขึ้นหลายครั้งกับความต้องการเดียวกัน

ปัญหาที่พบในกระบวนการวิเคราะห์ความต้องการคือการเลือกใช้วิธีที่เหมาะสม และการวิเคราะห์ที่จะครอบคลุมกลุ่มความต้องการได้อย่างสมบูรณ์ การตรวจสอบเพียงความต้องการใดความต้องการหนึ่ง หรือส่วนใดส่วนหนึ่ง อาจไม่เพียงพอในการหาข้อสรุปรวมของความต้องการ

4) การประเมินความต้องการ

ความต้องการที่ได้มาจากผู้ใช้นำก่อนที่จะส่งไปยังผู้พัฒนาจำเป็นต้องผ่านการประเมินเพื่อเป็นการควบคุมและรับรองว่าความต้องการได้มาและนำไปใช้ได้ถูกต้อง โดยกระบวนการในการประเมินความต้องการเป็นส่วนสำคัญที่จะตรวจสอบว่าความต้องการนั้นๆ ยังคงความถูกต้อง และเมื่อผ่านกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ไปแล้ว ความต้องการที่เป็นผลลัพธ์จะนำไปใช้ในการพัฒนาซอฟต์แวร์ต่อไปโดยไม่มีปัญหาสำหรับผู้ใช้

เช่นเดียวกับการวิเคราะห์ความต้องการ การประเมินความต้องการต้องอาศัยข้อมูลที่มีความชัดเจนเพื่อทำการสอบทวนความถูกต้อง ดังนั้นการประเมินความต้องการจึงอาศัยวิธีการแบบรูปนัยในวิศวกรรมความต้องการซอฟต์แวร์มาช่วยในการประเมิน โดยการประเมินจะเปรียบเทียบและอ้างอิงผ่านเอกสารความต้องการที่ทํ้าขึ้นมานอกเหนือไปจากขั้นตอนการประเมินร่วมกันระหว่างผู้พัฒนาและผู้ใช้

3.1.5 สรุปลักษณะของปัญหาในวิศวกรรมความต้องการซอฟต์แวร์

ปัญหาของวิศวกรรมความต้องการซอฟต์แวร์ เมื่อพิจารณาตามแบบจำลองของความ ต้องการข้างต้นจะพบว่าปัญหาที่พบมี 2 ลักษณะ ดังนี้

3.1.5.1 ปัญหาที่มาจากโดเมนของระบบที่พัฒนา

ซึ่งเกิดขึ้นซ้ำๆ และมีลักษณะคล้ายๆกัน และเมื่อโดเมนของระบบเปลี่ยนไปก็จะพบกับปัญหาใหม่ๆเพิ่มขึ้น โดยจะมีองค์ประกอบที่ทำให้เกิดปัญหาที่มีต้นเหตุมายังจากปัจจัยเพียงไม่กี่ปัจจัย และยิ่งโดเมนของระบบมีความคล้ายคลึงกันเพียงใดลักษณะปัญหาที่เกิดจากองค์ประกอบเหล่านี้ก็จะมีลักษณะและพฤติกรรมคล้ายกันเท่านั้น

3.1.5.2 ปัญหาในการปฏิบัติหรือกระบวนการ

ซึ่งเกิดจากการปฏิบัติที่แตกต่างกัน โดยลักษณะปัญหาที่เกิดขึ้นมักจะคล้ายๆกันเมื่อมีการปฏิบัติไปในแนวทางเดียวกัน แต่ก็จะมีปัญหาที่แตกต่างกันออกไปเมื่อปฏิบัติในแนวทางหรือวิธีใหม่ๆ โดยลักษณะปัญหาที่เกิดขึ้นเกิดจากการเลือกปฏิบัติหรือการเลือกใช้กระบวนการจะไม่เกี่ยวข้องกับโดเมนที่พัฒนาโดยตรง

ดังนั้นในการพิจารณาปัญหาเพื่อเปรียบเทียบกับแบบจำลอง จะพบว่าปัญหาส่วนหนึ่งที่มีการใช้วิธีการแบบรูปนัย จะสามารถจัดการได้ง่ายกว่า เนื่องจากมีการบันทึกและเก็บข้อมูลเป็นเอกสารที่สามารถนำมาอ้างอิง ทำให้สามารถพิสูจน์ จัดการ หรือทวนสอบได้ ดังนั้นปัญหาที่เกิดขึ้นในส่วนนี้จึงจัดการได้ง่าย และมีประสิทธิภาพมากกว่า และยังสามารถใช้เป็นกรณีศึกษาหรือองค์ความรู้เพื่อนำกลับไปใช้ใหม่ด้วย

จากรูปที่ 3.2 และ 3.3 ทำให้เห็นที่มาและลักษณะของปัญหาของวิศวกรรมความต้องการที่เกี่ยวข้องกับโดเมนของระบบ แต่ปัญหาในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์นั้นไม่ได้ปรากฏและอธิบายในรูปโดยตรง แต่เมื่อพิจารณาแล้วกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ก็มีเป้าหมายเพื่อจัดการปัญหาที่เกิดขึ้นกับโดเมนระบบที่พัฒนานั่นเอง ปัญหาที่เกิดขึ้นกับกระบวนการเป็นปัญหาของการนำไปใช้และความเข้าใจในกระบวนการ ดังนั้นการรู้ปัญหาในโดเมนจะทำให้เข้าใจปัญหาของกระบวนการมากขึ้นไปด้วย การเลือกใช้กระบวนการจะทําอย่างมีประสิทธิภาพเนื่องจากทราบปัญหาในโดเมน และเข้าใจกระบวนการมากขึ้นเมื่อมองย้อนไปยังปัญหาที่เกิดขึ้นกับโดเมน

3.2 การนำกลับมาใช้ใหม่ในวิศวกรรมความต้องการซอฟต์แวร์

ในหัวข้อนี้จะทำการวิเคราะห์ลักษณะการนำกลับมาใช้ใหม่ในวิศวกรรมความต้องการซอฟต์แวร์ โดยเริ่มพิจารณาจากแนวทางปฏิบัติและปัญหาที่สรุปได้จากหัวข้อ 3.1 เพื่อทำการวิเคราะห์การนำกลับมา

ใช้ใหม่แยกตามมุมมองด้านโดเมนของความต้องการ และมุมมองด้านกระบวนการของความต้องการ และในส่วนสุดท้ายจะพิจารณาการนำกลับมาใช้ใหม่สำหรับความต้องการที่เกี่ยวข้องกับความปลอดภัย

3.2.1 วิธีการปฏิบัติสำหรับวิศวกรรมความต้องการซอฟต์แวร์

วิธีการปฏิบัติ คือ แนวทางในการปฏิบัติที่ผ่านการนำไปใช้มาก่อนและประสบความสำเร็จมาแล้ว ดังนั้นเมื่อนำเอาวิธีการที่เคยประสบผลและอาจก่อให้เกิดปัญหาน้อยที่สุดมาใช้ จึงทำให้มั่นใจได้ว่าจะประสบผลสำเร็จเหมือนกับการนำไปปฏิบัติก่อนหน้านี้ ซึ่งการนำวิธีปฏิบัติไปใช้ส่วนใหญ่จะให้ความสำคัญกับกระบวนการที่ปฏิบัติ เพื่อหลีกเลี่ยงปัญหาหรือพบปัญหาน้อยที่สุด และเกิดประสิทธิภาพในการปฏิบัติสูงขึ้น

สำหรับวิธีการปฏิบัติที่นำไปใช้ในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ มีวัตถุประสงค์เพื่อให้กระบวนการวิศวกรรมความต้องการที่ปฏิบัติมีความง่ายและเป็นไปแนวทางที่มีประสิทธิภาพ เช่น ทำให้การเก็บรวบรวมความต้องการทำได้ง่ายแต่ยังได้ข้อมูลครบถ้วนและถูกต้อง ซึ่งการใช้วิธีปฏิบัติทำให้เชื่อได้ว่าจะได้ผลลัพธ์ที่ดีกว่าหรือเทียบเท่าเดิม เนื่องจากการนำข้อดีที่ผ่านมามาปรับปรุงใช้อีกในลักษณะที่คล้ายเดิม

อย่างไรก็ดีวิธีปฏิบัติสำหรับวิศวกรรมความต้องการซอฟต์แวร์ไม่ได้รองรับการนำไปใช้งานระบบที่มีลักษณะแตกต่างจากระบบที่ได้นำเสนอ และไม่ได้ให้หลักการในการนำวิธีปฏิบัติไปใช้งาน เช่น วิธีปฏิบัติสำหรับวิศวกรรมความต้องการซอฟต์แวร์ในหนังสือของ I. Sommerville และ P. Sawyer [28] ซึ่งแสดงวิธีปฏิบัติสำหรับกระบวนการต่างๆ ในวิศวกรรมความต้องการซอฟต์แวร์ ถึงแม้ว่าจะมีการให้ข้อสังเกตสำหรับการนำไปใช้ เช่น ค่าใช้จ่าย ความยากง่าย แต่ไม่ได้อธิบายแนวทางในการเลือกวิธีปฏิบัติ ซึ่งผู้นำนำเอาวิธีปฏิบัติเหล่านี้ไปใช้ต้องพิจารณาด้วยตนเองว่าลักษณะระบบของตนเหมาะสมกับวิธีปฏิบัตินั้นๆแค่ไหน

ดังนั้นวิธีปฏิบัติสำหรับวิศวกรรมความต้องการซอฟต์แวร์ จะใช้เป็นแนวทางปฏิบัติให้ได้ความต้องการ โดยพิจารณาจากแง่มุมของการนำกระบวนการและเทคนิคไปใช้ ดังนั้นจุดอ่อนอย่างหนึ่งของวิธีปฏิบัติสำหรับวิศวกรรมความต้องการซอฟต์แวร์คือ การที่ไม่มองไปบนต้นเหตุของปัญหา และเป็นเพียงแนวทางกว้างๆ สำหรับรูปการในหลายๆ ลักษณะ โดยไม่มีการเจาะจงไปในปัญหาที่มีลักษณะพิเศษที่มักจะขึ้นอยู่กับโดเมนของแต่ละระบบ

3.2.2 การนำกลับมาใช้ใหม่สำหรับวิศวกรรมความต้องการซอฟต์แวร์

เมื่อพิจารณาในเรื่องของการนำกลับมาใช้ใหม่สำหรับวิศวกรรมความต้องการซอฟต์แวร์ จะมีลักษณะที่แตกต่างไปจากการนำกลับมาใช้ใหม่ในกระบวนการอื่นๆของการพัฒนาซอฟต์แวร์ เช่น ในกระบวนการการทำให้เกิดผล (Implement) สามารถนำกลับมาใช้ใหม่ในรูปส่วนโปรแกรม (Component)

ที่เป็นซอร์สโค้ด (Source Code) ได้โดยตรง เป็นต้น แต่สำหรับกระบวนการวิศวกรรมความต้องการซอฟต์แวร์จะเป็นการนำกลับมาใช้ใหม่ขององค์ความรู้มากกว่าผลลัพธ์จากระบบเดิม เช่น เอกสารข้อกำหนดความต้องการซอฟต์แวร์จากระบบเดิม การนำกลับมาใช้ใหม่จะไม่นำเอาส่วนของข้อกำหนดความต้องการซอฟต์แวร์นั้นมาใช้ได้โดยตรง แต่จะเป็นการเอาความรู้ความเข้าใจในการได้มาของความต้องการและวิธีการตลอดจนบุคคลและระบบที่จะเข้ามามีส่วนเกี่ยวข้องเพื่อนำมาสร้างเป็นเอกสารข้อกำหนดความต้องการซอฟต์แวร์สำหรับระบบใหม่ที่พัฒนา เป็นต้น ซึ่งสรุปได้ว่าการนำกลับมาใช้ใหม่สำหรับวิศวกรรมความต้องการซอฟต์แวร์เป็นการนำเอาองค์ความรู้กลับมาใช้ใหม่เป็นหลัก

สำหรับวิศวกรรมความต้องการซอฟต์แวร์จะเห็นได้ว่ามีองค์ความรู้ที่สำคัญที่ประกอบไปด้วย 2 ส่วน ส่วนแรกคือองค์ความรู้เกี่ยวกับโดเมนของระบบที่พัฒนา อีกส่วนหนึ่งคือองค์ความรู้ในกระบวนการที่จะนำมาปฏิบัติ ดังนั้นเมื่อมองในแง่ของการนำกลับมาใช้ใหม่ องค์ความรู้ทั้ง 2 ส่วนนี้จึงเป็นส่วนสำคัญในระดับพื้นฐานที่จะนำกลับมาใช้ใหม่ จึงได้แบ่งการนำกลับมาใช้ใหม่ตาม 2 มุมมอง ดังนี้

- 1) มุมมองด้านโดเมนของความต้องการ (Requirements Domain Aspect)
- 2) มุมมองด้านกระบวนการความต้องการ (Requirements Process Aspect)

3.2.3 การนำกลับมาใช้ใหม่ในมุมมองด้านโดเมนของความต้องการ

การนำกลับมาใช้ใหม่ด้านโดเมนความต้องการ คือ การนำกลับมาใช้ใหม่สำหรับวิศวกรรมความต้องการซอฟต์แวร์ที่ขึ้นอยู่กับลักษณะหรือโดเมนของระบบที่พัฒนาซอฟต์แวร์ เช่น ความต้องการที่เป็นหน้าที่การทำงาน หรือ ข้อจำกัดของระบบ เป็นต้น

เมื่อแยกพิจารณาความต้องการซอฟต์แวร์ตามโดเมนของระบบ จะพบว่ากลุ่มผู้ที่เข้ามาเกี่ยวข้องจะมีแนวคิดและลักษณะที่แตกต่างกันออกไป แต่เมื่อพิจารณาว่าเป็นการนำกลับมาใช้ใหม่ทำให้ส่วนที่จะนำกลับมาใช้ใหม่แบ่งออกเป็น 2 ส่วน อย่างชัดเจน เพื่อช่วยให้สามารถตอบสนองกลุ่มผู้เกี่ยวข้องทั้ง 2 กลุ่มได้ แต่ก็ยังมีความเกี่ยวเนื่องและมีส่วนของการนำกลับมาใช้ใหม่ที่เกิดขึ้นร่วมกันอยู่ด้วย โดยสามารถสรุปการนำกลับมาใช้ใหม่เพื่อตอบสนองกลุ่มผู้เกี่ยวข้องทั้ง 2 กลุ่ม ดังนี้

1) ผู้ใช้

ผู้ใช้เป็นผู้ที่เข้าใจในโดเมนมากที่สุดแต่ก็อาจจะไม่สามารถถ่ายทอดและส่งมาถึงผู้พัฒนาได้อีกทั้งผู้ใช้ส่วนใหญ่ก็ไม่เข้าใจในกระบวนการพัฒนาดีพอ ดังนั้นการนำกลับมาใช้ใหม่ในส่วนของผู้ใช้ จะเป็นการนำ ลักษณะของโดเมน ปัญหาของโดเมน ข้อจำกัดและกฎเกณฑ์ ฯลฯ ที่เคยใช้กลับมาใช้ใหม่

เนื่องจากกลุ่มผู้ใช้ส่วนใหญ่มีความเข้าใจในการพัฒนาซอฟต์แวร์ค่อนข้างน้อย ดังนั้นในการนำกลับมาใช้ใหม่จึงพิจารณาจากผู้ใช้ที่รู้เรื่องการพัฒนาซอฟต์แวร์น้อยที่สุด และในทางกลับกันจะมองว่าผู้ใช้จะให้ข้อมูลมากกว่าข้อมูลเดิมที่มีอยู่ นั่นคือการนำกลับมาใช้ใหม่ในส่วนนี้มี

ทั้งการนำเอาองค์ความรู้ที่ผ่านมามาใช้ใหม่ และ การนำเอาโครงสร้างและลักษณะขององค์ความรู้เดิมกลับมาใช้ใหม่

2) ผู้พัฒนา

ในส่วนของผู้พัฒนามักมีปัญหาในการทำความเข้าใจความต้องการ ทำให้เกิดความผิดพลาดในลักษณะต่างๆขึ้น ซึ่งเป็นผลจากระบบที่ได้ไม่ตรงตามความต้องการที่แท้จริงของผู้ใช้ ดังนั้นการนำกลับมาใช้ใหม่ในส่วนของผู้พัฒนาจึงเป็นการนำ ปัญหาของการพัฒนา ความหมาย และนิยามสำหรับผู้พัฒนา ลักษณะของโดเมน สภาพแวดล้อมของระบบ ฯลฯ ซึ่งเป็นส่วนสำคัญของโดเมนของระบบที่ผู้พัฒนาต้องนำกลับมาใช้ใหม่

ผู้พัฒนาที่มีประสบการณ์น้อยโดยเฉพาะกับโดเมนที่กำลังพัฒนา จะเกิดปัญหาในหลายๆจุดของความต้องการที่มีลักษณะเฉพาะเจาะจง ซึ่งบางจุดอาจดูไม่สมเหตุสมผลหรือยากต่อการพัฒนา แต่อาจจะเป็นเช่นนั้นด้วยข้อกำหนดทางธุรกิจ ดังนั้นเพื่อป้องกันความผิดพลาดที่จะเกิดขึ้นจากสิ่งเหล่านี้ การให้รายละเอียดโดยการนำกลับมาใช้ใหม่ในแทบทุกส่วน ซึ่งรวมถึงข้อมูลพื้นฐานทั่วไปเกี่ยวกับโดเมนจึงเป็นสิ่งจำเป็น แม้องค์ความรู้เดิมจะมีส่วนแตกต่างกันออกไปบ้าง แต่ก็จะช่วยให้การนำมาใช้ในวิศวกรรมความต้องการซอฟต์แวร์ครอบคลุมปัญหา และอาจทำให้ได้ผลลัพธ์ใหม่ๆซึ่งดีกว่าเดิม

3.2.4 การนำกลับมาใช้ใหม่ในมุมมองด้านกระบวนการความต้องการ

การนำกลับมาใช้ใหม่ด้านกระบวนการความต้องการ คือ การนำกลับมาใช้ใหม่ของกระบวนการหรือเทคนิคที่ใช้ในวิศวกรรมความต้องการซอฟต์แวร์ ซึ่งมีกระบวนการที่เหมือนกันไม่ว่าระบบจะมีลักษณะอย่างไร สำหรับกระบวนการและเทคนิคในวิศวกรรมความต้องการสามารถแยกตามจุดประสงค์และลักษณะการทำงานหลักได้ตามที่กล่าวถึงแล้วก่อนหน้านี้ โดยสามารถแบ่งได้ดังนี้

1) การเก็บรวบรวมความต้องการ

โดยการเก็บความต้องการจะต้องเข้าใจกระบวนการที่จะทำให้ได้ข้อมูลจากผู้ใช้ เทคนิคและวิธีการต่างๆที่จะนำเข้ามาช่วยในการคัดกรองความต้องการที่แท้จริง การบันทึกและรวบรวมความต้องการ การค้นหาแหล่งข้อมูล และการนำความต้องการดิบมาแปลงเป็นข้อมูลเพื่อใช้ในการพัฒนา

การนำกลับมาใช้ใหม่ในการเก็บรวบรวมความต้องการจะพิจารณาไปที่การเลือกใช้เทคนิคในการเข้าไปเก็บหรือคัดกรองความต้องการ เทคนิคและวิธีการอื่นๆมักจะเป็นส่วนที่ช่วยให้การเก็บความต้องการมีประสิทธิภาพ ทำได้ง่าย และนำไปใช้ได้ดีขึ้น

2) การจัดทำเอกสารความต้องการ

เอกสารความต้องการซอฟต์แวร์ถือได้ว่าเป็นส่วนหนึ่งของซอฟต์แวร์ ซึ่งมีด้วยกันหลายรูปแบบ โดยมีจุดประสงค์เพื่อใช้อธิบายซอฟต์แวร์และเป็นเครื่องมือของผู้พัฒนาเป็นหลัก ดังนั้นกระบวนการและวิธีการที่นำมาใช้จึงขึ้นอยู่กับรูปแบบของกระบวนการพัฒนาซอฟต์แวร์เป็นหลัก ซึ่งผู้เขียนเอกสารจำเป็นต้องมีความเข้าใจในการนำไปใช้ด้วย

การนำกลับมาใช้ใหม่สำหรับการจัดทำเอกสารจึงพิจารณาไปที่รูปแบบต่างๆของเอกสารที่นำมาใช้ วิธีการที่เหมาะสมในการเขียน ตลอดจนเครื่องมือหรือเกณฑ์ในการจัดทำเอกสารว่าควรมีลักษณะอย่างไรจึงจะเหมาะสมกับการพัฒนา

3) การวิเคราะห์ความต้องการ

เทคนิคและกระบวนการในการวิเคราะห์ความต้องการ คือสิ่งที่เกิดขึ้นหลังการเก็บรวบรวมความต้องการมาได้ โดยอาจใช้เครื่องมือเข้ามาช่วยในการเปรียบเทียบความต้องการ หรือนำวิธีการมาช่วยในการหาความสำคัญ และตรวจสอบความถูกต้องกับความต้องการสืบส่วนอื่นๆ

การนำกลับมาใช้ใหม่สำหรับการวิเคราะห์ความต้องการจะมีการพิจารณาไปยังเทคนิคและเครื่องมือสำหรับการวิเคราะห์เปรียบเทียบ เพื่อให้การวิเคราะห์ความต้องการทำได้ง่ายขึ้น โดยเทคนิคต่าง ๆ นั้นจะแจกแจงตามสภาพแวดล้อมเพื่อให้การวิเคราะห์ความต้องการทำได้ง่ายและเหมาะสม

4) การประเมินความต้องการ

การประเมินความต้องการเป็นกระบวนการในการตรวจสอบความต้องการ ว่ามีกระบวนการครบถ้วน โดยให้ความสำคัญไปที่การสรุปความต้องการจากการประเมินร่วมกันระหว่างผู้พัฒนาและผู้ใช้ โดยเลือกใช้เครื่องมือหรือวิธีการตามสภาพแวดล้อมของการพัฒนา หรือกระบวนการที่นำมาใช้ในการพัฒนาซอฟต์แวร์เป็นหลัก

การนำกลับมาใช้ใหม่สำหรับการประเมินความต้องการใช้กระบวนการการพัฒนาซอฟต์แวร์เป็นตัวพิจารณาการนำกลับมาใช้ใหม่ โดยเฉพาะในส่วนของวิธีการจะขึ้นอยู่กับประเภทของซอฟต์แวร์ที่จะพัฒนาด้วย เครื่องมือที่นำมาใช้ก็จะขึ้นกับวิธีการเป็นหลัก

5) การบริหารความต้องการ

สำหรับกระบวนการในการบริหารความต้องการ ถือได้ว่าเป็นกระบวนการที่มีความพิเศษไปจากกระบวนการอื่นๆในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ เนื่องจากกระบวนการบริหารความต้องการซอฟต์แวร์จะเกี่ยวข้องกับการบริหาร โครงการ ตลอดจนเกี่ยวข้องกับกระบวนการอื่นๆในการพัฒนาซอฟต์แวร์ด้วย ดังนั้นการนำกลับมาใช้ใหม่สำหรับการบริหาร

ความต้องการจึงมีลักษณะทางกระบวนการและองค์ความรู้ที่จะนำกลับมาใช้ใหม่ต่างไปจากกระบวนการอื่นๆ

เมื่อพิจารณาการนำกลับมาใช้ใหม่สำหรับกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ องค์ความรู้ที่จะนำมาใช้จึงมีลักษณะเป็นไปเพื่อแสดงให้เห็น กระบวนการ วิธีการ เทคนิค และเครื่องมือ เพื่อการตัดสินใจนำมาใช้ และใช้เพื่อเพิ่มประสิทธิภาพของกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ และเมื่อทำการเรียบเรียงเพื่อกำหนดเป็นกลุ่มขององค์ความรู้ที่จะนำมาใช้

3.2.5 การนำกลับมาใช้ใหม่สำหรับความต้องการที่เกี่ยวข้องกับความปลอดภัย

วิทยานิพนธ์นี้เป็นการพิจารณาไปยังการหาแบบรูปสำหรับความต้องการซอฟต์แวร์ ดังนั้นจึงเป็นการวิเคราะห์ลักษณะความต้องการที่เกี่ยวข้องกับความปลอดภัยในแง่ของซอฟต์แวร์เป็นหลัก ซึ่งในหัวข้อ 3.2.3 และ 3.2.4 ได้วิเคราะห์ลักษณะการนำกลับมาใช้ใหม่สำหรับความต้องการซอฟต์แวร์ในมุมมองของกระบวนการ และโดเมนระบบที่เก็บความต้องการ ในหัวข้อนี้จึงจะวิเคราะห์ลักษณะของความต้องการที่เกี่ยวข้องกับความปลอดภัยสำหรับซอฟต์แวร์ตามที่มีความสัมพันธ์กัน

ความต้องการที่เกี่ยวข้องกับความปลอดภัยมีจุดหมายเพื่อความปลอดภัยไม่ว่าจะเป็นทางตรงหรือทางอ้อม ลักษณะของความต้องการที่เกี่ยวข้องกับความปลอดภัยจึงมีการกำหนดค่อนข้างชัดเจน แต่ขึ้นอยู่กับลักษณะของระบบ ซึ่งในระบบปลอดภัยเชิงวิกฤตที่แตกต่างกันจะมีลักษณะหรือสภาพแวดล้อม การใช้งาน อันตรายที่เกิดขึ้น และความเป็นไปได้ของการเกิดเหตุอันตรายแตกต่างกันไปด้วย ดังนั้นองค์ความรู้ของนำกลับมาใช้ใหม่ของความต้องการที่เกี่ยวข้องกับความปลอดภัย จึงเทียบได้กับองค์ความรู้ในมุมมองโดเมนของความต้องการ โดยเพื่อเติมองค์ความรู้เกี่ยวกับความปลอดภัย เช่น อันตรายที่เกิดขึ้นได้ ลักษณะของอุบัติเหตุ การกำหนดการยอมรับระดับความหายนะ เป็นต้น

สำหรับวิธีการที่นำมาใช้กับกระบวนการความต้องการซอฟต์แวร์สำหรับความต้องการที่เกี่ยวข้องกับความปลอดภัย สามารถใช้กระบวนการสำหรับกระบวนการความต้องการซอฟต์แวร์ของระบบโดยทั่วไปได้ เพียงแต่เป็นการให้ความสำคัญกับส่วนที่เกิดวิกฤตมากเป็นพิเศษ เช่นการวิเคราะห์อันตราย ซึ่งเป็นกระบวนการหนึ่งที่ใช้กับวิศวกรรมความต้องการอยู่แล้ว แต่จะมีความจำเริญกับระบบบางประเภทเท่านั้น และสำหรับระบบปลอดภัยเชิงวิกฤตการวิเคราะห์อันตรายจะถือได้ว่ามีความสำคัญจนขาดไม่ได้

3.3 วิเคราะห์โครงสร้างแบบรูปความต้องการ

3.3.1 ความสัมพันธ์ระหว่างโดเมนความต้องการและกระบวนการความต้องการ

ความสัมพันธ์ระหว่างปัญหาของโดเมนและปัญหาของกระบวนการสามารถพิจารณาได้ 2 แบบ ดังนี้

3.3.1.1 โดเมนความต้องการกำหนดกระบวนการความต้องการ

ปัญหาของโดเมนในแต่ละระบบมีลักษณะแตกต่างกัน เมื่อพิจารณาปัญหาของโดเมนจะเห็นได้ว่ามีความเฉพาะเจาะจงและควรแก้ปัญหานั้นด้วยวิธีการที่ถูกต้องเท่านั้น ซึ่งมองได้ว่ามันคือวิธีการปฏิบัติสำหรับกระบวนการวิศวกรรมความต้องการ ปัญหาในการพัฒนาและสภาพแวดล้อมของระบบเป็นข้อจำกัดที่สำคัญที่ทำให้การนำกระบวนการความต้องการมาใช้ถูกจำกัดไปที่วิธีหรือเทคนิคใดหรือกลุ่มใดกลุ่มหนึ่ง โดยหากพยายามใช้วิธีอื่นนอกจากที่ควรจะเป็นก็อาจเกิดปัญหาความขัดแย้งในกระบวนการ หรือส่งผลให้การทำงานช้าซ้อนและไม่ได้ผลลัพธ์ที่จะนำไปใช้ได้เหมาะสมกับสภาพแวดล้อมที่มีอยู่

ดังนั้นจึงสรุปได้ว่าโดเมนความต้องการจะเป็นตัวกำหนดกระบวนการความต้องการที่จะนำมาใช้ตามลักษณะที่ควรจะเป็นและสอดคล้องกับปัญหาของการพัฒนาและสภาพแวดล้อมของระบบ

3.3.1.2 โดเมนความต้องการชี้นำการใช้กระบวนการความต้องการ

ลักษณะของโดเมนและเป้าหมาย ซึ่งเป็นองค์ประกอบความรู้ของโดเมนความต้องการสามารถนำไปวิเคราะห์เพื่อนำกระบวนการความต้องการที่เหมาะสมมาใช้ได้ โดยหากพิจารณาที่องค์ประกอบความรู้ของกระบวนการความต้องการจะเห็นได้ว่ามี สถานการณ์ จุดประสงค์ และกลุ่มประเภทของระบบ ที่จะสอดคล้องกับการวิเคราะห์ในด้าน โดเมน และพิจารณาหาความสัมพันธ์เพื่อที่จะเลือกกระบวนการและเทคนิคที่เหมาะสมมาตอบสนองเป้าหมายตามที่ลักษณะของโดเมนเป็นตัวกำหนด นั่นคือสามารถพิจารณาจากโดเมนความต้องการเพื่อเป็นตัวชี้นำไปยังการตัดสินใจเลือกกระบวนการความต้องการที่จะนำมาใช้ได้

3.3.2 การแบ่งประเภทแบบรูปความต้องการ

การสร้างแบบรูปความต้องการให้ครอบคลุมและแก้ปัญหาในวิศวกรรมความต้องการซอฟต์แวร์ โดยสร้างแบบรูปเพื่อแก้ปัญหาในลักษณะใดลักษณะหนึ่งเพียงแบบเดียว จะเห็นว่าไม่สามารถตอบสนองความต้องการได้อย่างครบถ้วน จากหลักการในการเขียนแบบรูปและแบบรูปการออกแบบของ Eric Gramar และคณะ [4] จึงได้ข้อสรุปว่าแบบรูปสำหรับวิศวกรรมความต้องการซอฟต์แวร์ควรประกอบด้วยแบบรูปมากกว่าหนึ่งลักษณะรวมเป็นระบบแบบรูป ซึ่งคล้ายกันกับแบบรูปการออกแบบที่มีการแบ่งแบบรูปเป็น การสร้าง (Creational Patterns) โครงสร้าง (Structural Patterns) และพฤติกรรม (Behavioral Patterns) [4]

ในหัวข้อ 3.2 สามารถสรุปลักษณะปัญหาในวิศวกรรมความต้องการได้ 2 ประเภท ที่มีลักษณะแตกต่างกันอย่างชัดเจน คือ กลุ่มที่เป็นปัญหามาจากโดเมน และกลุ่มที่เป็นปัญหามาจากกระบวนการ

ดังนั้นแบบรูปความต้องการจึงประกอบด้วยแบบรูป 2 ประเภท โดยแบบรูปความต้องการทั้ง 2 ลักษณะจะมีองค์ประกอบที่แตกต่างกันตามการใช้งานและความจำเป็น ดังนี้

3.3.2.1 แบบรูปความต้องการที่นำมาใช้แก้ปัญหาโดเมน

แบบรูปในแนวทางนี้นำไปใช้แก้ปัญหาที่เห็นได้จากสภาพของระบบที่สนใจ โดยสมมุติฐานว่า เมื่อมีปัญหาเกิดขึ้นในโดเมนใดแล้ว ทุกๆระบบที่มีโดเมนคล้ายกันน่าจะมีปัญหาในลักษณะเดียวกัน

ผลเฉลยสำหรับปัญหาโดเมนอาจอยู่ในรูปของความต้องการที่นำมาใช้โดยตรงเป็นลักษณะในการเก็บหรือวิเคราะห์ความต้องการเฉพาะสำหรับ โดเมนนั้นที่ต้องทราบ หรือเป็นคำตอบของปัญหาที่ได้จากประสบการณ์ตรง

3.3.2.2 แบบรูปความต้องการที่นำมาใช้แก้ปัญหากระบวนการ

แบบรูปในแนวทางนี้เกิดจากสมมุติฐานที่ว่า กระบวนการที่ใช้สำหรับความต้องการสำหรับงานที่มีจุดประสงค์หรือรูปแบบคล้ายกัน จะสามารถใช้วิธีการในลักษณะเดียวกันเพื่อจัดการกับปัญหาที่เกิดขึ้นได้

แบบรูปในลักษณะนี้จะไม่ได้สนใจกับลักษณะโดเมนของระบบมากนัก เพราะเห็นว่ากระบวนการวิศวกรรมความต้องการมีลักษณะไม่ขึ้นกับลักษณะของระบบประเภทใดประเภทหนึ่งโดยตรง โดยมีวิธีการและเทคนิคที่จะเสริมกระบวนการและสามารถเน้นพิเศษในบางจุดได้ ดังนั้นปัญหาที่เกิดกับกระบวนการจึงเป็นเรื่องของวิธีการปฏิบัติ การเลือกเทคนิคเครื่องมือค่าใช้จ่าย และผลที่ได้รับ

3.3.3 ความสัมพันธ์ในแบบรูปความต้องการ

จากหัวข้อ 3.3.1 จะเห็นความสัมพันธ์ระหว่างโดเมนความต้องการและกระบวนการความต้องการ ซึ่งทำให้แบบรูปที่พิจารณาจากแนวทางทั้ง 2 มีความสัมพันธ์กันทั้งระหว่างแบบรูปประเภทเดียวกันและแบบรูปต่างชนิดกัน โดยแบบรูปความต้องการทั้ง 2 ประเภทมีความสัมพันธ์ระหว่างแบบรูปด้วยกัน ดังนี้

- 1) แบบรูปที่นำเสนอปัญหาโดเมน อาจมีความเกี่ยวข้องและอ้างอิงไปยังปัญหาโดเมนอื่นๆ โดยที่ทุกแบบรูปจะยังคงใช้งานแยกกันอย่างอิสระ
- 2) แบบรูปที่นำเสนอปัญหาโดเมน อาจมีการเสนอแบบรูปสำหรับปัญหากระบวนการที่ควรนำมาใช้ในกระบวนการวิศวกรรมความต้องการของโดเมน
- 3) แบบรูปที่นำเสนอปัญหากระบวนการ จะต้องอธิบายลักษณะของระบบที่ครอบคลุม และจำกัดระบบที่ใช้งานให้เหมาะสม เพื่อการนำมาใช้ที่ชัดเจน ไม่คลุมเครือ โดยมีจุดประสงค์หลักเพียงจุดประสงค์เดียวเพื่อลดความซับซ้อนและความสับสนในการใช้งาน

- 4) แบบรูปที่นำเสนอปัญหากระบวนการ อาจใช้ทดแทนกันได้ ซึ่งขึ้นอยู่กับปัจจัยอื่นๆ ที่ผู้ใช้จะตัดสินใจเอง ซึ่งทำให้ผู้ใช้ต้องสามารถทราบแบบรูปอื่นที่สามารถใช้ทดแทน
- 5) แบบรูปที่นำเสนอปัญหากระบวนการอาจอ้างอิงไปยังวิธีการอื่นๆ หรือวิธีการที่ควรจะนำมาใช้ร่วมกันเพื่อให้เกิดประสิทธิภาพมากขึ้น

3.4 การวิเคราะห์องค์ประกอบและโครงสร้างความสัมพันธ์ในแบบรูปความต้องการ

3.4.1 ลักษณะองค์ประกอบภายในแบบรูปความต้องการ

การนำกลับมาใช้ใหม่ในมุมมองโดเมนของความต้องการและกระบวนการความต้องการมีองค์ประกอบความรู้ที่แตกต่างกัน ทำให้แบบรูปความต้องการที่พิจารณาจากมุมมองทั้ง 2 มุมองค์ประกอบแตกต่างกันไปด้วย อย่างไรก็ตามลักษณะของแบบรูปความต้องการทั้งสองใช้การวิเคราะห์ตามแนวทางแบบรูป และมีการพิจารณาไปยังพื้นฐานขององค์ประกอบแบบเดียวกัน ดังนั้นจึงมีองค์ประกอบในหลายๆ ส่วนที่มีลักษณะเหมือนหรือคล้ายกัน โดยองค์ประกอบในแบบรูปจะมีลักษณะดังนี้

- 1) ลักษณะของปัญหา สภาพแวดล้อม และผลเฉลย เป็นกลุ่มขององค์ประกอบพื้นฐานของแบบรูป ดังนั้นแบบรูปความต้องการทั้ง 2 ประเภทจะมีองค์ประกอบเหล่านี้เป็นส่วนสำคัญในการใช้งาน ซึ่งรายละเอียดขององค์ประกอบส่วนนี้จะให้ความหมาย และกำหนดรายละเอียดที่แตกต่างตามมุมมองที่สร้างขึ้น
- 2) ส่วนสนับสนุนจะมีองค์ประกอบในแบบรูปความต้องการแต่ละประเภทเหมือนกัน แต่ในการนำมาใช้งานจะมีรายละเอียดต่างกันขึ้นอยู่กับประเภทของแบบรูปและองค์ประกอบส่วนหลักของแบบรูปความต้องการ
- 3) แบบรูปจำเป็นต้องมีระบบเรียกชื่อ ซึ่งชื่อที่ใช้ควรมีลักษณะเดียวกัน การตั้งชื่อหรือจัดกลุ่มจะต้องเป็นระบบ และควรแยกประเภทของแบบรูปทั้งสองได้
- 4) ในบางครั้งแบบรูปที่นำมาใช้แก้ปัญหาโดเมนจะอ้างอิงไปยังเทคนิคหรือกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ที่มีความเหมาะสมในการนำมาใช้กับโดเมน ดังนั้นแบบรูปที่นำมาใช้แก้ปัญหาโดเมนจึงมีองค์ประกอบที่ใช้อ้างอิงแบบรูปที่นำมาใช้แก้ปัญหากระบวนการ
- 5) ปัญหาที่เสนอในแบบรูปความต้องการทั้ง 2 มีลักษณะที่แตกต่างกัน แต่มีรูปแบบในการเสนอปัญหาที่คล้ายกัน คือ การอธิบายปัญหาจะพิจารณาปัญหาตามต้นเหตุที่แท้จริง โดยแบบรูปความต้องการสำหรับมุมมองโดเมนจะพิจารณาไปที่เป้าหมายของการทำความเข้าใจความต้องการ และแบบรูปความต้องการสำหรับมุมมองกระบวนการจะพิจารณาไปยังจุดประสงค์ของกระบวนการ

- 6) ลักษณะสภาพแวดล้อมและผลผลิตที่นำเสนอในแบบรูปความต้องการทั้ง 2 มีลักษณะที่แตกต่างกันตามจุดประสงค์หรือเป้าหมายของแบบรูป และเกิดจากการพิจารณาสภาพแวดล้อมในลักษณะที่แตกต่างกัน
- 7) ส่วนสนับสนุนจะมีองค์ประกอบการเข้าใจการใช้และองค์ประกอบตัวอย่างที่เหมือนกันเพื่อใช้ในการแสดงรายละเอียดเพื่อสนับสนุนการใช้งาน
- 8) สภาวะผลลัพธ์เป็นส่วนสนับสนุนที่แสดงผลลัพธ์จากการใช้แบบรูปความต้องการลักษณะองค์ประกอบสภาวะผลลัพธ์จึงขึ้นอยู่กับประเภทของแบบรูป
- 9) การอ้างอิงไปยังแบบรูปที่เกี่ยวข้องสำหรับแบบรูปความต้องการทั้ง 2 ประเภทจะใช้อ้างอิงสำหรับแบบรูปความต้องการประเภทเดียวกันเท่านั้น

ลักษณะโครงสร้างแบบรูปความต้องการควรประกอบด้วยแบบรูปย่อยๆ 2 ชนิด รวมกันเป็นระบบแบบรูป โดยแต่ละประเภทจะมีลักษณะการนำเสนอปัญหา สภาพแวดล้อม และผลผลิต ในมุมมองต่างกัน เพื่อให้เกิคมิติในการแก้ปัญหา และในแบบรูปความต้องการประเภทเดียวกันจะมีโครงสร้างเหมือนกันเพื่อให้ง่ายต่อการใช้และความเข้าใจ

3.4.2 วิเคราะห์องค์ประกอบของแบบรูปสำหรับปัญหาโดเมน

3.4.2.1 ปัญหา

ปัญหาในมุมมองด้านโดเมน คือ ความไม่ชัดเจนในความต้องการและจุดประสงค์ ส่วนหนึ่งอาจเนื่องมาจากขาดความเข้าใจในตัวระบบเก่าและขาดวิสัยทัศน์ต่อระบบใหม่ อีกประเด็นหนึ่งคือไม่ได้มีการตั้งเป้าหมายของระบบที่ต้องการ ซึ่งส่งผลให้ขาดทิศทางของเพื่อให้ได้สิ่งที่ต้องการ องค์ประกอบส่วนของปัญหาในแบบรูปสำหรับปัญหาโดเมนประกอบด้วย เป้าหมาย ปัญหาของโดเมน และปัญหาของการพัฒนา

นอกจากนั้นหากพิจารณาถึงความเป็นไปได้และเทคโนโลยีที่มีอยู่ หากผู้ต้องการระบบยังเข้าใจในขีดความสามารถของเทคโนโลยีไม่ดีพอ ก็ย่อมส่งผลด้านแนวคิดต่อระบบที่จะพัฒนา ไม่ว่าจะเป็นการมองข้ามเทคโนโลยีใหม่ๆ หรือแม้แต่ไม่รับทราบขีดจำกัดของเทคโนโลยีที่จะนำมาใช้ ซึ่งส่งผลร้ายต่อการครองจากผู้พัฒนาหรือการคาดหวังระบบจากผู้ใช้

การเข้าใจปัญหาจะเริ่มจากการเข้าใจปัญหาโดเมนหรือปัญหาที่เป็นต้นเหตุ ถัดมาจึงตรวจสอบเป้าหมายเพื่อสรุปจุดสำคัญในความต้องการ และลำดับสุดท้ายจึงเป็นการเข้าใจปัญหาในแง่ของการพัฒนาและเทคโนโลยีที่อาจนำมาใช้ ซึ่งจะเห็นว่าลักษณะของส่วนอธิบายปัญหาทั้ง 3 มีนัยที่สำคัญและเกี่ยวข้องกัน หากมองในแง่ของบุคคลที่จะเข้ามามีส่วนเกี่ยวข้องจะเห็นว่าปัญหาโดเมนเป็นปัญหาของผู้ใช้ปัจจุบัน เป้าหมายเป็นความต้องการของผู้ใช้ในอนาคด และปัญหาการพัฒนาเป็นปัญหาของผู้พัฒนาระบบ

3.4.2.2 สภาพแวดล้อม

ระบบแต่ละระบบต่างก็มีเอกลักษณ์เฉพาะตัวและขึ้นอยู่กับประเภทของโดเมน โดยมีสภาพแวดล้อมเป็นตัวกำหนดให้แต่ละระบบในโดเมนเดียวกันมีความแตกต่างกัน ซึ่งสภาพแวดล้อมของระบบในโดเมนจะเป็นผลของปัจจัยเกี่ยวข้องที่เกิดขึ้นทั้งจากภายในและภายนอก องค์ประกอบส่วนของสภาพแวดล้อมในแบบรูปสำหรับปัญหาโดเมนประกอบด้วย ลักษณะของโดเมน ความหมายและนิยาม สภาพแวดล้อมของระบบ และข้อจำกัดและกฎเกณฑ์ของระบบ

สภาพแวดล้อมภายในเป็นปัจจัยที่มาจากลักษณะพฤติกรรมของระบบในโดเมน เป็นตัวกำหนดกรอบในการพัฒนา โดยมีสภาพแวดล้อมภายนอกที่เชื่อมโยงไปสู่ปัจจัยอื่นๆที่ไม่สามารถกำหนดและเปลี่ยนแปลงภายใต้กรอบของสภาพแวดล้อมของระบบที่พัฒนา สภาพแวดล้อมของระบบประกอบด้วย กฎเกณฑ์ เงื่อนไข และข้อจำกัดของโดเมน

การอธิบายสภาพแวดล้อมคือ การอธิบายลักษณะโดเมนซึ่งเป็นลักษณะภายใน โดยพิจารณาประกอบกับปัจจัยภายนอกซึ่งก็คือสภาพแวดล้อมของระบบ ภายใต้เงื่อนไขและขอบเขตของข้อจำกัดและกฎเกณฑ์ของระบบ

การอธิบายลักษณะโดเมนจะทำให้เข้าใจเหตุผลทางธุรกิจ (Business Logic) การอธิบายสภาพแวดล้อมของระบบจะเห็นผลกระทบและความเกี่ยวข้องจากภายนอก โดยข้อจำกัดและกฎเกณฑ์เป็นส่วนที่แบ่งแยกให้เกิดความชัดเจนระหว่างสิ่งที่ทำได้และสิ่งที่ทำไม่ได้ สิ่งที่ต้องทำและสิ่งที่ห้ามทำ

3.4.2.3 ผลเฉลย

ผลเฉลยที่เป็นส่วนสำคัญในวิศวกรรมความต้องการซอฟต์แวร์ที่เจาะจงในแต่ละโดเมน หรืออาจเป็นความต้องการซอฟต์แวร์ที่ได้ ซึ่งผลเฉลยที่ผู้ใช้แบบรูปจะได้รับก็คือลักษณะของความต้องการซอฟต์แวร์ รายละเอียดเกี่ยวกับความต้องการซอฟต์แวร์เหล่านั้น ตลอดจนความสัมพันธ์และการจัดลำดับของความต้องการซอฟต์แวร์ องค์ประกอบส่วนของผลเฉลยในแบบรูปสำหรับปัญหาโดเมนประกอบด้วย ความหมายความต้องการและรายละเอียดเกี่ยวกับความต้องการในด้านต่างๆ และการแก้ปัญหา

ผลเฉลยทำให้ทราบความต้องการของผู้ใช้ที่มีต่อโดเมนที่พิจารณาอยู่ ซึ่งคือการแก้ปัญหาในงานวิศวกรรมความต้องการ ผลเฉลยจึงเป็นการสรุปว่าความต้องการจะต้องเป็นอย่างไร เมื่อเกิดปัญหาไม่สามารถหาความต้องการที่เหมาะสมและถูกต้องที่สุดได้จะแก้ไขอย่างไร ซึ่งผลเฉลยจะทำให้ทราบว่าความต้องการคืออะไรบ้าง ความชัดเจนในการใช้ความต้องการเป็นอย่างไร

และใช้ได้เหมาะสมหรือไม่ และมีความเกี่ยวข้องระหว่างกันอย่างไรในแต่ละความต้องการ เพื่อสามารถเลือกใช้ความต้องการบนความสัมพันธ์ที่เหมาะสมได้

ความต้องการสามารถนำมาใช้ได้โดยตรง แต่รายละเอียดปลีกย่อยหรือความไม่ชัดเจนในตัวความต้องการทำให้เราต้องพิจารณาใช้ความต้องการ โดยอาศัยแรงชักจูงเป็นเครื่องมือสนับสนุนการตัดสินใจ แต่ยังมีในบางสถานการณ์ที่ความต้องการอาจมีความขัดแย้งหรือเกี่ยวข้องกันอยู่ซึ่งสามารถทราบได้จากความเชื่อมโยงและข้อจำกัดของความต้องการ

อนึ่งที่มาของความต้องการจากมุมมองต่างๆ จะเป็นผลโดยตรงต่อการกำหนดระดับความสำคัญ และในบางบทบาทอาจส่งผลต่อข้อจำกัดของระบบ ดังนั้น เมื่อได้ผลเฉลยออกมาจึงควรใช้องค์ประกอบผู้เกี่ยวข้องและมุมมองต่างๆ เป็นเครื่องมือในการทวนสอบความต้องการ และยังช่วยจัดการกับความขัดแย้งที่บางครั้งไม่อาจหาข้อสรุปที่ดีกว่าการชอนดามไปยังผู้มีสิทธิ์ขาดในการตัดสินใจด้านนั้นๆ ซึ่งการแก้ปัญหาเหล่านั้นอาจถูกบันทึกผลลัพธ์ในการแก้ปัญหาไปแล้วในรูปแบบของวิธีการเพื่อแก้ปัญหาโดเมน

3.4.2.4 ส่วนสนับสนุน

ส่วนสนับสนุนในการใช้งาน เป็นส่วนเพิ่มเติมเพื่อให้การใช้แบบรูปมีประสิทธิภาพมากขึ้น ซึ่งในแต่ละแบบรูปอาจไม่ต้องมีองค์ประกอบเหล่านี้ก็ได้ ซึ่งโดยส่วนใหญ่จะมีลักษณะเหมือนส่วนสนับสนุนที่ใช้กับแบบรูปประเภทอื่น

องค์ความรู้ที่สำคัญที่ใช้ในการนำกลับมาใช้ใหม่สำหรับส่วนสนับสนุนควรประกอบด้วยสถานะผลลัพธ์ การนำไปใช้ที่ทราบ ตัวอย่าง แบบรูปที่เกี่ยวข้อง และแบบรูปสำหรับปัญหากระบวนการ

3.4.2.5 องค์ประกอบที่อยู่ในแบบรูปความต้องการสำหรับปัญหาโดเมน

องค์ความรู้ที่ควรมีอยู่ในแบบรูปความต้องการสำหรับปัญหาโดเมน มีรายละเอียดและความสำคัญสรุปได้ดังตารางที่ 3.1 ดังนี้

ตารางที่ 3.1 รายละเอียดและความสำคัญสำหรับองค์ความรู้ในมุมมองโดเมนความต้องการ

องค์ความรู้	รายละเอียด	ความสำคัญ
เป้าหมาย	เป็นส่วนกำหนดหน้าที่หลักของระบบ ใช้อธิบายเป้าหมายสูงสุดในการพัฒนาระบบ	ทำให้เข้าใจและเล็งเห็นความสำคัญของระบบมากขึ้น เป็นส่วนที่สามารถอธิบายความต้องการที่แท้จริงของระบบได้

ตารางที่ 3.1 รายละเอียดและความสำคัญสำหรับองค์ความรู้ในมุมมองโดเมนความต้องการ (ต่อ)

องค์ความรู้	รายละเอียด	ความสำคัญ
ปัญหาของโดเมน	เป็นการอธิบายและแสดงปัญหาของโดเมนความต้องการ	เป็นตัวแสดงปัญหาที่อาจจะเกิดขึ้นได้ ซึ่งสามารถนำมาใช้เป็นแนวทางในการกำหนดระบบให้เกิดความชัดเจน และถูกต้อง ช่วยให้เข้าใจรายละเอียดและที่มาของปัญหาโดเมน
ปัญหาของการพัฒนา	เป็นส่วนที่บ่งบอกข้อจำกัดในความเป็นไปได้ของการพัฒนา ซึ่งข้อมูลส่วนนี้อาจเกิดขึ้นเมื่อมีการพัฒนาหรือเก็บข้อมูลไปบ้างแล้ว จึงจะปรากฏ	ทำให้เห็นข้อจำกัดในการพัฒนาและเทคโนโลยีที่จะนำมาใช้โดยเกิดจากลักษณะความต้องการบางลักษณะที่มี การเจาะจงหรือเรียกร้องเป็นพิเศษ
ลักษณะของโดเมน	เป็นส่วนที่อธิบายตัวโดเมน ซึ่งเป็นภาพรวมของระบบ รวมไปถึงการติดต่อกับส่วนอื่นๆ ไม่ว่าจะเป็นซอฟต์แวร์ ฮาร์ดแวร์ หรือผู้ใช้	ทำให้เข้าใจระบบที่จะพัฒนา สามารถนำไปใช้อธิบายภาพรวมของระบบทำให้เข้าใจความเกี่ยวข้องและลักษณะการทำงาน
สภาพแวดล้อมของระบบ	เป็นส่วนที่ใช้อธิบายสภาพทั่วไปที่สอดคล้องและเกี่ยวข้องกับการพัฒนา เช่น กลไกหรือธุรกิจที่เกี่ยวข้องสัมพันธ์ ผู้ที่จะเข้ามามีบทบาทในระบบ บุคคลที่จะเกี่ยวข้องกับการพัฒนา หรือองค์กรที่เข้ามาเกี่ยวข้องด้วย เป็นต้น	การเข้าใจลักษณะสภาพแวดล้อมที่จะเข้ามาเกี่ยวข้อง จะทำให้เห็นลักษณะของปัญหาที่อาจจะเกิดขึ้นได้ ช่วยให้เห็นผลกระทบและความสัมพันธ์กับส่วนต่างๆ ได้ดียิ่งขึ้น
ข้อจำกัดและกฎเกณฑ์	เป็นตัวกำหนดขอบเขตและข้อจำกัดของตัวระบบในการพัฒนา รวมถึงรายละเอียดทางธุรกิจที่ผู้พัฒนาต้องทราบ เช่น ข้อกำหนด มาตรฐาน กฎหมาย เป็นต้น	เป็นส่วนที่ผู้พัฒนาและผู้ใช้จะมองข้ามไปไม่ได้ เพราะเป็นตัวกำหนดเงื่อนไขของลักษณะที่เป็นไปไม่ได้หรือไม่เป็นไปตามกฎข้อตกลงที่จะต้องปฏิบัติตามอย่างเคร่งครัด

ตารางที่ 3.1 รายละเอียดและความสำคัญสำหรับองค์ความรู้ในมุมมองโดเมนความต้องการ (ต่อ)

องค์ความรู้	รายละเอียด	ความสำคัญ
ความหมายและนิยาม	เป็นส่วนที่ใช้อธิบายความหมายสำหรับผู้พัฒนา ซึ่งเป็นการนิยามความหมายในมุมมองของผู้ใช้ให้ชัดเจนและอยู่ในลักษณะที่ผู้พัฒนาจะเข้าใจได้ง่าย	ในบางครั้งก็ยังมีการนิยามคำเรียกใหม่ ๆ กับสิ่งที่เกี่ยวข้อง ซึ่งผู้พัฒนาที่มีความจำเป็นต้องมีความเข้าใจในคำเหล่านั้นและต้องเข้าใจไปในทางเดียวกันกับผู้ใช้ ต้องไม่เกิดเหตุการณ์ที่ผู้ใช้และผู้พัฒนามีความเข้าใจไม่ตรงกัน ซึ่งจะส่งผลให้ผู้พัฒนาเข้าใจในความต้องการผิดไป
ความต้องการ	เป็นสิ่งที่ระบุถึงความต้องการของผู้ใช้ ทั้งความต้องการที่เป็นหน้าที่และความต้องการที่ไม่ใช่หน้าที่ เช่น ความสามารถของระบบ คุณภาพของระบบ การบำรุงรักษาระบบ เป็นต้น	ความต้องการเป็นหัวใจและผลลัพธ์ที่สำคัญในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ เป็นส่วนที่กำหนดหน้าที่ความสามารถโดยตรงของซอฟต์แวร์ที่พัฒนา
การแก้ปัญหา	เป็นส่วนที่แสดงวิธีการแก้ปัญหหรือการดำเนินการในงานวิศวกรรมความต้องการซอฟต์แวร์ที่บรรลุผล	สามารถช่วยในการทำงานได้เป็นอย่างดี การแก้ปัญหสามารถนำมาใช้เป็นจุดสังเกตหรือแนวทางในการแก้ปัญหที่พบ

3.4.3 วิเคราะห์องค์ประกอบสำหรับแบบรูปปัญหากระบวนการ

3.4.3.1 ปัญหา

การจะนำเทคนิคหรือวิธีการใดมาใช้ในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์เป็นเรื่องที่เป็นปัญหาเพราะวิธีการที่นำมาใช้อาจไม่เหมาะสมทั้งยากเกินไปและไม่ได้รับผลตามจุดประสงค์ของการทำวิศวกรรมความต้องการซอฟต์แวร์ และอีกปัญหานึงของกระบวนการคือการไม่มีจุดประสงค์ที่ชัดเจนของการนำเทคนิคหรือวิธีการมาใช้ หรือเป็นการใช้ที่ไม่ตรงกับจุดประสงค์ของมันนั่นเอง

รูปแบบของการพัฒนาและตัวบุคคลก็ถือเป็นปัญหาสำคัญของการตัดสินใจเลือกใช้เทคนิคหรือวิธีการต่างๆที่มีอยู่เป็นจำนวนมาก ซึ่งกล่าวได้ว่าการละเลยประเด็นในเรื่องของวิธีการพัฒนาและทีมพัฒนาเป็นต้นเหตุไปสู่ปัญหาการใช้กระบวนการที่ไม่เหมาะสมกับคนในทีมพัฒนา

เมื่อจะจัดการกับกระบวนการวิศวกรรมความต้องการซอฟต์แวร์จำเป็นต้องเข้าใจจุดประสงค์ของกระบวนการที่ปฏิบัติ เช่น เข้าใจจุดประสงค์ของการตรวจทานความถูกต้องของความต้องการที่ได้มาจากผู้ใช้ เป็นต้น เมื่อมีจุดประสงค์ที่ชัดเจนก็จะนำไปสู่การเลือกใช้เทคนิคและวิธีที่เหมาะสมได้ โดยการพิจารณาความเป็นไปได้ในการนำมาปฏิบัติประกอบการตัดสินใจ ซึ่งคือการพิจารณาปัญหาของวิธีการพัฒนา เมื่อสังเกตจะพบว่าจุดประสงค์ของกระบวนการและวิธีการพัฒนามักเป็นปัญหาที่เกิดขึ้นเสมอ แม้จะมีที่มาจากหลายลักษณะแต่ก็สามารถสรุปได้ว่าเป็นเพราะให้ความสำคัญกับสิ่งเหล่านี้น้อยเกินไป

3.4.3.2 สภาพแวดล้อม

การทำวิศวกรรมความต้องการซอฟต์แวร์อาจมีเทคนิคหรือวิธีที่นำมาใช้ได้หลายวิธี แต่ปัจจัยสำคัญที่เป็นตัวกำหนดว่าจะใช้วิธีใดก็คือสภาพแวดล้อม ซึ่งในมุมมองของกระบวนการสภาพแวดล้อมก็คือสิ่งที่กำหนดพฤติกรรมของการทำงานนั่นเอง องค์ประกอบที่ใช้ในการระบุสภาพแวดล้อมในกระบวนการทำงานประกอบด้วย สถานการณ์ และกลุ่มประเภทของระบบ

สถานการณ์คือสภาพแวดล้อมของการทำงานที่เป็นตัวบิบบันให้เกิดพฤติกรรมที่ต่างกันไป โดยมีกลุ่มประเภทของระบบแสดงลักษณะเฉพาะของระบบที่ส่งผลต่อการกระบวนการทำงาน เมื่อสถานการณ์มีความชัดเจนก็ย่อมจะสามารถกำหนดเทคนิคหรือวิธีที่นำมาใช้ได้ด้วยการพิจารณากลุ่มประเภทระบบที่เหมาะสมประกอบ และหากสภาพแวดล้อมที่ความเปลี่ยนแปลงไม่ชัดเจนหรือไม่สามารถสรุปได้ การเลือกใช้เทคนิควิธีที่เหมาะสมก็จะทำไม่ได้

3.4.3.3 ผลเฉลย

ในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ การเลือกใช้เทคนิคหรือวิธีการที่เหมาะสมให้ได้ผลลัพธ์ที่ดีถือเป็นผลเฉลยที่อยู่ในแบบรูป แต่ไม่เพียงแค่ว่าทราบเทคนิคและวิธีการที่ใช้เท่านั้น การใช้เทคนิคและวิธีการในแง่อื่นๆ ก็เป็นเรื่องสำคัญด้วย และหากพิจารณาให้ครอบคลุมผลเฉลยที่ควรจะได้รับ โดยแบบรูปสำหรับปัญหากระบวนการ ก็ควรบอกในรายละเอียดของเทคนิคและวิธีการที่น่าเสนอ ประสิทธิภาพที่จะเป็นประโยชน์ ค่าใช้จ่ายที่ตามมารวมทั้งประเด็นปลีกย่อยที่ต้องระมัดระวัง

ผลเฉลยสำหรับกระบวนการที่แบบรูปเสนอเป็นลักษณะของการถ่ายทอดเทคนิค วิธีการปฏิบัติ และประสิทธิภาพ ที่ใช้กับวิธีการในแบบรูป โดยมีวิธีการเป็นตัวอธิบายขั้นตอนเพื่อให้สามารถปฏิบัติได้อย่างถูกต้อง และอาศัยเครื่องมือเป็นอุปกรณ์ช่วยให้ทำงานได้มีประสิทธิภาพมากขึ้น สำหรับค่าใช้จ่ายเป็นปัจจัยสำคัญที่จะใช้ตัดสินใจเลือกปฏิบัติตามวิธีการในแบบรูปหรือไม่

ประสบการณ์ถือเป็นส่วนที่ช่วยให้การใช้ผลเฉลยของแบบรูปเป็นไปอย่างสมบูรณ์ เพราะคือส่วนที่สามารถใช้บ่งชี้และสนับสนุนยังจุดซึ่งอาจถูกละเลยได้ง่าย รวมทั้งยังทำให้สามารถตัดสินใจและใช้แบบรูปได้ดียิ่งขึ้น เพราะเห็นจุดสำคัญในการช่วยแก้ปัญหาและจุดสำคัญที่ทำให้การใช้ไม่บรรลุผล

3.4.3.4 ส่วนสนับสนุน

เนื่องจากสภาวะผลลัพธ์ การเข้าใจ และตัวอย่าง มีลักษณะการใช้งานและความหมาย เช่นเดียวกับที่ใช้ในแบบรูปสำหรับปัญหาโดเมน ส่วนแบบรูปที่เกี่ยวข้องในส่วนของแบบรูปสำหรับปัญหากระบวนการจะมีลักษณะต่างออกไป เพราะแบบรูปที่เกี่ยวข้องจะหมายถึงแบบรูปสำหรับปัญหากระบวนการด้วยตนเองที่มีส่วนเกี่ยวข้องกัน เช่น มีลักษณะที่ซ้ำแทนกันได้ หรือเป็นวิธีและเทคนิคที่จะสนับสนุนซึ่งกันและกัน โดยองค์ประกอบเหล่านี้ถือเป็นส่วนที่ช่วยให้การใช้แบบรูปมีความสมบูรณ์สามารถใช้งานได้อย่างมีประสิทธิภาพและมีความเข้าใจในการใช้แบบรูปมากขึ้น

3.4.3.5 องค์ประกอบที่อยู่ในแบบรูปความต้องการสำหรับปัญหากระบวนการ

องค์ความรู้ที่ควรมีอยู่ในแบบรูปความต้องการสำหรับปัญหากระบวนการ มีรายละเอียดและความสำคัญสรุปได้ดังตาราง 3.2 ดังนี้

ตารางที่ 3.2 รายละเอียดและความสำคัญสำหรับองค์ความรู้ในมุมมองกระบวนการความต้องการ

องค์ความรู้	รายละเอียด	ความสำคัญ
จุดประสงค์	เป็นการแสดงให้เห็นจุดประสงค์ของงานส่วนที่ทำอยู่ ทำให้เห็นเป้าหมายและผลลัพธ์ที่คาดหวัง	การมีจุดประสงค์ของงานทำให้เห็นเป้าหมายของกระบวนการชัดเจน ซึ่งจะเป็นส่วนสนับสนุนและตัดสินใจว่าจะนำกระบวนการแบบใดมาใช้
สถานการณ์	เป็นส่วนที่ใช้อธิบายลักษณะของโครงการ หรือเหตุการณ์ที่จำเป็นต้องใช้กระบวนการวิศวกรรมความต้องการซอฟต์แวร์ โดยเป็นจะบ่งบอกสถานะและพฤติกรรมที่จะนำมาพิจารณา	สถานการณ์ต่างๆจะเป็นส่วนสำคัญที่จะใช้เลือกวิธีการที่เหมาะสม และเป็นเหตุผลในการตัดสินใจใช้เทคนิคหรือกระบวนการ

ตารางที่ 3.2 รายละเอียดและความสำคัญสำหรับองค์ความรู้ในมุมมองกระบวนการความต้องการ
(ต่อ)

องค์ความรู้	รายละเอียด	ความสำคัญ
กลุ่มประเภทของระบบ	เป็นคำบอกให้ทราบว่าวิธีการหรือเทคนิคที่นำมาใช้มีความเหมาะสมกับระบบลักษณะใดหรือมีผลสำเร็จมาจากระบบประเภทใด	การทราบลักษณะระบบที่ประสบผลกับวิธีการที่กล่าวถึงจะทำให้สามารถตัดสินใจเลือกวิธีการได้ถูกต้องในบางสถานการณ์อาจมีวิธีการหรือเทคนิคที่นำมาใช้ได้หลายวิธี และยังเป็นการตั้งข้อสังเกตทำให้เห็นว่าวิธีการนั้นๆจะเป็นประโยชน์กับระบบประเภทใดด้วย
ประสบการณ์	เป็นความรู้และประสบการณ์ที่สามารถถ่ายทอดหรือบันทึกปัญหาและการแก้ปัญหาที่เกิดขึ้น	ข้อมูลที่ได้จากประสบการณ์เปรียบเสมือนกรณีศึกษาในการใช้วิธีหรือเทคนิคที่มีอยู่ ซึ่งจะสามารถนำมาเป็นแนวทางในการใช้และปฏิบัติตามวิธีการหรือเทคนิคที่น่าเสนอได้
ขั้นตอนวิธี	เป็นขั้นตอนและวิธีการที่นำมาใช้เป็นการอธิบายการนำไปใช้เป็นลำดับขั้นตอนที่มีความชัดเจนเพื่อจะสามารถนำไปปฏิบัติตามได้โดยง่าย	การมีขั้นตอนที่ชัดเจนทำให้สามารถปฏิบัติตามวิธีการได้อย่างถูกต้องและได้ประสิทธิผลมากกว่า การใช้งานที่ไม่มีขั้นตอนชัดเจนหรือมีลำดับขั้นตอนที่ไม่ถูกต้องจะทำให้การปฏิบัติตามวิธีการหรือเทคนิคทำได้ยากและอาจไม่ได้ผล
ค่าใช้จ่าย	เป็นค่าใช้จ่ายที่จะตามมาจากการใช้วิธีการหรือเทคนิคที่กล่าวถึงซึ่งไม่ได้หมายความว่าความถึงแต่ค่าใช้จ่ายที่เป็นเงินที่เสียไป แต่มีความหมายรวมถึงเวลา ความยากวิธีการ หรือสิ่งที่ต้องเสียไปเมื่อใช้วิธีการหรือเทคนิคนั้น	ค่าใช้จ่ายจะเป็นตัวแปรสำคัญในการเลือกใช้วิธีการหรือเทคนิค ซึ่งสุดท้ายแล้ววิธีการที่ดีสุดอาจไม่ใช่ทางเลือกที่ดีที่สุด แม้จะทำให้การทำงานมีประสิทธิภาพและได้ผลมากกว่า

ตารางที่ 3.2 รายละเอียดและความสำคัญสำหรับองค์ความรู้ในมุมมองกระบวนการความต้องการ
(ต่อ)

องค์ความรู้	รายละเอียด	ความสำคัญ
เครื่องมือ	เป็นการนำเสนอเครื่องมือที่จะสามารถนำมาช่วยในวิธีการที่นำเสนอได้ ซึ่งเครื่องมือมีความหมายตั้งแต่ โปรแกรมประยุกต์ที่นำมาช่วยงานตลอดจนเครื่องมือทางเทคนิคที่จะมาเพิ่มประสิทธิภาพหรือกลไกที่จะทำให้การทำงานดีขึ้น นอกจากนี้เครื่องมือแล้วยังนำเสนอการพิจารณาหรือการตัดสินใจในการเลือกใช้เครื่องมือด้วย	การนำเครื่องมือมาใช้และใช้งานได้อย่างถูกต้องเป็นการเพิ่มประสิทธิภาพของวิธีการในการปฏิบัติ และการเลือกเครื่องมือที่จะนำมาใช้ให้เหมาะสมก็เป็นสิ่งสำคัญอีกอย่าง ซึ่งจะทำให้การปฏิบัติทำได้ดีและได้ผลมากขึ้น

3.4.4 วิเคราะห์องค์ประกอบสำหรับแบบรูปความต้องการที่เกี่ยวข้องกับความปลอดภัย

แบบรูปความต้องการที่เกี่ยวข้องกับความปลอดภัยเป็นแบบรูปในลักษณะเดียวกันกับแบบรูปความต้องการสำหรับปัญหาโดเมน ซึ่งมองได้ว่าเป็นแบบรูปความต้องการสำหรับปัญหาโดเมนที่นำไปใช้กับระบบปลอดภัยเชิงวิกฤต ดังนั้นองค์ประกอบในส่วนนี้จึงเหมือนกันกับองค์ประกอบในแบบรูปความต้องการสำหรับปัญหาโดเมน แต่จะมีองค์ประกอบอื่นเพิ่มเติมตามจุดประสงค์ในด้านความต้องการที่เกี่ยวข้องกับความปลอดภัยด้วย

3.4.4.1 ปัญหา

ส่วนของปัญหาในมุมมองที่เกี่ยวข้องกับความปลอดภัยจะพิจารณาเหมือนกับส่วนของปัญหาในแบบรูปความต้องการสำหรับปัญหาโดเมนทุกประเด็น และไม่มีแนวทางขององค์ความรู้ใดที่เพิ่มเติมมาในส่วนปัญหา

3.4.4.2 สภาพแวดล้อม

ส่วนอธิบายสภาพแวดล้อมที่เกี่ยวข้องกับลักษณะของระบบในโดเมนระบบปลอดภัยเชิงวิกฤต โดยเป็นองค์ประกอบที่เพิ่มขึ้นจากแบบรูปสำหรับปัญหาโดเมน คือ การยอมรับอันตราย

3.4.4.3 ผลเฉลย

ส่วนที่อธิบายผลเฉลยสำหรับโดเมนของระบบปลอดภัยเชิงวิกฤต จะเน้นที่การอธิบายความต้องการที่มีความจำเป็นและมีลักษณะพิเศษสำหรับระบบปลอดภัยเชิงวิกฤต ซึ่งในองค์ประกอบของผลเฉลยของแบบรูปความต้องการสำหรับระบบปลอดภัยเชิงวิกฤตก็คือส่วนหนึ่งขององค์ประกอบที่มีอยู่ในแบบรูปสำหรับปัญหาโดเมนแต่จะให้ความสำคัญและเพิ่มเติมนรายละเอียดสำคัญให้มากขึ้น โดยมีองค์ประกอบที่เพิ่มขึ้นคือ ความต้องการความปลอดภัย และนโยบายความปลอดภัย

3.4.4.4 ส่วนสนับสนุน

ส่วนช่วยสนับสนุนการใช้งานของแบบรูปสำหรับปัญหาโดเมนของระบบปลอดภัยเชิงวิกฤตจะเป็นองค์ประกอบเพิ่มเติมที่ช่วยให้การทำวิศวกรรมความต้องการซอฟต์แวร์ทำได้ง่ายขึ้น โดยเป็นข้อมูลที่มีความสำคัญที่ใช้เพื่อด้านความปลอดภัยคือ อุบัติเหตุ และความเสี่ยงและอันตราย

3.4.4.5 องค์ประกอบที่อยู่ในแบบรูปความต้องการที่เกี่ยวข้องกับความปลอดภัย

องค์ความรู้ที่ควรมีอยู่ในแบบรูปความต้องการที่เกี่ยวข้องกับความปลอดภัย มีรายละเอียดและความสำคัญสรุปได้ดังตารางที่ 3.3 ดังนี้

ตารางที่ 3.3 รายละเอียดและความสำคัญสำหรับองค์ความรู้ในมุมมองกระบวนการความต้องการ

องค์ความรู้	รายละเอียด	ความสำคัญ
การยอมรับ อันตราย	เป็นเงื่อนไขสำคัญของระบบที่ถูกกำหนดตั้งแต่ต้น โครงการ เพื่อกำหนดระดับความปลอดภัยในปัจจุบันต่างๆ ที่สามารถส่งผลต่อชีวิต	ช่วยในการพิจารณาความปลอดภัย ทำให้ทราบระดับความสำคัญและผู้ที่ จะรับผิดชอบและดูแลความปลอดภัยในประเด็นที่กล่าวถึง
ความต้องการ ความปลอดภัย	เป็นหน้าที่ของระบบที่เกี่ยวข้องกับความปลอดภัยโดยตรง หน้าที่ส่วนความปลอดภัยจะเป็นการกำหนดในเรื่องของการป้องกัน การตรวจสอบ และการสนองตอบ ต่อลักษณะที่มีความปลอดภัย	ส่วนปลอดภัยถือเป็นความสามารถพื้นฐานที่ระบบจะต้องแจกแจงออกมา ซึ่งเป็นหัวใจในการนำไปใช้วิเคราะห์และออกแบบระบบให้ปลอดภัยจากอันตราย

ตารางที่ 3.3 รายละเอียดและความสำคัญสำหรับองค์ความรู้ในมุมมองกระบวนการความต้องการ
(ต่อ)

องค์ความรู้	รายละเอียด	ความสำคัญ
นัยความ ปลอดภัย	เป็นหน้าที่การทำงานทั่วไปของระบบ ซึ่งอาจจะเกิดอันตรายหรือเป็นสาเหตุของอันตรายได้ โดยเป็นผลที่ได้จากการพิจารณาความเป็นไปได้ควบคู่กับความรุนแรงที่เกิดตามมา	การพิจารณาความปลอดภัยในบางประเด็นอาจถูกมองข้าม นัยความปลอดภัยเป็นการแยกพิจารณาตามความน่าจะเป็นและอันตรายที่เกิดขึ้น เพื่อไม่ให้มองข้ามประเด็นเหล่านี้ไป
อุบัติเหตุ	เป็นมูลเหตุสำคัญที่ทำให้เกิดอันตรายได้ ซึ่งพิจารณาที่มาจากผิดพลาดของฮาร์ดแวร์ ความผิดพลาดที่เกิดจากตัวซอฟต์แวร์และความผิดพลาดจากการกระทำของผู้ใช้งาน	อุบัติเหตุทำให้ทราบปัจจัยที่ก่อให้เกิดความเสียหายที่ไม่อาจชี้ชัดได้ ความปลอดภัยของระบบต้องพิจารณาอุบัติเหตุในด้านต่างๆ ให้ครอบคลุมให้มากที่สุด เพื่อหาวิธีการจัดการกับอุบัติเหตุเหล่านั้น
ความเสี่ยงและ อันตราย	เป็นผลของเหตุการณ์หรืออันตรายที่เกิดขึ้น เป็นการสรุปอันตรายในลักษณะต่างๆที่เกิดขึ้นกับระบบ รวมทั้งแสดงผลลัพธ์ที่ได้หรือความเสียหายที่ตามมา	เป็นส่วนสำคัญที่นำมาใช้เป็นเกณฑ์ในการพิจารณาระบบ โดยระบบควรจะต้องพิจารณาหรือวิเคราะห์อันตรายเหล่านี้ เพื่อหาวิธีที่เหมาะสมในการจัดการ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

สรุปโครงสร้างและองค์ประกอบของแบบรูปความต้องการ

ในบทนี้จะเสนอนิยามของแบบรูปความต้องการ และสรุปโครงสร้างและความสัมพันธ์ของแบบรูปความต้องการ โดยในส่วนขององค์ประกอบในแบบรูปความต้องการจะเป็นการให้ความหมายและอธิบายรายละเอียด ตลอดจนกำหนดรูปแบบของเนื้อหาความในองค์ประกอบ เพื่อความชัดเจนและเหมาะสมที่สุด

4.1 โครงสร้างและความสัมพันธ์

4.1.1 นิยาม

แบบรูปความต้องการและโครงสร้างแบบรูปความต้องการสามารถกำหนดนิยามได้ดังนี้

แบบรูปความต้องการ คือ แบบรูปที่นำมาใช้กับวิศวกรรมความต้องการซอฟต์แวร์ โดยใช้ในการแก้ปัญหาที่เกิดขึ้นในงานวิศวกรรมความต้องการซอฟต์แวร์ ทั้งในส่วนของกระบวนการและการหาผลลัพธ์ที่เหมาะสมกับลักษณะโดเมน

โครงสร้างแบบรูปความต้องการ (Requirement Pattern Structure) คือ ความสัมพันธ์และองค์ประกอบของแบบรูปความต้องการ ซึ่งนำไปใช้ในการอธิบายความเกี่ยวข้องของแบบรูปความต้องการแต่ละประเภท รวมถึงการกำหนดความหมายและลักษณะขององค์ประกอบในแบบรูป

จากนิยามข้างต้นเป็นการกำหนดความหมายที่ชัดเจนให้แก่แบบรูปความต้องการ ซึ่งในงานวิทยานิพนธ์นี้ได้กล่าวถึงลักษณะของแบบรูปความต้องการที่เฉพาะเจาะจงหลายประเภท จึงมีการกำหนดนิยามของแบบรูปความต้องการประเภทต่างๆไว้ดังนี้

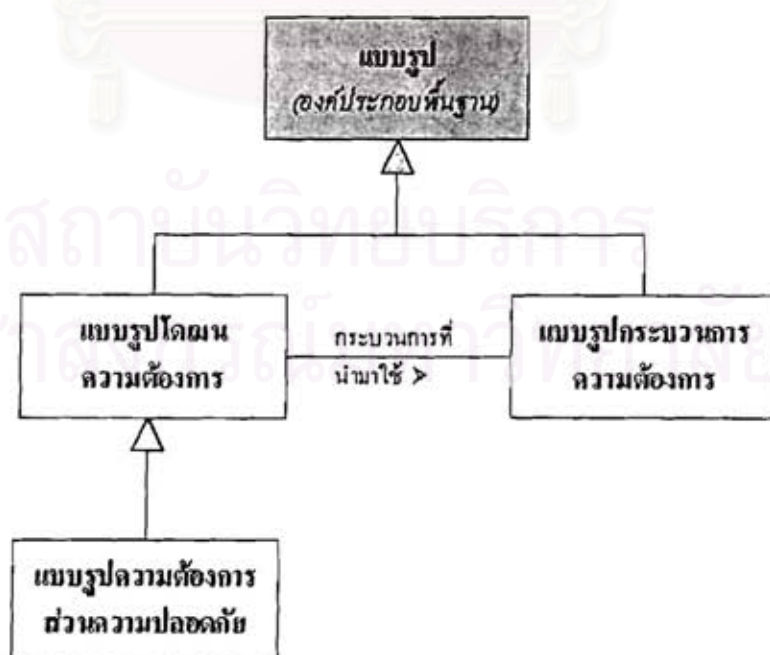
แบบรูปโดเมนความต้องการ (Requirement Domain Pattern) คือ แบบรูปความต้องการที่นำมาใช้แก้ปัญหาตามโดเมนของระบบ โดยเป็นการแก้ปัญหาที่เกิดขึ้นในงานวิศวกรรมความต้องการซอฟต์แวร์ของลักษณะโดเมนที่เจาะจง

แบบรูปกระบวนการความต้องการ (Requirement Process Pattern) คือ แบบรูปความต้องการที่นำมาใช้แก้ปัญหาในกระบวนการการทำงาน โดยเป็นการแก้ปัญหาที่เกิดขึ้นกับเทคนิคหรือวิธีการในวิศวกรรมความต้องการซอฟต์แวร์ที่นำมาใช้

แบบรูปความต้องการส่วนความปลอดภัย (Requirement Safety-Related Pattern) คือ แบบรูปโดเมนความต้องการที่นำมาใช้แก้ปัญหาที่เกิดขึ้นกับโดเมนที่เป็นระบบปลอดภัยเชิงวิกฤต โดยมีลักษณะ คุณสมบัติ และจุดประสงค์เช่นเดียวกับแบบรูปโดเมนความต้องการ หรืออาจกล่าวได้ว่าแบบรูปความต้องการส่วนความปลอดภัย คือแบบรูปโดเมนความต้องการประเภทหนึ่งที่ใช้แก้ปัญหาในส่วนขอระบบปลอดภัยเชิงวิกฤตที่เจาะจง

4.1.2 ความสัมพันธ์ระหว่างแบบรูป

แบบรูปความต้องการที่เสนอมานี้ เมื่อพิจารณาความสัมพันธ์หรือความเกี่ยวข้องระหว่างแบบรูปแต่ละประเภท จะเห็นได้ว่ามีความเกี่ยวข้องกันอย่างชัดเจน คือ แบบรูปโดเมนความต้องการมีการเสนอแบบรูปกระบวนการความต้องการที่มีความเหมาะสมในเรื่องของกระบวนการที่นำมาช่วย และแบบรูปความต้องการส่วนความปลอดภัยก็จัดได้ว่าเป็นประเภทย่อยของแบบรูปความโดเมนต้องการ โดยสามารถสรุปลักษณะความสัมพันธ์ของแบบรูปความต้องการทั้ง 3 ประเภทได้ดังแผนภาพคลาสในรูปที่ 4.1



รูปที่ 4.1 ความสัมพันธ์ของแบบรูปทั้ง 3 ประเภท

องค์ประกอบออกเป็นส่วนๆ โดยแต่ละส่วนมีจุดมุ่งหมายเพื่อประโยชน์ที่แตกต่างแต่นำมาใช้ร่วมกันได้เป็นอย่างดี จึงทำให้การนำไปใช้งานทำได้ง่าย ชัดเจน และเป็นระบบ

ลักษณะโครงสร้างของแบบรูปทั้ง 3 ประเภทสามารถสรุปได้ดังนี้

- 1) แบบรูปทั้ง 3 อยู่ในลักษณะโครงสร้างที่เป็นความสัมพันธ์ระหว่าง ปัญหา สภาพแวดล้อม และผลเฉลย โดยมีองค์ประกอบสำคัญที่ขยายความและสัมพันธ์กับโครงสร้างหลักทั้ง 3 ส่วน
- 2) แบบรูปโดเมนความต้องการและแบบรูปกระบวนการความต้องการมีความสัมพันธ์กันในลักษณะทิศทางเดียว โดยแบบรูปโดเมนความต้องการจะเสนอแบบรูปกระบวนการที่สามารถนำมาใช้งานประกอบได้
- 3) แบบรูปความต้องการส่วนความปลอดภัยเป็นแบบรูปที่ขยายมาจากแบบรูปโดเมนความต้องการ
- 4) การใช้แบบรูปความต้องการควรที่จะใช้แบบรูปโดเมนความต้องการและแบบรูปกระบวนการความต้องการร่วมกัน

4.2 องค์ประกอบพื้นฐานของแบบรูป

องค์ประกอบพื้นฐานของแบบรูปเป็นองค์ประกอบที่ปรากฏอยู่ในแบบรูปความต้องการทุกประเภท โดยองค์ประกอบเหล่านี้เป็นองค์ประกอบที่มีการนำไปใช้กับแบบรูปชนิดอื่นๆ มาก่อน ดังนั้นในวิทยานิพนธ์นี้จึงไม่กำหนดรูปแบบขององค์ประกอบในส่วนนี้ขึ้นมาใหม่แต่จะกำหนดรูปแบบให้เป็นไปในแนวทางใดแนวทางหนึ่งที่มีความเหมาะสม โดยองค์ประกอบที่ใช้ในแบบรูปมีความหมายและลักษณะการใช้งานสรุปได้ดังนี้

4.2.1 ชื่อแบบรูป

องค์ประกอบชื่อแบบรูป คือ ชื่อที่ใช้เรียกแบบรูป และยังเป็นชื่อที่ใช้ประโยชน์เพื่อการจัดประเภทหรือแบ่งกลุ่ม ตัวอย่างเช่น “ATP - Safety train separate” ซึ่งเป็นชื่อแบบรูปโดเมนความต้องการที่ใช้ในการทดลองของวิทยานิพนธ์นี้ โดย “ATP” เป็นชื่อของระบบย่อย และ “Safety train separate” เป็นชื่อที่เรียกใช้เรียกแบบรูป

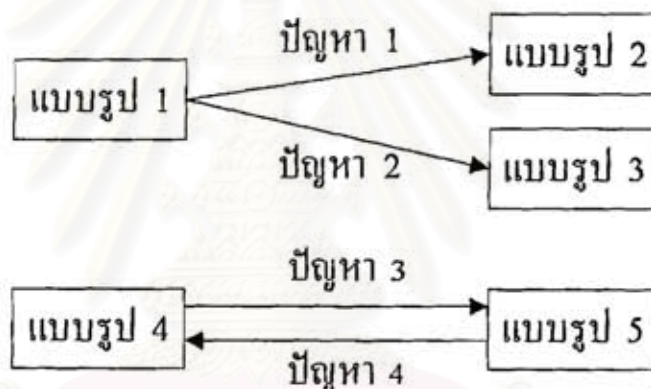
องค์ประกอบชื่อแบบรูปที่ใช้ในวิทยานิพนธ์นี้อยู่ในรูปแบบคำจำกัดความสั้นๆ ตามลักษณะของแบบรูป หรือเรียกตามชื่อที่นิยมใช้เรียกกัน โดยนอกจากเป็นการบอกลักษณะที่ใช้แล้วยังบอกลักษณะหรือกลุ่มประเภทของแบบรูปด้วย เช่น ใช้ “ATP” ในชื่อของแบบรูปโดเมนความต้องการที่เกี่ยวข้องกับระบบเดินรถไฟอัตโนมัติ และ “Elicitation” ในชื่อของแบบรูปกระบวนการความต้องการที่เกี่ยวข้องกับการเก็บข้อมูลความต้องการ

4.2.2 แบบรูปที่เกี่ยวข้อง

องค์ประกอบแบบรูปที่เกี่ยวข้อง คือ การแสดงความเกี่ยวเนื่องกับแบบรูปอื่นๆ อันเนื่องมาจากปัญหาในแบบรูปมีความสอดคล้องหรือเกี่ยวเนื่องไปยังปัญหาส่วนอื่นๆ

ในแบบรูปโดเมนความต้องการใช้ความหมายของแบบรูปที่เกี่ยวข้อง สำหรับแบบรูปความต้องการประเภทเดียวกัน โดยเป็นการอ้างถึงแบบรูปแบบทางเดียวโดยการอธิบายปัญหาที่เกี่ยวข้อง

แบบรูปที่เกี่ยวข้องเป็นการอ้างถึงจากทางเดียว ดังรูปที่ 4.6 แบบรูป 1 อ้างถึงแบบรูปที่เกี่ยวข้องผ่านปัญหา 1 ไปยังแบบรูป 2 และ ผ่านปัญหา 2 ไปยังแบบรูป 3 ซึ่งแบบรูป 2 และ แบบรูป 3 ไม่จำเป็นต้องอ้างกลับมายัง แบบรูป 1 หากทั้งสองไม่มีปัญหาให้อ้างกลับ ในขณะที่แบบรูป 4 อ้างผ่านปัญหา 3 ไปยังแบบรูป 5 และแบบรูป 5 อ้างกลับมายังแบบรูป 4 โดยผ่านปัญหา 4 ซึ่งเป็นปัญหาค้นตัวกับปัญหา 3



รูปที่ 4.6 ลักษณะการอ้างถึงแบบรูปที่เกี่ยวข้อง

องค์ประกอบแบบรูปที่เกี่ยวข้องสำหรับแบบรูปโดเมนความต้องการชนิดต่างๆมีความแตกต่างกันอยู่บ้าง ซึ่งจะได้อธิบายในลำดับถัดไปตามประเภทของแบบรูปความต้องการ

4.2.3 แรงชักจูง

องค์ประกอบแรงชักจูง คือ การชี้ลักษณะที่จะะจกลงไป เพื่อช่วยตัดสินใจเลือกแบบรูปมาแก้ปัญหา ซึ่งองค์ประกอบแรงชักจูงจะขึ้นจากการสังเกตผลของแบบรูป หากผู้ใช้แบบรูปพบว่าแรงชักจูงนั้นชี้นำผลที่ไม่ตรงหรือไม่เกี่ยวเนื่องกับที่คาดหมายไว้ ก็จะทราบได้ว่าการใช้แบบรูปนั้นไม่ตรงกับที่ต้องการ ในทางกลับกันองค์ประกอบแรงชักจูงก็ช่วยให้เกิดความมั่นใจและมีความชัดเจนในการใช้แบบรูปนั้นมากขึ้น

รูปแบบขององค์ประกอบแรงชักจูงที่ใช้ในแบบรูปความต้องการ อยู่ในลักษณะของรายการ ของลักษณะที่ชี้นำ

4.2.4 สภาวะผลลัพธ์ (Result Context)

องค์ประกอบสภาวะผลลัพธ์ คือ การแสดงผลที่คาดว่าจะเกิดขึ้นตามมาหลังจากใช้แบบรูป เช่น ลักษณะการทำงานของระบบที่ได้ ลักษณะพิเศษของระบบที่ปรากฏ เป็นต้น การพิจารณาผลลัพธ์ที่ตามมา ก่อนใช้แบบรูป จะช่วยให้ผู้ใช้แบบรูปสามารถตัดสินใจเลือกหรือไม่เลือกใช้แบบรูปได้ดียิ่งขึ้น เพราะได้รับทราบผลลัพธ์หรือผลกระทบที่จะเกิดขึ้นตามมา

รูปแบบขององค์ประกอบสภาวะผลลัพธ์ที่ใช้ในแบบรูปความต้องการจะเป็นการบรรยายผลลัพธ์ที่สังเกตได้

4.2.5 การนำไปใช้ที่ทราบ (Known Use)

องค์ประกอบการนำไปใช้ที่ทราบ คือ การแสดงให้เห็นว่าแบบรูปสามารถนำไปใช้แก้ปัญหาได้จริง ซึ่งจะเป็นการขกกรณีการใช้แบบรูปที่ประสบผลสำเร็จ ดังนั้นการนำไปใช้ที่ทราบจึงช่วยให้ผู้ใช้แบบรูปมีความมั่นใจในตัวแบบรูป และเห็นว่าผลลัพธ์ที่ได้ปรากฏขึ้นจริงอย่างไร

องค์ประกอบการนำไปใช้ที่ทราบในแบบรูปทุกๆ ไปนิยมยกตัวอย่างผลสำเร็จที่เกิดขึ้นเปรียบเสมือนการอ้างอิงไปยังผลลัพธ์ที่มีการสร้างขึ้นจริงจากแบบรูป ดังนั้นรูปแบบของการนำไปใช้ที่ทราบที่ใช้ในแบบรูปความต้องการจึงเป็นลักษณะของการบรรยายไปยังผลสำเร็จในการใช้แบบรูป

4.2.6 ตัวอย่าง

องค์ประกอบตัวอย่าง คือ การแสดงตัวอย่างของการนำแบบรูปไปใช้ ซึ่งรูปแบบของตัวอย่างนั้นก็จะแปรตามลักษณะโครงสร้างและการใช้งานของแบบรูปแต่ละชนิด โดยแบบรูปโดเมนความต้องการและแบบรูปกระบวนการความต้องการต่างก็มีรูปแบบขององค์ประกอบตัวอย่างที่แตกต่างกัน

อย่างไรก็ดีในวิทยานิพนธ์นี้ได้พยายามจัดรูปแบบขององค์ประกอบตัวอย่างสำหรับแบบรูปความต้องการ ซึ่งจะมีรูปแบบขึ้นอยู่กับแบบรูปแต่ละประเภท โดยจะกล่าวในลำดับถัดไป

นอกจากองค์ประกอบที่กล่าวมาข้างต้นแล้ว ยังมีปัญหา ผลเฉลย และสภาพแวดล้อม ที่เป็นองค์ประกอบในแบบรูปพื้นฐาน ซึ่งทั้ง 3 ส่วน เป็นเพียงโครงร่างเท่านั้น การใช้งานทั้ง 3 ส่วนจะเป็นไปตามชนิดของแบบรูป ซึ่งในวิทยานิพนธ์นี้ได้พิจารณาส่วนที่ขยายความเพื่อให้เกิดความชัดเจนและสมบูรณ์ตามจุดประสงค์ โดยจะขอกกล่าวถึงองค์ประกอบที่มาแทนความหมายเหล่านี้ในแบบรูปความต้องการแต่ละประเภทในลำดับต่อไป

4.3 องค์ประกอบของแบบรูปโดเมนความต้องการ

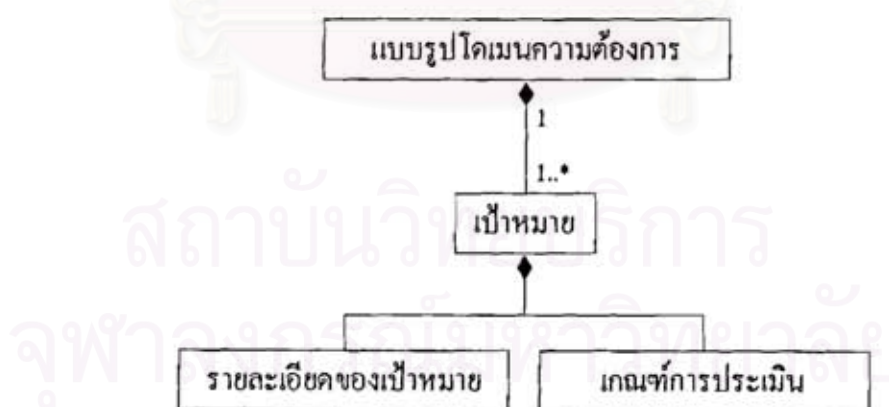
องค์ประกอบของแบบรูปโดเมนความต้องการประกอบด้วย องค์ประกอบที่มาจากแบบรูปพื้นฐาน และองค์ประกอบที่เจาะจงสำหรับโดเมน ซึ่งในหัวข้อนี้จะแสดงองค์ประกอบที่เพิ่มเข้ามาโดยอธิบายความหมายและรูปแบบขององค์ประกอบเพื่อให้มีความชัดเจนและสมบูรณ์ในการใช้งาน โดยมีรายละเอียดขององค์ประกอบต่างๆ ดังนี้

4.3.1 เป้าหมาย (Goal)

เป้าหมาย คือ หัวใจของความต้องการระบบ เนื่องจากเป้าหมายจะเป็นหลักในการกำหนดและวัดผลสำเร็จของระบบ หรือกล่าวได้ว่าซอฟต์แวร์ที่พัฒนาจะถือว่าประสบความสำเร็จหรือล้มเหลวจะอาศัยเป้าหมายเป็นตัวตัดสิน

องค์ประกอบเป้าหมาย คือ ส่วนสำคัญในการทำวิศวกรรมความต้องการ โดยใช้อธิบายเป้าหมายหลักในการใช้แบบรูปโดเมนความต้องการเมื่อการทำงานเกิดปัญหาและต้องการใช้แบบรูป เป้าหมายจะเป็นตัวแทนของความต้องการ ปัญหาในวิศวกรรมความต้องการก็มีมูลเหตุมาจากเป้าหมาย

แบบรูปโดเมนความต้องการจำเป็นต้องมีเป้าหมายกำหนดไว้เสมอและอาจมีมูลเหตุที่เป็นเป้าหมายให้ใช้แบบรูปหลายเป้าหมาย โดยรูปแบบของเป้าหมายจะเป็นดังรูปที่ 4.7 โดยประกอบด้วยส่วนที่บรรยายรายละเอียดและเกณฑ์การยอมรับเป้าหมาย ซึ่งเกณฑ์การยอมรับถือเป็นส่วนสำคัญที่เป็นตัวแทนการบรรลุผลของวิศวกรรมความต้องการ



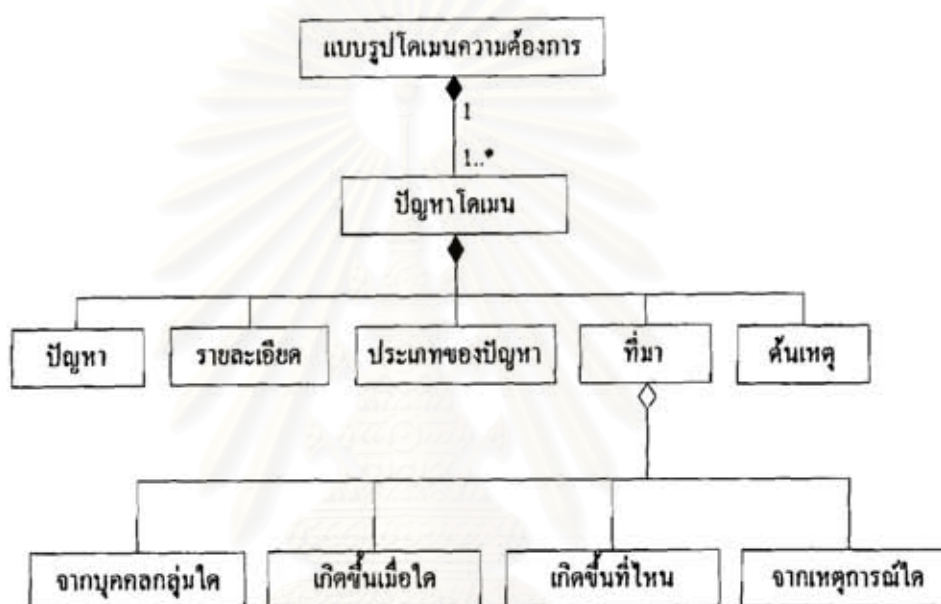
รูปที่ 4.7 รูปแบบขององค์ประกอบเป้าหมายของแบบรูปโดเมนความต้องการ

4.3.2 ปัญหาโดเมน (Domain Problem)

ปัญหาโดเมน คือ ลักษณะของระบบที่เป็นมูลเหตุของการพัฒนา กล่าวได้ว่าเป็นลักษณะปัญหาที่ระบบปัจจุบันประสบอยู่ก่อนที่จะมีการพัฒนาระบบใหม่ เป็นส่วนที่แสดงให้เห็นลักษณะของธุรกรรมภายในระบบ ปัญหาโดเมนช่วยให้การแสดงความต้องการพื้นฐานของระบบมีความชัดเจนขึ้น เป็นเหตุเป็น

ผลและมีที่นำไปตามลักษณะปัญหาที่ปรากฏอยู่ โดยอธิบายถึงปัญหาในเชิงธุรกิจตลอดจนจุดสำคัญที่ต้องการเพิ่มประสิทธิภาพและประสิทธิผลของการทำงาน

องค์ประกอบปัญหาโดเมน คือ การอธิบายปัญหาที่เกิดขึ้นจากลักษณะของตัวระบบหรือโดเมน ซึ่งเป็นที่มาของความต้องการซอฟต์แวร์ โดยปัญหาโดเมนจะช่วยให้ผู้ใช้แบบรูปมีความเข้าใจต้นเหตุหรือความสำคัญในความต้องการ ซึ่งรูปแบบขององค์ประกอบปัญหาโดเมนสามารถแสดงได้ดังรูปที่ 4.8



รูปที่ 4.8 รูปแบบองค์ประกอบปัญหาโดเมนของแบบรูปโดเมนความต้องการ

จากรูปที่ 4.8 อธิบายได้ว่าแบบรูปโดเมนความต้องการมีองค์ประกอบปัญหาโดเมนหลายอัน โดยองค์ประกอบปัญหาโดเมนประกอบด้วย ปัญหา รายละเอียด ประเภทของปัญหา ที่มา และสาเหตุ โดยมีรูปแบบแต่ละส่วน ดังนี้

- 1) ปัญหา คือ ชื่อที่ใช้เรียกหรือใช้บ่งบอกปัญหา
- 2) รายละเอียด คือ การอธิบายปัญหา สภาพของปัญหา รายละเอียดที่สำคัญของปัญหา
- 3) ประเภทของปัญหา คือ ประเภทของปัญหาโดเมนที่กำหนดไว้แล้ว ซึ่งมีประเภทที่เป็นไปได้ 5 รูปแบบ ดังนี้

การแก้ปัญหาเฉพาะหน้า (Solve) เป็นปัญหาโดเมนที่มีที่มาจากความจำเป็นในการแก้ปัญหาในลักษณะเร่งด่วน เป็นความต้องการที่เกิดขึ้นและหวังผลภายในระยะเวลาอันสั้น

การเพิ่มประสิทธิภาพ (Improve) เป็นปัญหาโดเมนที่มาจากการพยายามปรับปรุงบางสิ่งบางอย่างในระบบให้ดีขึ้นกว่าเดิม หรือเป็นไปเพื่อเพิ่มประสิทธิภาพของระบบ

โดยที่ความต้องการนั้นไม่ได้ต้องการให้เกิดความเปลี่ยนแปลงหรือกระทบกับการทำงานมากนัก

การเปลี่ยนแปลงการทำงาน (Change) เป็นปัญหาโดเมนที่เกิดจากการปรับเปลี่ยนการทำงาน ซึ่งทำให้การดำเนินการมีการเปลี่ยนแปลงและต้องทำการแก้ไขให้เหมาะสม

การริเริ่มการทำงานแบบใหม่ (Initiate) เป็นปัญหาโดเมนที่มาจาก การริเริ่มการทำงานใหม่ๆ หรือความจำเป็นต้องเปลี่ยนแปลงระบบงาน เป็นความต้องการที่เกิดขึ้นใหม่ ไม่เคยมีแนวทางมาก่อน ทำให้ระบบหรือการทำงานเปลี่ยนแปลงไปจากเดิม

การกำหนดเกี่ยวกับงาน (Define) เป็นปัญหาโดเมนที่เกิดขึ้นจากความจำเป็นในการกำหนดเงื่อนไขในระบบหรือการทำงาน ซึ่งจะต้องกำหนดเกี่ยวกับงานให้สอดคล้องกับข้อจำกัดหรือความต้องการหลักของระบบ

- 4) *ที่มา* คือ ที่มาหรือความเป็นมาของปัญหานั้น อาจหมายถึงจุดเริ่มต้นหรือปัจจัยที่ก่อให้เกิดปัญหาโดเมนขึ้น โดยสามารถจำแนกที่มาได้เป็น 4 ประเภท ดังนี้

ที่มาจากกลุ่มบุคคล คือ ปัญหาที่มาจากบุคคลกลุ่มใดกลุ่มหนึ่ง ซึ่งเป็นต้นเหตุของปัญหาหรือเป็นผู้ให้ความต้องการ

ที่มาจากช่วงเวลา คือ ปัญหาที่เกิดขึ้นอย่างซ้ำๆ ในช่วงเวลาเดิมๆ ปัญหาที่เกิดขึ้นเป็นประจำในช่วงเวลาที่แน่นอน เช่น การเกิดปัญหาทุกรอบเดือน

ที่มาจากสถานที่ คือ ปัญหาที่เกิดขึ้นในสถานที่หรือบริเวณที่มีลักษณะคล้ายๆ เดิม

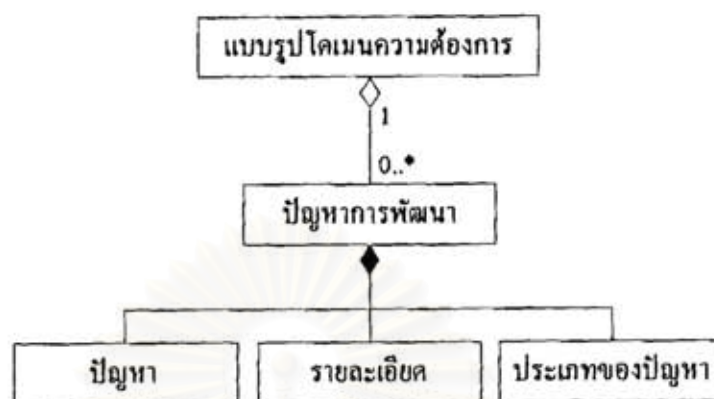
ที่มาจากเหตุการณ์ คือ ปัญหาที่เกิดขึ้นตามมาเสมอเมื่อมีเหตุการณ์บางอย่างในลักษณะคล้ายๆ เดิมเกิดขึ้น

- 5) *ต้นเหตุ* คือ สาเหตุที่แท้จริงของปัญหาโดเมน ซึ่งได้จากการวิเคราะห์สาเหตุหลักของปัญหา โดยอยู่ในรูปแบบการบรรยาย

4.3.3 ปัญหาการพัฒนา (Development Problem)

ปัญหาของการพัฒนา คือ มุมมองของความต้องการที่พิจารณาขึ้นตามทีมนพัฒนาระบบ ทั้งในเรื่องของเทคโนโลยีที่ใช้อยู่ ข้อจำกัดต่างๆ ของการพัฒนาในปัจจุบัน ซึ่งเป็นสาเหตุสำคัญทำให้ไม่สามารถพัฒนาระบบไปตามเป้าหมายที่สมบูรณ์ได้

องค์ประกอบปัญหาการพัฒนาเป็นการอธิบายปัญหาในการทำความเข้าใจหรือปัญหาที่เกิดขึ้นจากมุมมองของผู้พัฒนาในการพยายามพัฒนาระบบให้เป็นไปตามความต้องการของผู้ใช้ โดยสามารถแสดงรายละเอียดขององค์ประกอบปัญหาการพัฒนาได้ดังรูปที่ 4.9



รูปที่ 4.9 รูปแบบองค์ประกอบปัญหาการพัฒนาของแบบรูปโดเมนความต้องการ

จากรูปที่ 4.9 แบบรูปโดเมนความต้องการอาจมีปัญหาการพัฒนาได้มากกว่าหนึ่งหรืออาจไม่มีปัญหาระบุไว้ก็ได้ โดยองค์ประกอบปัญหาพัฒนาจะประกอบไปด้วย 3 ส่วน ดังนี้

- 1) ปัญหา คือ ชื่อที่ใช้เรียกหรือใช้บ่งบอกปัญหา
- 2) รายละเอียด คือ การอธิบายปัญหา สภาพ และรายละเอียดที่สำคัญของปัญหา
- 3) ประเภทของปัญหา คือ ประเภทของปัญหาในการพัฒนาซอฟต์แวร์ให้เป็นไปตามที่กำหนดไว้ ซึ่งมี 4 ประเภท ดังนี้

ข้อจำกัดของเทคโนโลยี เป็นปัญหาที่เกิดขึ้นจากขีดจำกัดของเทคโนโลยีที่มีอยู่ ทำให้ไม่สามารถทำให้ความต้องการเกิดขึ้นได้จริง

ประสิทธิภาพของระบบ เป็นปัญหาที่เกิดจากความเป็นไปได้ของการพัฒนาตามความต้องการที่ส่งผลต่อประสิทธิภาพการทำงานของระบบ

ความสามารถในการพัฒนา เป็นขีดจำกัดของบุคคลในทีมพัฒนาที่ไม่สามารถทำในระดับที่ความต้องการกำหนดได้ เช่น ขาดแคลนความรู้ความสามารถเฉพาะด้าน

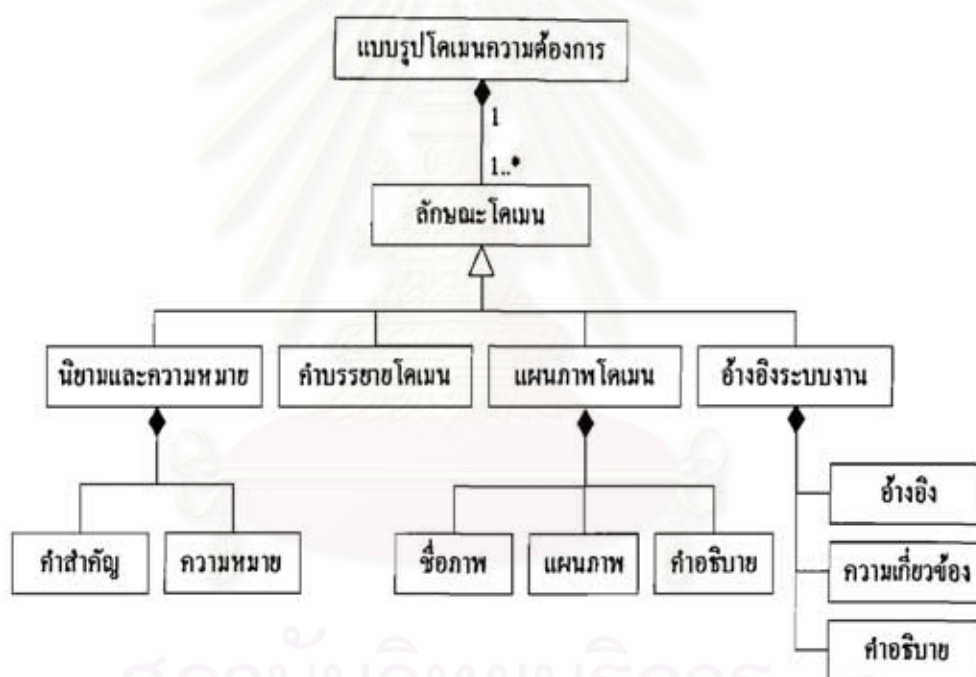
ค่าใช้จ่ายในการพัฒนา เป็นปัญหาของค่าใช้จ่าย ทั้งในด้านของเวลา เงินลงทุน หรือทรัพยากรที่ใช้ ซึ่งเป็นผลตามมาจากลักษณะของความต้องการ

4.3.4 ลักษณะโดเมน (Business Domain)

ลักษณะโดเมน คือ ส่วนที่ช่วยให้เข้าใจพฤติกรรมหรือลักษณะมูลฐาน โดยลักษณะโดเมนจะอธิบายบริบทของโดเมน และจุดมุ่งหมาย เช่น ลักษณะโดเมนของระบบรถไฟฟ้าคือ โดเมนของระบบขนส่งมวลชน เพื่อรองรับผู้โดยสารเป็นจำนวนมาก โดยมีการจัดการระบบอย่างเข้มงวดและเป็นแบบแผน และทำงานเป็นระบบตามโครงสร้างเครือข่ายการคมนาคม เป็นต้น

ลักษณะโดเมนยังใช้ให้ความหมายสำหรับโดเมน ไม่ว่าจะเป็นความหมายของบุคคลที่เกี่ยวข้องกับการใช้งานระบบ หรือความหมายของส่วนที่เป็นองค์ประกอบ หรือค่าเฉพาะที่เกี่ยวข้องกับโดเมน เพื่อให้สามารถเข้าใจความหมายขององค์ประกอบในโดเมนได้อย่างสมบูรณ์และถูกต้องตรงกัน

องค์ประกอบลักษณะโดเมนเป็นส่วนที่อธิบายสภาพหรือพฤติกรรมของระบบ กิจกรรมการทำงาน รวมไปถึงลักษณะองค์กรที่ใช้งานระบบ องค์ประกอบลักษณะโดเมนเป็นสภาวะแวดล้อมภายใน ซึ่งเป็นส่วนสำคัญที่จะนำมาใช้พิจารณาแบบรูปว่ามีความเหมาะสมกับปัญหาหรือไม่ เมื่อมีการพิจารณาปัญหานั้น นำแบบรูปโดเมนความต้องการมาใช้ ลักษณะโดเมนจะเป็นเงื่อนไขที่จะนำมาใช้พิจารณาว่าลักษณะระบบที่ใช้แบบรูปมีสภาพคล้ายหรือตรงกันหรือไม่ โดยสามารถแสดงรายละเอียดขององค์ประกอบลักษณะโดเมนได้ดังรูปที่ 4.10



รูปที่ 4.10 รูปแบบองค์ประกอบลักษณะโดเมนของแบบรูปโดเมนความต้องการ

การอธิบายลักษณะโดเมนมีอยู่ด้วยกันหลายวิธีและสามารถอธิบายได้อย่างหลากหลาย รูปแบบขององค์ประกอบลักษณะโดเมนที่กำหนดในวิทยานิพนธ์จึงสามารถแจกแจงได้หลายแบบ ซึ่งอาจใช้อย่างใดอย่างหนึ่งในการอธิบายหรือใช้ร่วมกันเพื่ออธิบายลักษณะโดเมนก็ได้ รูปแบบขององค์ประกอบลักษณะโดเมนสามารถแสดงได้ดังรูปที่ 4.10 โดยรูปแบบต่างๆ ขององค์ประกอบลักษณะโดเมนมีรายละเอียด ดังนี้

- 1) นิยามและความหมาย เป็นรูปแบบของการบรรยายและให้ความหมายสำหรับคำศัพท์เฉพาะ หรือเรื่องเฉพาะที่เกี่ยวข้องกับโดเมน เพื่อให้เกิดความเข้าใจที่ชัดเจนและถูกต้อง รูปแบบของนิยามและความหมายประกอบด้วย 2 ส่วน ดังนี้

คำสำคัญ เป็นคำที่มีความหมายเป็นพิเศษแตกต่างจากที่มีการใช้อยู่ทั่วไป หรืออาจเป็นคำที่มีความหมายที่ต้องมีการอธิบายในความหมายของคำเพิ่มเติมตามที่มีความหมายเฉพาะใน โดเมน

ความหมาย เป็นส่วนของการอธิบายคำสำคัญ เป็นการบรรยายความหมายของคำสำคัญให้มีความชัดเจนและเจาะจงไปยังความหมายที่ใช้ใน โดเมน

- 2) คำบรรยายโดเมน เป็นรูปแบบของการบรรยายลักษณะ โดเมนทั้งลักษณะ โครงสร้าง การทำงาน องค์กร พฤติกรรมการทำงาน และลักษณะการใช้งานระบบ อาจเป็นการบรรยายแบบข้อความหรือบรรยายแบบรายการ
- 3) แผนภาพโดเมน เป็นรูปแบบของการอธิบายโดเมนด้วยแผนภาพ ซึ่งเป็นที่นิยมและสามารถเสนอลักษณะ โดเมนได้ดีทั้งในรูปแบบ โครงสร้างและพฤติกรรมของโดเมน รูปแบบของแผนภาพโดเมนประกอบด้วย 3 ส่วน ดังนี้

ชื่อภาพ เป็นชื่อที่ตั้งขึ้นหรือใช้เรียกแผนภาพที่นำมาอธิบายลักษณะ โดเมน

แผนภาพ เป็นตัวภาพที่นำมาประกอบ

คำอธิบาย เป็นส่วนที่ใช้อธิบายรายละเอียดของแผนภาพที่นำมาประกอบ

- 4) อ้างอิงระบบงาน เป็นการอ้างอิงไปยังระบบงานที่เกี่ยวข้องหรือลักษณะการทำงานบางอย่างที่สามารถทดแทนหรือมีลักษณะที่เหมือนกันกับโดเมนที่กล่าวถึง ซึ่งรูปแบบของอ้างอิงระบบงานประกอบด้วย 3 ส่วน ดังนี้

อ้างอิง เป็นส่วนที่กล่าวถึงหรืออ้างอิงไปยังแหล่งที่มา หรือแหล่งข้อมูลเพิ่มเติมสำหรับระบบงานที่นำมาเทียบเคียงกับลักษณะ โดเมน

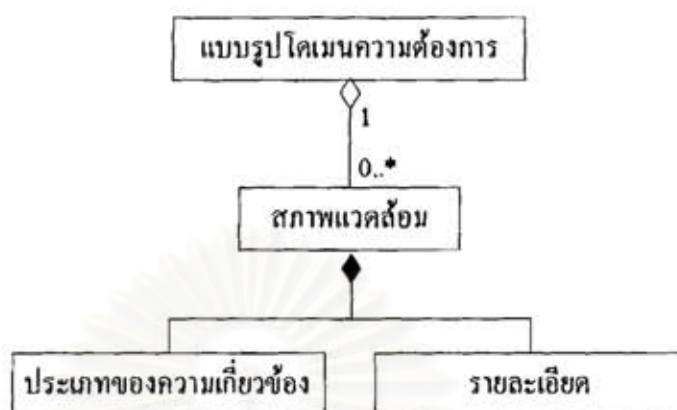
ความเกี่ยวข้อง เป็นคำอธิบายสั้นๆเกี่ยวกับความสัมพันธ์หรือความเกี่ยวข้องของระบบงานที่เทียบเคียงได้

คำอธิบาย เป็นการอธิบายเกี่ยวกับระบบที่เทียบเคียงว่ามีรายละเอียดเบื้องต้นอย่างไร และขยายความในส่วนที่เหมือนหรือแตกต่างกัน

4.3.5 สภาพแวดล้อม (Environments)

สภาพแวดล้อม คือ ส่วนที่อธิบายโดเมนที่สนใจว่ามีความเกี่ยวข้องกับระบบในโดเมน องค์กร หรือคนและกลุ่มคนอื่นๆ อย่างไร โดยที่ความเกี่ยวข้องเหล่านั้นอาจส่งผลต่อพฤติกรรมของระบบ ซึ่งผลจากการพิจารณาและสังเกตสภาพแวดล้อม จะนำมาใช้กำหนดและเปรียบเทียบสำหรับการพัฒนาระบบ

องค์ประกอบสภาพแวดล้อมเป็นการกล่าวถึงปัจจัยที่มาจากภายนอก ได้แก่ ระบบ องค์กร และกลุ่มบุคคลภายนอก โดยองค์ประกอบสภาพแวดล้อมจะใช้เป็นปัจจัยประกอบการพิจารณาเลือกใช้แบบรูป โดเมนความต้องการ ซึ่งรูปแบบองค์ประกอบสภาพแวดล้อมสามารถแสดงได้ดังรูปที่ 4.11



รูปที่ 4.11 รูปแบบองค์ประกอบสภาพแวดล้อมของแบบรูปโดเมนความต้องการ

ในแบบรูปโดเมนความต้องการอาจมีปัจจัยภายนอกหลายด้านหรืออาจไม่มีความเกี่ยวข้องกับปัจจัยภายนอกก็ได้ ดังนั้นองค์ประกอบสภาพแวดล้อมจึงประกอบด้วยปัจจัยที่เกี่ยวข้องได้หลายปัจจัย โดยรูปแบบขององค์ประกอบสภาพแวดล้อมประกอบด้วย 2 ส่วน ดังนี้

- 1) ประเภทของความเกี่ยวข้อง ซึ่งเป็นรูปแบบต่างๆ ที่เป็นไปได้ของความเกี่ยวข้องกับลักษณะภายนอก โดยประเภทของความเกี่ยวข้องมีด้วยกัน 3 ประเภท ดังนี้

ระบบภายนอกที่สัมพันธ์ คือความเกี่ยวข้องกับระบบอื่นๆ ที่ต่างก็ทำงานเป็นเอกเทศ แต่อาจมีส่วนที่ติดต่อกัน เชื่อมโยง หรือการทำงานบางประการร่วมกัน

องค์กรภายนอกที่สัมพันธ์ คือความเกี่ยวข้องกับองค์กรภายนอก ทั้งที่มีผลต่อการกำหนดทิศทางของระบบโดยตรงและโดยทางอ้อม

บุคคลภายนอกที่สัมพันธ์ คือกลุ่มบุคคลที่มีผลกระทบจากระบบหรือส่งผลกระทบต่อการทำงานของระบบโดยที่ไม่สามารถกำหนดกลุ่มบุคคลเหล่านี้ได้อย่างชัดเจนหรือยากเกินกว่าจะจำแนกได้ ซึ่งรวมไปถึงกลุ่มบุคคลที่คาดว่าจะเข้ามาใช้ระบบที่มีความหลากหลาย

- 2) รายละเอียดของความเกี่ยวข้อง เป็นการอธิบายลักษณะภายนอกที่สัมพันธ์ โดยการอ้างถึงและบรรยายรายละเอียดของความสัมพันธ์นั้น

4.3.6 ข้อจำกัดและกฎเกณฑ์ (System Constraint)

ข้อจำกัดและกฎเกณฑ์ คือ ส่วนที่แสดงเงื่อนไขสำคัญที่กำหนดความต้องการของตัวระบบหรือข้อจำกัดในเชิงธุรกรรมของโดเมน รวมทั้งเป็นเงื่อนไขของการพัฒนาระบบด้วย ข้อจำกัดและกฎเกณฑ์อาจมีการเปลี่ยนแปลงได้ เช่น ในกรณีที่มีปัจจัยที่สำคัญกว่าเข้ามาใหม่การกำหนดบางอย่างจึงอาจจะต้องเปลี่ยนแปลงไปให้เกิดความเหมาะสม

องค์ประกอบข้อจำกัดและกฎเกณฑ์เป็นส่วนที่อธิบายขอบเขตข้อจำกัดของระบบ ข้อจำกัดเหล่านี้ อาจเป็นข้อบังคับหรือมาตรฐานที่กำหนดขึ้นสำหรับระบบ โดเมน รูปแบบขององค์ประกอบข้อจำกัดและ กฎเกณฑ์จะแสดงรายการข้อบังคับดังรูปที่ 4.12



รูปที่ 4.12 รูปแบบขององค์ประกอบข้อจำกัดและกฎเกณฑ์ของแบบรูปโดเมนความต้องการ

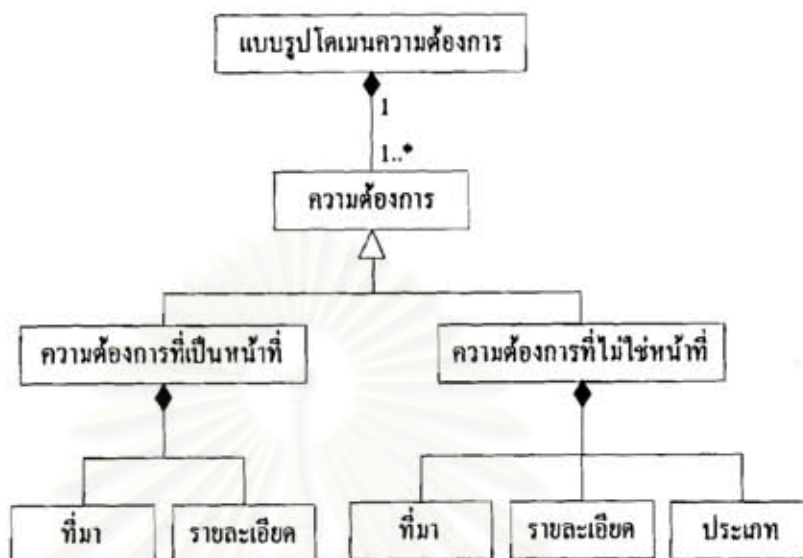
องค์ประกอบข้อจำกัดและกฎเกณฑ์อยู่ในรูปแบบของการบรรยายข้อจำกัดหรือกฎเกณฑ์ที่มี ซึ่งใน แบบรูปโดเมนความต้องการอาจไม่มีข้อจำกัดหรือกฎเกณฑ์เลยก็ได้ และอาจมีข้อจำกัดหรือกฎเกณฑ์ที่ เกี่ยวข้องหลายอัน

4.3.7 ความต้องการ (Requirements)

ความต้องการ คือ รายละเอียดที่ระบบจะต้องทำได้ ซึ่งเกิดจากการรวบรวมความต้องการของผู้ใช้ โดยความต้องการจะเป็นตัวกำหนดว่าระบบจะต้องทำอะไรได้บ้าง โดยสรุปจากความต้องการที่ผู้ใช้ระบบ มีขึ้น

สิ่งที่ไม่ได้มีอยู่ในความต้องการอาจไม่ได้หมายความว่าสามารถละเลยไม่ต้องมีในระบบ ยกเว้นแต่ ความต้องการที่กำหนดไม่ให้มีบางสิ่ง ความต้องการจะอธิบายองค์ประกอบที่ระบบจะต้องรองรับในส่วนที่ มีความชัดเจนแล้ว ซึ่งบางอย่างที่ระบบอาจจะมีหรือไม่มีก็ได้ หรืออาจมีในลักษณะที่เป็นไปได้หลายระดับ จัดได้ว่าเป็นความต้องการที่ไม่ใช่หน้าที่

องค์ประกอบความต้องการคือ ส่วนของความต้องการที่เป็นหน้าที่และไม่ใชหน้าที่ ซึ่งถือเป็นส่วน สำคัญของผลเฉลยของแบบรูปโดเมนความต้องการ ลักษณะขององค์ประกอบความต้องการสามารถแสดง ได้ดังรูปที่ 4.13



รูปที่ 4.13 รูปแบบองค์ประกอบความต้องการของแบบรูปโดเมนความต้องการ

องค์ประกอบความต้องการในแบบรูปโดเมนความต้องการเป็นองค์ประกอบสำคัญที่ขาดไม่ได้ โดยรูปแบบองค์ประกอบความต้องการมี 2 ประเภท ดังนี้

- 1) ความต้องการที่เป็นหน้าที่ คือ องค์ประกอบความต้องการ ในด้านของความต้องการที่เป็นหน้าที่ โดยประกอบด้วย 2 ส่วน ดังนี้

ที่มา เป็นส่วนที่อธิบายที่มาของความต้องการนั้นว่ามาจากอะไร

รายละเอียด เป็นการบรรยายอย่างสั้นเกี่ยวกับความต้องการ เพื่อบอกว่าความต้องการนั้นต้องทำอะไร

- 2) ความต้องการที่ไม่ใช่หน้าที่ คือ รูปแบบของความต้องการที่ไม่ใช่หน้าที่ซึ่งประกอบด้วย 3 ส่วน ดังนี้

ที่มา เป็นส่วนที่อธิบายที่มาของความต้องการนั้นว่ามาจากอะไร

รายละเอียด เป็นการบรรยายอย่างสั้นเกี่ยวกับความต้องการ เพื่อบอกว่าความต้องการนั้นต้องการให้เกิดผลอย่างไรกับซอฟต์แวร์

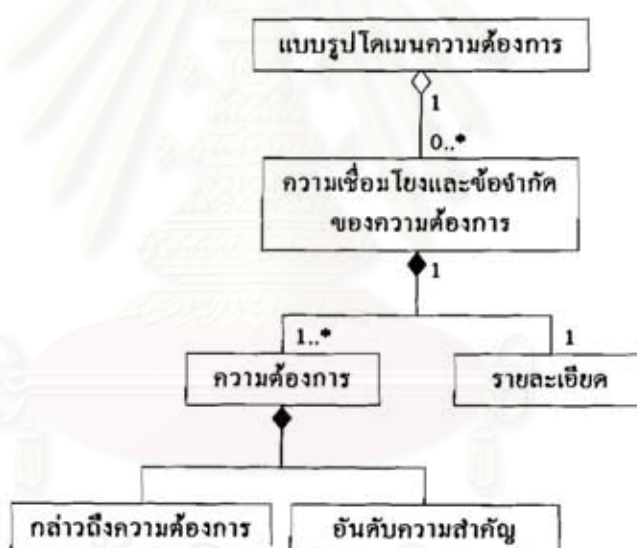
ประเภท เป็นส่วนที่ใช้จำแนกประเภทของความต้องการที่ไม่ใช่หน้าที่ ซึ่งประเภทของความต้องการที่ไม่ใช่หน้าที่ที่กำหนดมี 4 ประเภท ดังนี้

- 1) ความต้องการที่เป็นคุณสมบัติของซอฟต์แวร์ (Product Requirements)
- 2) ความต้องการทางด้านการกระบวนการและวิธีการ (Process Requirements)
- 3) ความต้องการด้านประสิทธิภาพ (Performance Requirements)
- 4) ความต้องการด้านส่วนต่อประสาน (Interface Requirements)

4.3.8 ความเชื่อมโยงและข้อจำกัดของความต้องการ (Requirements Relationship and Constraint)

ความเชื่อมโยงและข้อจำกัดของความต้องการ คือ การอธิบายความสัมพันธ์ระหว่างแต่ละความต้องการ เพื่อให้สามารถแจกแจงระดับความสำคัญได้ และในกรณีที่ความต้องการเกิดความขัดแย้ง ไม่สอดคล้อง หรือไม่สามารถบรรลุเป้าหมายร่วมกัน ความเชื่อมโยงและข้อจำกัดจะเป็นตัวตัดสินใจและช่วยในการเลือกความต้องการที่สำคัญกว่าได้

องค์ประกอบความเชื่อมโยงและข้อจำกัดของความต้องการ คือ การขยายความส่วนขององค์ประกอบความต้องการ ในแง่ของการเชื่อมโยง ความสัมพันธ์ และขอบเขตของความต้องการ เพื่อให้มีความชัดเจนและเป็นการจัดลำดับความสำคัญของความต้องการ ซึ่งรูปแบบขององค์ประกอบความเชื่อมโยงและข้อจำกัดของความต้องการแสดงได้ดังรูปที่ 4.14



รูปที่ 4.14 รูปแบบองค์ประกอบความเชื่อมโยงและข้อจำกัดของความต้องการของแบบรูปโดเมนความต้องการ

จากรูปที่ 4.14 องค์ประกอบความเชื่อมโยงและข้อจำกัดของความต้องการ เป็นองค์ประกอบที่แสดงความเกี่ยวข้องหรือความสำคัญของความต้องการ โดยจะมีหรือไม่ก็ได้ ซึ่งรูปแบบของความเชื่อมโยงและข้อจำกัดของความต้องการประกอบด้วย 2 ส่วน ดังนี้

- 1) ความต้องการ คือ ความต้องการหรือกลุ่มของความต้องการที่มีความสัมพันธ์กัน หรือให้ความสำคัญเพื่อพิจารณาในรายละเอียด ซึ่งประกอบด้วย 2 ส่วน ดังนี้

กล่าวถึงความต้องการ เป็นส่วนที่อ้างอิงไปถึงองค์ประกอบความต้องการที่ได้มีการเสนอในแบบรูป

อันดับความสำคัญ เป็นการให้คะแนนความสำคัญกับตัวความต้องการนั้น อาจใช้
ในจุดประสงค์เชิงเปรียบเทียบหรือเพื่อแสดงว่ามีความต้องการมีความสำคัญในอันดับที่
เท่ากัน ซึ่งอันดับ 1 แทนความหมายว่ามีความสำคัญเป็นอันดับสูงสุด

- 2) รายละเอียด คือ การอธิบายความสัมพันธ์ หรือข้อจำกัดของความต้องการหรือความ
ต้องการที่ได้กล่าวถึง ซึ่งความต้องการแต่ละชุด จะมีรายละเอียดทำหน้าที่อธิบายประกอบ

4.3.9 ผู้เกี่ยวข้องและมุมมองต่างๆ (Stakeholders and Viewpoints)

ผู้เกี่ยวข้องและมุมมองต่างๆ คือ การอธิบายความต้องการที่มาจากมุมมองที่เกี่ยวข้องทั้งหมด ซึ่ง
เกิดขึ้นจากผู้เกี่ยวข้องที่มีบทบาทแตกต่างกัน ทำให้เห็นภาพและความแตกต่างในแนวคิดที่พิจารณาจาก
จุดประสงค์หรือเป้าหมายจากหลายๆ ด้าน และจะสามารถนำไปตัดสินใจโดยอิงกับบทบาทและ
ความสำคัญของผู้เกี่ยวข้องที่เป็นที่มาของความต้องการ

มุมมองในแง่ของความต้องการยังมีความสำคัญในเรื่องบทบาทของผู้เกี่ยวข้อง ผู้กำหนดเงื่อนไขและ
เป้าหมาย หรือผู้มีอำนาจในการตัดสินใจ ซึ่งทำให้เห็นข้อเท็จจริงในด้านอื่นๆ ที่ละเลยไม่ได้ เช่น บทบาทของ
ผู้มีอำนาจในโครงการกำหนดอย่างชัดเจนในด้านเวลาในการทำงานของระบบ แม้จะมีรายละเอียดเรื่อง
อื่นๆ เช่น ด้านความถูกต้องของระบบ จากคนที่มียบทบาทรองลงไป ก็ยังต้องให้ความสำคัญกับด้านเวลา
ของการทำงานเป็นเรื่องสำคัญกว่าองค์ประกอบผู้เกี่ยวข้องและมุมมองต่างๆ เป็นการแสดงความเห็น
พื้นฐานที่มาจากบุคคลหรือกลุ่มบุคคลที่เกี่ยวข้องโดยตรงกับความต้องการ ซึ่งใช้แสดงมุมมองเริ่มต้นที่มี
ต่อความต้องการ โดยมุมมองที่แสดงจะถือเป็นความคิดเห็น ไม่ใช่ข้อสรุปที่ระบบจะต้องเป็นไปตามนั้น
หรืออาจอยู่ในรูปข้อเสนอ ซึ่งรูปแบบขององค์ประกอบผู้เกี่ยวข้องและมุมมองต่างๆ แสดงได้ดังรูปที่ 4.15



รูปที่ 4.15 รูปแบบขององค์ประกอบผู้เกี่ยวข้องและมุมมองต่างๆ ของแบบรูปโดเมนความต้องการ

จากรูปที่ 4.15 องค์ประกอบผู้เกี่ยวข้องและมุมมองต่างๆอาจเป็นการพิจารณาผู้เกี่ยวข้องใดๆ ในระบบ ซึ่งอาจมีหรือไม่มีความเห็นพื้นฐานเกี่ยวกับระบบส่วนนี้ก็ได้ โดยองค์ประกอบผู้เกี่ยวข้องและมุมมองต่างๆ จะทำการพิจารณารายละเอียดตามรูปแบบผู้เกี่ยวข้องซึ่งมีรายละเอียด ดังนี้

- 1) ผู้เกี่ยวข้อง คือ บุคคลหรือกลุ่มบุคคลที่เกี่ยวข้องกับความต้องการ การกล่าวถึงผู้เกี่ยวข้องกระทำโดยอ้างอิงผ่านบทบาทที่มีต่อระบบ ซึ่งผู้เกี่ยวข้องอาจมีความเห็นเกี่ยวกับความต้องการที่แสดงในองค์ประกอบความต้องการได้หลากหลาย โดยแต่ละความเห็นมีรูปแบบที่ประกอบด้วย 2 ส่วน ดังนี้

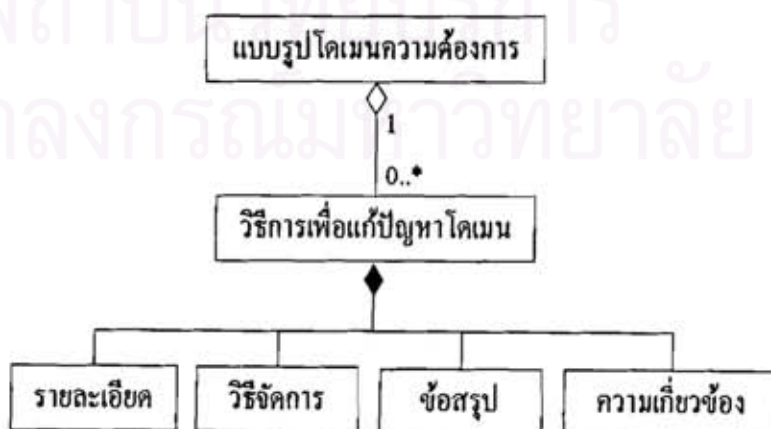
กล่าวถึงความต้องการ เป็นส่วนที่อ้างอิงไปถึงองค์ประกอบความต้องการที่ได้มีการเสนอในแบบรูป

ความเห็นที่มี เป็นส่วนที่ใส่ความเห็นของผู้เกี่ยวข้องที่มีต่อความต้องการตามที่อ้างอิงจากส่วนกล่าวถึงความต้องการ

4.3.10 วิธีการเพื่อแก้ปัญหาโดเมน (Domain Solution)

วิธีการเพื่อแก้ปัญหาโดเมน คือ ผลเฉลยในรูปของการจัดการกับปัญหาที่ผ่านมา โดยเป็นส่วนประกอบที่เกิดจากข้อสรุปของลักษณะความต้องการซอฟต์แวร์และวิธีการจัดการจนได้ผลลัพธ์นั้นมา โดยอยู่ในรูปถึงผลสรุปถึงวิธีการเพื่อแสดงวิธีการแก้ปัญหาที่เกิดขึ้นกับความต้องการในโดเมน ตัวอย่างเช่น การประชุมร่วมหัวหน้าวิศวกร ผู้ใช้งาน และ ทีมพัฒนาระบบ ในการสรุปว่าระบบรถไฟฟ้าได้คืนจะวิ่งด้วยความเร็วสูงสุดไม่เกิน 120 กม.ต่อชม. เป็นต้น

องค์ประกอบวิธีการเพื่อแก้ปัญหาโดเมนเป็นการเสนอผลลัพธ์ผ่านวิธีการที่ประสบผล โดยเป็นวิธีการที่ใช้แก้ปัญหาที่เกี่ยวข้องกับความต้องการที่ทำให้บรรลุผลได้สำเร็จ โดยรูปแบบขององค์ประกอบวิธีการเพื่อแก้ปัญหาโดเมนสามารถแสดงได้ดังรูปที่ 4.16



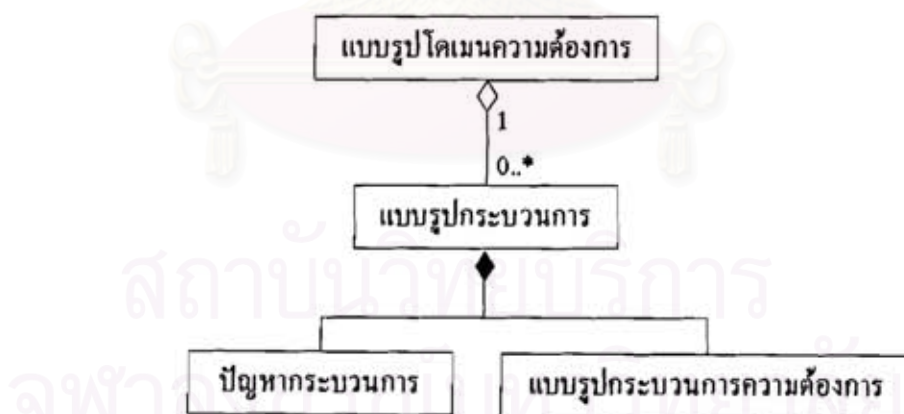
รูปที่ 4.16 รูปแบบองค์ประกอบวิธีการเพื่อแก้ปัญหาโดเมนของแบบรูปโดเมนความต้องการ

จากรูปที่ 4.16 องค์ประกอบวิธีการเพื่อแก้ปัญหาโดเมนประกอบด้วย 4 ส่วน ดังนี้

- 1) รายละเอียด คือ การอธิบายตัวปัญหาที่สืบเนื่องมาจากความต้องการที่ในส่วนนี้จะได้เสนอวิธีการแก้ไขปัญหา
- 2) วิธีจัดการ คือ การกล่าวถึงวิธีการที่เคยนำมาใช้แก้ปัญหา เป็นการบรรยายให้ทราบว่ามีขั้นตอน วิธีการ เทคนิค หรือความเห็นเกี่ยวกับปัญหาที่เกิดขึ้นและวิธีที่จะแก้ไขได้อย่างไร
- 3) ข้อสรุป คือ การสรุปการแก้ปัญหาอย่างสั้นๆ
- 4) ความเกี่ยวข้อง คือ การอธิบายความเกี่ยวข้องของปัญหากับความต้องการที่เป็นที่มาของปัญหา ซึ่งความต้องการที่เกี่ยวข้องอาจหมายถึงความต้องการตัวใดตัวหนึ่งหรือหลายตัวที่มีความเกี่ยวข้องกันก็ได้

4.3.11 แบบรูปกระบวนการ (Process Patterns)

องค์ประกอบแบบรูปกระบวนการ คือ การเสนอปัญหาที่เกิดขึ้นในการเลือกใช้เทคนิคหรือวิธีการกับการกำหนดความต้องการในแบบรูป ซึ่งในส่วนนี้จะแสดงจุดสำคัญของที่มาหรือความจำเป็นในการใช้เทคนิคและวิธีการ โดยเป็นการอ้างอิงไปยังแบบรูปกระบวนการความต้องการผ่านปัญหา ความจำเป็นหรือความเหมาะสม ซึ่งมีรูปแบบแสดงได้ดังรูปที่ 4.17



รูปที่ 4.17 รูปแบบองค์ประกอบแบบรูปกระบวนการของแบบรูปโดเมนความต้องการ

จากรูปที่ 4.17 แสดงให้เห็นว่าในการใช้แบบรูปโดเมนความต้องการอาจมีปัญหาก่อให้เกิดการเจาะจงหรือเกิดความเหมาะสมในการใช้แบบรูปกระบวนการ หากมีลักษณะที่จะใช้แบบรูปกระบวนการความต้องการจะมีรูปแบบการเสนอประกอบด้วย 2 ส่วน ดังนี้

- 1) ปัญหากระบวนการ คือ การอธิบายตัวปัญหาหรือลักษณะที่เจาะจงให้เกิดความจำเป็นหรือความเหมาะสมในการใช้แบบรูปกระบวนการความต้องการ

- 2) แบบรูปกระบวนการความต้องการ คือ การอ้างอิงไปถึงแบบรูปกระบวนการความต้องการที่สอดคล้องหรือเหมาะสมที่จะนำมาใช้ตามลักษณะของปัญหากระบวนการ

4.3.12 ตัวอย่าง

องค์ประกอบตัวอย่างที่ได้กล่าวถึงไปแล้ว เป็นการกล่าวถึงองค์ประกอบตัวอย่างที่อยู่ในแบบรูปทั่วไป สำหรับองค์ประกอบตัวอย่างในแบบรูปโดเมนความต้องการจะนำส่วนของเอกสารข้อกำหนดความต้องการหรือผลลัพธ์ที่คล้ายกันมาแสดงเป็นตัวอย่าง รูปแบบขององค์ประกอบตัวอย่างในแบบรูปโดเมนความต้องการคือการยกตัวอย่างจากส่วนของเอกสารที่เป็นผลลัพธ์ขึ้นมา

4.4 องค์ประกอบของแบบรูปกระบวนการความต้องการ

องค์ประกอบของแบบรูปกระบวนการความต้องการประกอบด้วย องค์ประกอบที่มาจากแบบรูปพื้นฐาน และองค์ประกอบที่เจาะจงสำหรับการแก้ปัญหาของเทคนิคและวิธีการในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ ซึ่งในหัวข้อนี้จะแสดงองค์ประกอบอธิบายความหมาย และแสดงรูปแบบขององค์ประกอบที่เสนอในวิทยานิพนธ์ ซึ่งมีรายละเอียดแยกตามองค์ประกอบดังนี้

4.4.1 จุดประสงค์ (Objective)

จุดประสงค์ใช้อธิบายและช่วยให้เข้าใจจุดประสงค์ในการใช้เทคนิควิธี โดยจุดประสงค์จะสรุปสิ่งที่จะต้องทำ และแสดงลักษณะปัญหาที่จะนำเทคนิควิธีไปใช้ ซึ่งจะสรุปลักษณะของความมุ่งหมายที่เป็นประเด็นไปสู่เทคนิคหรือวิธีที่เหมาะสมถูกต้อง

องค์ประกอบจุดประสงค์ คือ การอธิบายสาเหตุหลักของการใช้เทคนิคหรือวิธีการที่เสนอในแบบรูปกระบวนการความต้องการ ซึ่งองค์ประกอบจุดประสงค์จะเป็นการเสนอมุมมองของปัญหาว่าปัญหาทางด้านกระบวนการวิศวกรรมความต้องการซอฟต์แวร์มีที่มาหรือจุดประสงค์เพื่ออะไร โดยรูปแบบขององค์ประกอบจุดประสงค์แสดงได้ดังรูปที่ 4.18



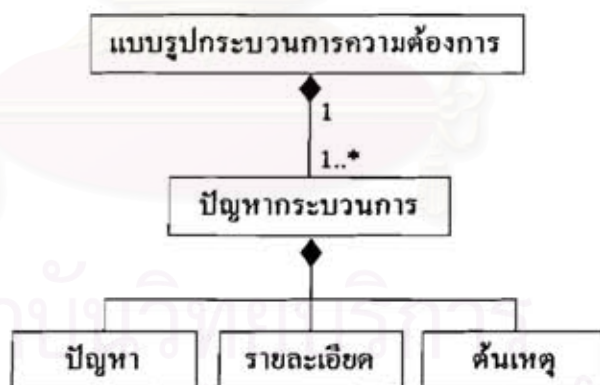
รูปที่ 4.18 รูปแบบองค์ประกอบจุดประสงค์ของแบบรูปกระบวนการความต้องการ

รูปที่ 4.18 แสดงให้เห็นว่าในแบบรูปกระบวนการต้องการหนึ่งๆ จำเป็นต้องมีจุดประสงค์อย่างน้อย 1 ข้อ ซึ่งรูปแบบขององค์ประกอบจุดประสงค์ในแบบรูปเป็นการอธิบายจุดประสงค์ของเทคนิคหรือวิธีการอย่างสั้นๆ ซึ่งหากมีหลายจุดประสงค์ก็จะมีกรอธิบายหลายข้อแยกตามแต่ละจุดประสงค์

4.4.2 ปัญหากระบวนการ (Process Problem)

ปัญหาของวิธีการพัฒนาใช้ขยายความเข้าใจเกี่ยวกับกระบวนการ พฤติกรรม ตลอดจนความสามารถเฉพาะด้านของคนในทีมพัฒนา ที่เป็นปัญหาและข้อจำกัดในการพัฒนา เช่น ในลักษณะพฤติกรรมหรือรูปแบบการทำงานอาจมีความยุ่งยากเมื่อต้องทำงานบางแบบ เนื่องมาจากเป็นงานที่ไม่ได้กำหนดคนและลักษณะงานไว้แต่ต้นด้วยเพราะไม่ได้คาดการณ์ล่วงหน้า

องค์ประกอบปัญหากระบวนการ คือ ส่วนที่แสดงให้เห็นว่าปัญหาที่เกิดขึ้นกับเทคนิคและวิธีการในแบบรูปมีลักษณะอย่างไร ซึ่งในการใช้งานแบบรูปจะนำองค์ประกอบส่วนนี้มาพิจารณาเพื่อดูว่ามีความเหมาะสมที่จะต้องใช้แบบรูปนี้หรือไม่ องค์ประกอบปัญหากระบวนการจะแสดงจุดที่เกิดปัญหาที่จะนำไปสู่การแก้ไขตามแบบรูป ดังนั้นหากปัญหาที่ประสบอยู่มีลักษณะไม่ตรงกับองค์ประกอบปัญหากระบวนการ การใช้แบบรูปกระบวนการความต้องการก็จะไม่สามารถช่วยในส่วนที่ยังคงเป็นปัญหาอยู่ได้ โดยลักษณะขององค์ประกอบปัญหากระบวนการสามารถแสดงได้ดังรูปที่ 4.19



รูปที่ 4.19 รูปแบบองค์ประกอบปัญหากระบวนการของแบบรูปกระบวนการความต้องการ

ในรูปที่ 4.19 ปัญหากระบวนการนั้นเป็นส่วนสำคัญของแบบรูปกระบวนการความต้องการ การเสนอแบบรูปจะชี้ไปยังตัวปัญหาซึ่งก็คือส่วนขององค์ประกอบปัญหากระบวนการ ดังนั้นแบบรูปกระบวนการจึงต้องมีการระบุปัญหากระบวนการอย่างน้อย 1 ปัญหา โดยองค์ประกอบปัญหากระบวนการประกอบด้วย 3 ส่วน คือ

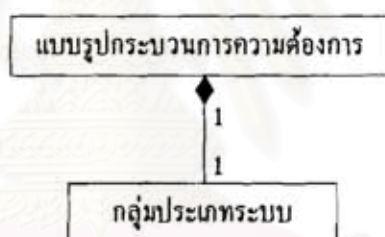
- 1) ปัญหา คือ การเรียกปัญหาอย่างสั้นหรือเป็นชื่อที่ใช้เรียกหรือสื่อไปถึงปัญหานั้น
- 2) รายละเอียด คือ ส่วนที่อธิบายลักษณะของปัญหาเกี่ยวกับเทคนิคและวิธีโดยละเอียด
- 3) ต้นเหตุ คือ การสรุปเกี่ยวกับที่มาที่แท้จริงของตัวปัญหา

4.4.3 กลุ่มประเภทของระบบ (Application Area)

กลุ่มประเภทของระบบ คือ ลักษณะขอบเขตทางสถานะของระบบที่มีความคล้ายคลึงกัน ซึ่งส่งผลให้ระบบที่อยู่ในกลุ่มมีความพิเศษเฉพาะทั้งที่เป็นข้อกำหนด ข้อห้าม ซึ่งจะเป็นส่วนหนึ่งที่กำหนดลักษณะที่มีผลโดยตรงต่อกระบวนการในการพัฒนา

กลุ่มระบบบางประเภทยังมีข้อกำหนดเฉพาะที่เจาะจงไปยังกระบวนการหรือวิธีการที่จะนำมาใช้ ทำให้การพัฒนาสำหรับกลุ่มระบบประเภทนั้นจะต้องมีรายละเอียดบางประเด็นเพิ่มขึ้น ในทางกลับกันบางกลุ่มประเภทของระบบก็สามารถละเลยบางประเด็นออกไปทำให้การพัฒนาทำได้ง่ายขึ้น

องค์ประกอบกลุ่มประเภทของระบบ คือ ส่วนที่กำหนดขอบเขตที่จะมีการใช้แบบรูปกระบวนการความต้องการ โดยจะหมายถึงลักษณะของระบบที่มีการนำเทคนิคหรือวิธีการที่เสนอไปใช้ โดยแบบรูปกระบวนการความต้องการต้องแสดงให้เห็นว่าลักษณะระบบที่เหมาะสมกับเทคนิคหรือวิธีการที่แบบรูปเสนอคือลักษณะใด โดยที่รูปแบบที่เสนอขององค์ประกอบกลุ่มประเภทระบบจะเป็นการบรรยายลักษณะระบบที่มีความเหมาะสม ในลักษณะการแจกแจงและให้ข้อสังเกต ดังแสดงได้ตามรูปที่ 4.20



รูปที่ 4.20 รูปแบบองค์ประกอบกลุ่มประเภทของระบบของแบบรูปกระบวนการความต้องการ

แม้ว่าเทคนิคหรือวิธีการหนึ่งๆ อาจมีความเหมาะสมกับระบบหลายประเภท แต่ในองค์ประกอบกลุ่มประเภทรระบบจะอ้างถึงลักษณะโดยรวมที่สำคัญ โดยถือว่ามิลักษณะระบบแบบเดี่ยวเท่านั้นที่มีความเหมาะสมและถือเป็นขอบเขตในการเลือกใช้แบบรูป ดังนั้นกลุ่มประเภทของระบบจึงเป็นลักษณะที่กว้างและไม่เจาะจงหรือจำแนกประเภทตามลักษณะประเภทรระบบที่นิยมกัน

4.4.4 สถานการณ์ (Situation)

สถานการณ์ คือ เหตุการณ์ที่เกิดขึ้นระหว่างกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ ซึ่งส่งผลทั้งทางตรงและทางอ้อมต่อผลลัพธ์และพฤติกรรมที่จะได้รับ โดยเหตุการณ์เหล่านี้จะเป็นตัวสร้างข้อจำกัดและชี้นำผลลัพธ์ของระบบ ทำให้ผลลัพธ์ที่ได้อาจไม่ปฏิบัติตามจุดประสงค์ แต่เมื่อสถานการณ์มีลักษณะชัดเจนก็จะสามารถกำหนดกรอบของเทคนิควิธีที่จะนำมาใช้ได้ง่ายขึ้น

องค์ประกอบสถานการณ์ คือ ส่วนของแบบรูปกระบวนการความต้องการที่แสดงลักษณะแวดล้อมที่เหมาะสมในการใช้แบบรูป ซึ่งในแง่ของการใช้เทคนิคหรือวิธีการลักษณะสภาพแวดล้อมก็คือ

สถานการณ์นั่นเอง โดยสถานการณ์จะอธิบายถึงลักษณะโดยรวมของการทำงาน ลักษณะความเกี่ยวข้องกับกระบวนการวิศวกรรมความต้องการซอฟต์แวร์

ลักษณะองค์ประกอบสถานการณ์สามารถแสดงได้ดังรูปที่ 4.21 ซึ่งหมายความว่าแบบรูปกระบวนการหนึ่งๆ จะต้องระบุสถานการณ์ที่จะมีการใช้แบบรูปอย่างน้อยหนึ่งอย่าง แต่ก็อาจมีมากกว่าหนึ่งสถานการณ์ที่มีความเหมาะสมในการใช้แบบรูป โดยรูปแบบของสถานการณ์ที่แสดงจะเป็นการบรรยายสภาพของสถานการณ์นั้นๆ

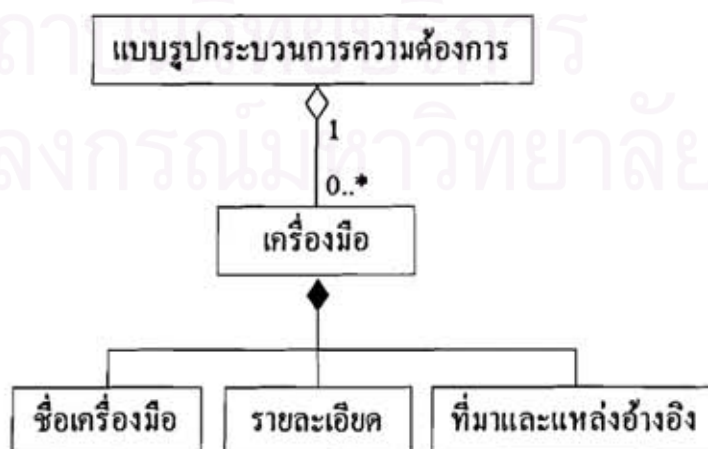


รูปที่ 4.21 รูปแบบองค์ประกอบสถานการณ์ของแบบรูปกระบวนการความต้องการ

4.4.5 เครื่องมือ (Tools)

เครื่องมือ คือ สิ่งที่จะช่วยในการทำงานในแบบรูปนี้ได้ ทั้งที่เป็นรูปธรรม เช่น ซอฟต์แวร์ที่ใช้ในการทำงาน หรือเป็นนามธรรม เช่น วิธีการที่จะช่วยสนับสนุนหรือหาผลลัพธ์ทางอ้อม เป็นต้น โดยผลที่ได้จากการใช้เครื่องมือจะต้องไม่เปลี่ยนแปลงจุดประสงค์ที่ต้องการ ไป

องค์ประกอบเครื่องมือ คือ ส่วนของแบบรูปกระบวนการความต้องการที่เสนอเครื่องมือที่สามารถนำมาใช้ได้เหมาะสม โดยหมายรวมไปถึงเทคนิคและวิธีการต่างๆ ตลอดจนเครื่องมือที่น่าจะนำมาใช้ประกอบการทำงานได้ โดยลักษณะรูปแบบขององค์ประกอบเครื่องมือสามารถแสดงได้ดังรูปที่ 4.22



รูปที่ 4.22 รูปแบบองค์ประกอบเครื่องมือของแบบรูปกระบวนการความต้องการ

จากรูปที่ 4.22 องค์ประกอบเครื่องมือเป็นองค์ประกอบเสริมในแบบรูปกระบวนการความต้องการ ซึ่งในเทคนิคและวิธีการที่เสนออาจไม่มีเครื่องมือที่เหมาะสมหรือสามารถนำมาใช้ประกอบ และมีความเป็นไปได้ที่จะมีเครื่องมือที่สนับสนุนอยู่หลายตัว โดยองค์ประกอบเครื่องมือประกอบด้วย 3 ส่วน คือ

- 1) ชื่อเครื่องมือ คือ ชื่อที่ใช้เรียกเครื่องมือตัวนั้น
- 2) รายละเอียด คือ การอธิบายและให้รายละเอียดเกี่ยวกับเครื่องมือโดยคร่าวๆ
- 3) ที่มาและแหล่งอ้างอิง คือ แหล่งอ้างอิง แหล่งที่มีข้อมูลเกี่ยวกับเครื่องมือ หรือแหล่งที่สามารถไปนำเครื่องมือนั้นมาใช้ได้

4.4.6 วิธีการ (Method)

วิธีการ คือ ขั้นตอนวิธีหรือหลักการของวิธีในแบบรูป รวมไปถึงเทคนิคและวิธีการปฏิบัติที่นำมาใช้ โดยกำหนดในลักษณะของสิ่งที่ต้องปฏิบัติเป็นลำดับหรือประกอบกันจนวิธีการนั้นเสร็จสมบูรณ์ในการใช้งาน

องค์ประกอบวิธีการ คือ ส่วนที่อธิบายการใช้งานหรือลำดับการทำงานสำหรับเทคนิคหรือวิธีการที่เสนอในแบบรูป องค์ประกอบวิธีการจึงถือเป็นส่วนสำคัญในแบบรูปกระบวนการความต้องการ ในส่วนของวิธีการนั้นต้องเสนอขั้นตอนที่มีความเหมาะสมและชัดเจนในการนำไปใช้งาน โดยเสนอในรูปแบบลักษณะขั้นตอนการทำงาน ดังรูปที่ 4.23



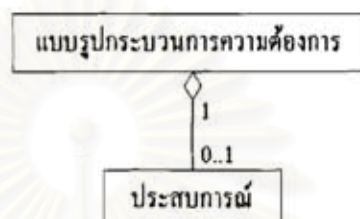
รูปที่ 4.23 รูปแบบองค์ประกอบวิธีการของแบบรูปกระบวนการความต้องการ

ในรูปที่ 4.23 องค์ประกอบวิธีการที่อยู่ในแบบรูปกระบวนการความต้องการมีเพียงแต่วิธีการเดียวเท่านั้นที่ถูกนำเสนอ ซึ่งจะเสนอวิธีการ โดยเรียงลำดับตามขั้นตอนการทำงาน ซึ่งรูปแบบของวิธีการจะเป็นการอธิบายขั้นตอนการทำงานทีละลำดับขั้น หรืออาจแสดงเพียงส่วนเดียวเพื่ออธิบายวิธีการ โดยสรุป

4.4.7 ประสบการณ์ (Experience)

ประสบการณ์ คือ สิ่งที่ได้จากการทำงานโดยวิธีเดียวกับแบบรูป ซึ่งจะเป็นการอธิบายประสบการณ์ในการแก้ปัญหา ประสบการณ์ในการเลือกเครื่องมือหรือเทคนิค รวมไปถึงประสบการณ์ที่ได้จากกรณีที่ใช้วิธีการไม่บรรลุผล

องค์ประกอบประสบการณ์ คือ ส่วนที่แสดงความรู้หรือประสบการณ์จากการทำงานกับเทคนิคหรือวิธีการ ในแบบที่แบบรูปเสนอ ซึ่งเป็นการบันทึกเกร็ดความรู้ หรือข้อสรุปที่น่าสนใจเกี่ยวกับการทำงาน ในส่วนขององค์ประกอบประสบการณ์เป็นส่วนที่ช่วยเพิ่มเติมความรู้ในเทคนิคหรือวิธีการที่แบบรูปกล่าวไว้ แต่ก็ไม่ใช่ส่วนที่มีความจำเป็น ซึ่งแสดงได้ดังรูปที่ 4.24



รูปที่ 4.24 รูปแบบองค์ประกอบประสบการณ์ของแบบรูปกระบวนการความต้องการ

ในรูปที่ 4.24 องค์ประกอบประสบการณ์อาจไม่มีระบุในแบบรูปกระบวนการความต้องการ หรือมีอยู่หลายอันก็ได้ โดยรูปแบบขององค์ประกอบประสบการณ์จะเป็นการอธิบายหรือแสดงความรู้ ข้อสังเกต ข้อสรุป หรือรายละเอียดอื่นๆ ที่มีประโยชน์ซึ่งได้มาจากประสบการณ์ในการทำงาน โดยเป็นการถ่ายทอดประสบการณ์ตามเรื่องราวที่ได้ประสบมา

4.4.8 ค่าใช้จ่าย (Cost)

ค่าใช้จ่าย คือ ปัจจัยที่ต้องใช้ในการทำงานด้วยวิธีในแบบรูป ซึ่งนอกจากค่าใช้จ่ายที่มองว่าเป็นมูลค่าที่เป็นตัวเงินยังรวมไปถึงค่าใช้จ่ายทางด้านเวลา ค่าใช้จ่ายด้านความรู้ความสามารถและค่าใช้จ่ายของการลงแรงในการทำงานด้วย

องค์ประกอบค่าใช้จ่าย คือ การแสดงต้นทุนของการใช้เทคนิคหรือวิธีการที่แบบรูปกระบวนการความต้องการได้เสนอ ซึ่งข้อมูลส่วนนี้ผู้ใช้แบบรูปจะสามารถนำไปใช้พิจารณาความเหมาะสม ความคุ้มค่าในการทำตามเทคนิคหรือวิธีการ ซึ่งในแบบรูปกระบวนการความต้องการจำเป็นต้องมีการระบุค่าใช้จ่ายให้ชัดเจนซึ่งจะขาดไม่ได้ ลักษณะขององค์ประกอบค่าใช้จ่ายสามารถแสดงได้ดังรูปที่ 4.25



รูปที่ 4.25 รูปแบบองค์ประกอบค่าใช้จ่ายของแบบรูปกระบวนการความต้องการ

รูปที่ 4.25 เป็นการแสดงส่วนสำคัญ 4 ส่วนที่รวมกันเป็นค่าใช้จ่าย ซึ่งค่าใช้จ่ายที่เกิดขึ้นอาจมาจากส่วนใดส่วนหนึ่งเป็นหลักหรือทั้ง 4 ส่วนร่วมกัน โดยมีรายละเอียด ดังนี้

- 1) **เชิงมูลค่า** คือ การแสดงค่าใช้จ่ายในแง่ของมูลค่าเงินที่ต้องนำมาใช้หรือเสียไปกับเทคนิคหรือวิธีการในรูปแบบ
- 2) **เชิงกำลังคน** คือ การแสดงกำลังคนที่ต้องลงแรงกับเทคนิคหรือวิธีการในรูปแบบ
- 3) **เชิงความรู้** คือ การแสดงความรู้ที่จำเป็นซึ่งอาจต้องศึกษา จัดหา หรือค้นคว้าตามมา
- 4) **เชิงเวลา** คือ การแสดงเวลาที่จะต้องใช้เพิ่มขึ้นหากมีการนำเทคนิคหรือวิธีการนี้มาใช้

การแสดงค่าใช้จ่ายไม่ว่าจะเป็นเชิงมูลค่า เชิงกำลังคน เชิงความรู้ หรือเชิงเวลา จะไม่แสดงในเชิงตัวเลขเนื่องจากค่าใช้จ่ายจะขึ้นอยู่กับปัจจัยหลายด้านที่กำหนด การแสดงผลเชิงตัวเลขจึงอาจไม่เป็นจริงหรือเท่ากันในทุกกรณี ดังนั้นการแสดงค่าใช้จ่ายจะเป็นการแสดงผลโดยสรุปจากการให้ข้อสังเกตกับต้นทุนที่มีหรือเกิดขึ้นในแต่ละด้าน

4.4.9 ตัวอย่าง

องค์ประกอบตัวอย่างสำหรับแบบรูปกระบวนการความต้องการ จะเป็นการแสดงตัวอย่างการใช้เทคนิคและวิธีที่แบบรูปได้เสนอ ซึ่งลักษณะที่เป็นไปได้ของตัวอย่างก็คือการขกเหตุการณ์ในการทำงานขึ้นมา รูปแบบขององค์ประกอบตัวอย่างในแบบรูปกระบวนการจึงเป็นการบรรยายเหตุการณ์ที่มีการใช้เทคนิคหรือวิธีการ

4.5 องค์ประกอบของแบบรูปความต้องการส่วนความปลอดภัย

องค์ประกอบของแบบรูปความต้องการส่วนความปลอดภัยมีองค์ประกอบเช่นเดียวกับกับแบบรูปโดเมนความต้องการ แต่มีส่วนที่เกี่ยวข้องกับความปลอดภัยเพิ่มเข้ามา ในหัวข้อนี้จึงเป็นการเสนอองค์ประกอบในส่วนที่เพิ่มขึ้นในแบบรูปความต้องการส่วนความปลอดภัย ซึ่งประกอบด้วยองค์ประกอบต่างๆ ดังนี้

4.5.1 การยอมรับอันตราย (Safety Acceptance)

การยอมรับอันตราย คือ เงื่อนไขในสภาพแวดล้อมที่กำหนดขึ้นจากเงื่อนไขหลักของโครงการ เพื่อใช้เป็นข้อกำหนดในส่วนของความปลอดภัย โดยเป็นการระบุระดับการยอมรับความผิดพลาดของระบบ

องค์ประกอบของการยอมรับอันตราย คือ ส่วนที่แสดงสภาพแวดล้อมอย่างหนึ่งของปัญหา เป็นการอธิบายข้อจำกัดในเรื่องการยอมรับผลของอันตรายที่อาจเกิดขึ้นกับระบบ ซึ่งการยอมรับอันตรายจะต้องมี

รายละเอียดแสดงการรับรองหรือการรับผิดชอบต่ออันตรายที่เกิดขึ้น โดยมีรูปแบบขององค์ประกอบการยอมรับอันตรายดังรูปที่ 5.26



รูปที่ 4.26 รูปแบบองค์ประกอบการยอมรับอันตรายของแบบรูปกระบวนการความต้องการ

รูปที่ 4.26 แสดงส่วนประกอบที่อยู่ในองค์ประกอบการยอมรับอันตราย ซึ่งในแบบรูปความต้องการส่วนความปลอดภัยอาจมีการประเด็นความปลอดภัยที่ต้องกำหนดหลักเกณฑ์ในการยอมรับหลายส่วน ซึ่งการยอมรับอันตรายจะประกอบด้วย 3 ส่วน คือ

- 1) รายละเอียด คือ การอธิบายลักษณะอันตรายและเงื่อนไขการยอมรับอันตรายที่เกิดขึ้น
- 2) ระดับความอันตราย คือ การระบุระดับความอันตราย ซึ่งประเมินเป็น 4 ระดับ ดังนี้
 - 1) ต่ำ
 - 2) ปานกลาง
 - 3) สูง
 - 4) สำคัญยิ่ง
- 3) ระดับการจัดการ คือ การระบุระดับการจัดการและรับผิดชอบในอันตรายที่กล่าวถึง

4.5.2 ความต้องการความปลอดภัย (Safety Requirements)

ความต้องการความปลอดภัย คือ ความต้องการที่เน้นในส่วนของคุณภาพความปลอดภัย โดยอาจเห็นว่าซ้ำซ้อนกับองค์ประกอบความต้องการซึ่งนำเสนอในแบบรูปสำหรับปัญหาโดเมน แต่เนื่องจากความต้องการเหล่านี้มีความชัดเจนและมีลักษณะปลีกย่อยที่ควรแสดงรายละเอียดที่จำเป็นเพื่อใช้ในการเป็นผลเฉลย ดังนั้นจึงแยกองค์ประกอบส่วนนี้ออกมาเป็นอีกส่วนหนึ่ง

ความต้องการความปลอดภัยจะแสดงรายละเอียดของความต้องการที่เกี่ยวข้องกับความปลอดภัยและประเภทของความต้องการ เช่น ความต้องการในการป้องกัน ความต้องการในการตอบสนอง เป็นต้น

โดยลักษณะเหล่านี้จะแสดงลักษณะความต้องการที่สามารถนำไปใช้กำหนดระบบได้อย่างชัดเจนและมีประสิทธิภาพ

องค์ประกอบความต้องการความปลอดภัย คือ ส่วนของผลเฉลยในแบบรูปความต้องการส่วนความปลอดภัย ซึ่งอธิบายความต้องการความปลอดภัยที่เกี่ยวข้องกับแบบรูป ความต้องการความปลอดภัย เป็นการกำหนดปัจจัยหรือคุณสมบัติของระบบว่าจะต้องทำอะไรเพื่อให้เกิดความปลอดภัยตามลักษณะความเสี่ยงหรืออันตรายที่มีโอกาสเกิดขึ้นสูง ซึ่งมีรายละเอียดขององค์ประกอบความต้องการความปลอดภัยดังรูปที่ 4.27



รูปที่ 4.27 รูปแบบองค์ประกอบความต้องการความปลอดภัยของแบบรูปกระบวนการความต้องการ

รูปที่ 4.27 แสดงให้เห็นว่าองค์ประกอบความต้องการความปลอดภัยเป็นส่วนสำคัญของแบบรูปความต้องการส่วนความปลอดภัย โดยแบบรูปต้องมีส่วนองค์ประกอบความต้องการความปลอดภัยอยู่ด้วยเสมอ และอาจมีความต้องการความปลอดภัยที่เกี่ยวข้องอยู่มากกว่าหนึ่งความต้องการ โดยในแต่ละความต้องการขององค์ประกอบความต้องการที่เกี่ยวข้องกับความปลอดภัยจะประกอบด้วย 2 ส่วน ดังนี้

- 1) ประเภท คือ การระบุประเภทของความต้องการความปลอดภัย ซึ่งมี 3 ประเภท ดังนี้

การป้องกัน คือ ความต้องการที่จะต้องป้องกันไม่ให้มีความเสี่ยงหรืออันตรายที่ระบุเกิดขึ้น

การตรวจสอบ คือ ความต้องการในการตรวจสอบหรือค้นหาว่ามีความเสี่ยงหรืออันตรายที่ระบุเกิดขึ้นหรือไม่ เพื่อสามารถดำเนินการต่อได้อย่างถูกต้อง

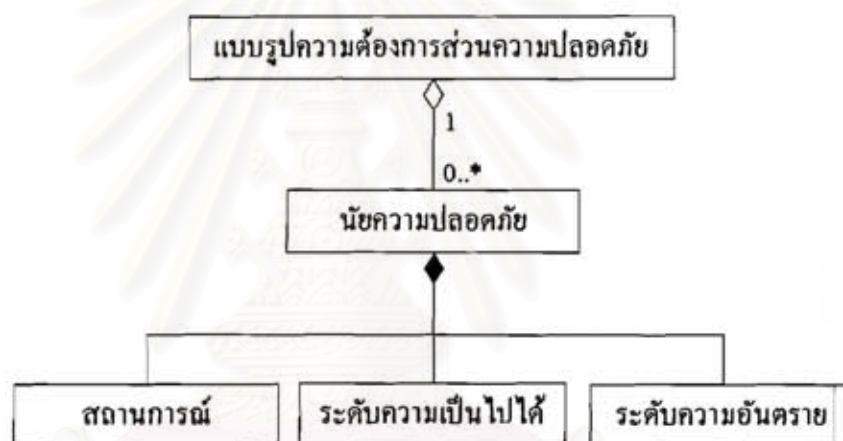
การตอบสนอง คือ ความต้องการในการจัดการหรือแก้ไขปัญหา เมื่อความเสี่ยงหรืออันตรายที่ระบุเกิดขึ้น

- 2) รายละเอียด คือ ส่วนที่อธิบายรายละเอียดของความต้องการที่จะทำการป้องกัน ตรวจสอบ หรือตอบสนอง

4.5.3 นัยความปลอดภัย (Safety Significant)

นัยความปลอดภัย คือ ความต้องการที่ต้องจัดระดับของความปลอดภัย ซึ่งขึ้นอยู่กับความสัมพันธ์ของลักษณะความน่าจะเป็นและความร้ายแรงของเหตุการณ์ โดยนัยความปลอดภัยเป็นความต้องการที่ไม่สามารถกำหนดเป็นรายละเอียดที่ชัดเจนได้ และอาจไม่นำไปเป็นเงื่อนไขของการพัฒนาเนื่องจากความต้องการลักษณะนี้ให้ความสำคัญกับนัยที่อาจเปลี่ยนแปลงได้

องค์ประกอบนัยความปลอดภัย คือ ส่วนหนึ่งของส่วนผลเฉลยในแบบรูป และเป็นลักษณะของความต้องการที่เกี่ยวข้องกับความปลอดภัย โดยเป็นการเสนอความต้องการที่ขึ้นอยู่กับความรุนแรงและความเป็นไปได้ที่จะเกิดขึ้น โดยมีรูปแบบดังรูปที่ 4.28



รูปที่ 4.28 รูปแบบขององค์ประกอบนัยความปลอดภัยของแบบรูปกระบวนการความต้องการ

รูปที่ 4.28 เป็นองค์ประกอบนัยความปลอดภัย ซึ่งในแบบรูปอาจไม่มีความต้องการนัยปลอดภัยที่เกี่ยวข้องอยู่หรือมีอยู่หลายตัวก็ได้ โดยองค์ประกอบนัยความปลอดภัยประกอบด้วย 3 ส่วน ดังนี้

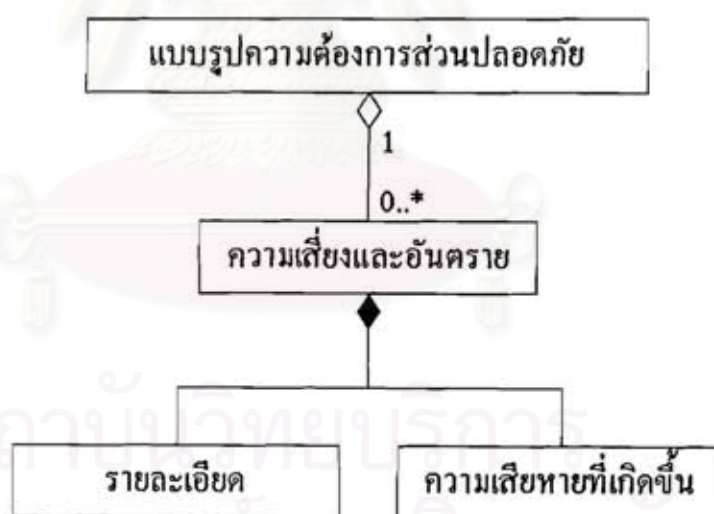
- 1) สถานการณ์ คือ การอธิบายลักษณะอันตรายและเงื่อนไขการยอมรับอันตรายที่เกิดขึ้น
- 2) ระดับความเป็นไปได้ คือ การระบุระดับความอันตราย ซึ่งแบ่งเป็น 5 ระดับ ดังนี้
 - 1) เป็นประจำ
 - 2) มีความเป็นไปได้
 - 3) เป็นบางโอกาส
 - 4) นานๆ ครั้ง
 - 5) เป็นไปได้น้อย
- 3) ระดับความอันตราย คือ การระบุระดับการจัดการและรับผิดชอบในอันตราย ซึ่งแบ่งเป็น 4 ระดับ ดังนี้
 - 1) ภายนอก

- 2) ช่วงอันตราย
- 3) เล็กน้อย
- 4) ไม่สำคัญ

4.5.4 ความเสี่ยงและอันตราย (Risk and Hazard)

ความเสี่ยงและอันตราย คือ การแสดงลักษณะความไม่ปลอดภัยที่สามารถเกิดขึ้น ทั้งผลลัพธ์ที่เกิดขึ้นอย่างแน่นอนและที่อาจเกิดขึ้นจากความผิดพลาดของระบบหรือการทำงาน ความเสี่ยงจากปัจจัยต่างๆต้องนำมาวิเคราะห์หาสาเหตุและกำหนดความต้องการที่สัมพันธ์เพื่อป้องกันอันตราย ลักษณะของอันตรายที่สรุปสามารถนำมาใช้วิเคราะห์อันตรายโดยการค้นหาย้อนกลับ (Backward Search) [10]

องค์ประกอบความเสี่ยงและอันตราย คือ ส่วนที่อธิบายความเสี่ยงหรืออันตรายที่สามารถเกิดขึ้นได้ เป็นองค์ประกอบในแบบรูปความต้องการส่วนความปลอดภัยที่ช่วยให้การวิเคราะห์อันตรายสามารถทำได้ง่าย ความเสี่ยงและอันตรายที่แสดงในแบบรูปจะกล่าวถึงความเสี่ยงและอันตรายที่เกิดขึ้นเฉพาะกับลักษณะในแบบรูปเท่านั้น โดยองค์ประกอบความเสี่ยงและอันตรายแสดงได้ดังรูปที่ 4.29



รูปที่ 4.29 รูปแบบขององค์ประกอบความเสี่ยงและอันตรายของแบบรูปกระบวนการความต้องการ

รูปที่ 4.29 แสดงองค์ประกอบความเสี่ยงและอันตราย ซึ่งในแบบรูปความต้องการส่วนความปลอดภัยจะระบุความเสี่ยงและอันตรายเฉพาะที่สำคัญเท่านั้น โดยรูปแบบขององค์ประกอบความเสี่ยงและอันตรายประกอบด้วย 2 ส่วน ดังนี้

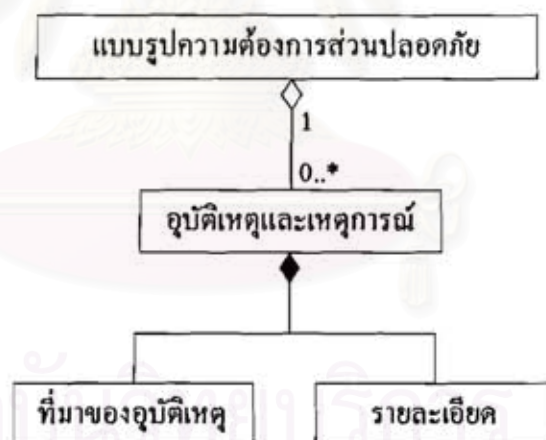
- 1) รายละเอียด คือ ส่วนที่กล่าวเกี่ยวกับรายละเอียดของความเสี่ยงและอันตรายที่มาหรือต้นเหตุโดยคร่าว
- 2) ความเสียหายที่เกิดขึ้น คือ ส่วนที่อธิบายความเสียหายที่เกิดจากประเด็นที่กล่าวถึง

4.5.5 อุบัติเหตุและเหตุการณ์ (Accident and Incident)

อุบัติเหตุและเหตุการณ์ คือ การบกพร่องที่เกิดขึ้นในกรณีที่สุดิวสัย หรือเหตุการณ์ในกรณีที่ใกล้เคียงจะก่อให้เกิดอุบัติเหตุ ซึ่งนำไปสู่ผลเสียต่อชีวิตหรือผลต่อความปลอดภัยของระบบโดยตรง โดยอุบัติเหตุและเหตุการณ์ถือเป็นต้นเหตุที่นำไปสู่การทำงานที่ผิดพลาด สำหรับอุบัติเหตุสามารถแบ่งได้เป็น 3 กลุ่ม คือ อุบัติเหตุจากความผิดพลาดของซอฟต์แวร์ อุบัติเหตุจากความผิดพลาดของฮาร์ดแวร์ และอุบัติเหตุจากการใช้งานของผู้ใช้ระบบ

การทราบรายละเอียดของอุบัติเหตุที่อาจนำไปสู่ความผิดพลาดทำให้ระบบไม่ปลอดภัยจะช่วยให้วิศวกรมีความต้องการซอฟต์แวร์สามารถวิเคราะห์ผลลัพธ์ที่อาจเกิดตามมา และได้ข้อสังเกตเกี่ยวกับลักษณะที่เป็นสาเหตุของความผิดพลาดของการทำงาน สามารถนำมาใช้วิเคราะห์อันตรายโดยการค้นหาไปข้างหน้า (Forward Search) [10]

องค์ประกอบอุบัติเหตุและเหตุการณ์ คือ ส่วนที่อธิบายต้นเหตุที่นำไปสู่อันตราย ซึ่งเป็นอีกส่วนหนึ่งที่ช่วยในการวิเคราะห์อันตราย ในแบบรูปความต้องการส่วนความปลอดภัยจะระบุอุบัติเหตุต่างๆที่เกิดขึ้นได้ โดยองค์ประกอบอุบัติเหตุและเหตุการณ์สามารถแสดงได้ดังรูปที่ 4.30



รูปที่ 4.30 รูปแบบองค์ประกอบอุบัติเหตุของแบบรูปกระบวนการความต้องการ

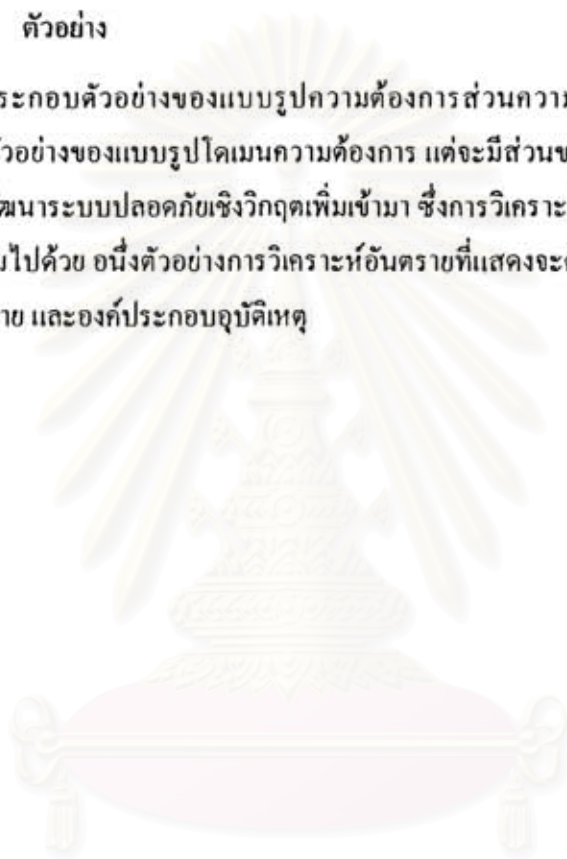
รูปที่ 4.30 แสดงองค์ประกอบอุบัติเหตุและเหตุการณ์ซึ่งเป็นส่วนหนึ่งของแบบรูปความต้องการส่วนความปลอดภัย ซึ่งในแบบรูปอาจประกอบด้วยอุบัติเหตุและเหตุการณ์หลายตัว โดยแต่ละตัวจะมีรูปแบบเป็น 2 ส่วน ดังนี้

- 1) ที่มาของอุบัติเหตุ เป็นลักษณะที่มาหรือต้นเหตุของอุบัติเหตุซึ่งจำแนกที่มาได้ 3 ประเภท ดังนี้
 - 1) ความผิดพลาดของซอฟต์แวร์
 - 2) ความผิดพลาดของฮาร์ดแวร์

- 3) ความผิดพลาดจากผู้ใช้งาน
- 2) รายละเอียด เป็นส่วนที่อธิบายลักษณะของอุบัติเหตุที่เกิดขึ้น และความเป็นไปได้ของอันตรายที่ตามมา

4.5.6 ตัวอย่าง

องค์ประกอบตัวอย่างของแบบรูปความต้องการส่วนความปลอดภัยจะมีลักษณะเช่นเดียวกับองค์ประกอบตัวอย่างของแบบรูปโดเมนความต้องการ แต่จะมีส่วนของการวิเคราะห์อันตรายซึ่งเป็นส่วนสำคัญในการพัฒนาระบบปลอดภัยเชิงวิกฤตเพิ่มเข้ามา ซึ่งการวิเคราะห์อันตรายมีด้วยกันหลายวิธี รูปแบบที่ใช้ก็มีมากตามไปด้วย หนึ่งในตัวอย่างการวิเคราะห์อันตรายที่แสดงจะต้องสอดคล้องกับองค์ประกอบความเสี่ยงและอันตราย และองค์ประกอบอุบัติเหตุ



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

การสร้างแบบรูปความต้องการ

วิทยานิพนธ์นี้ได้เสนอโครงสร้างและองค์ประกอบสำหรับแบบรูปความต้องการ ซึ่งในบทที่ 4 ได้เสนอรายละเอียดเหล่านี้ไปแล้ว แต่ยังไม่ได้มีการนำมาใช้งานหรือสร้างเป็นแบบรูปขึ้น ดังนั้นในบทนี้จึงจะทำการสร้างแบบรูปความต้องการทั้ง 3 ประเภทขึ้น ซึ่งแบบรูปเหล่านี้ยังจะได้นำไปใช้ในส่วนของการทดลองในบทที่ 8 อีกด้วย

5.1 แบบรูปความต้องการที่สร้างขึ้น

ในบทนี้จะสร้างแบบรูปความต้องการทั้ง 3 ประเภท โดยแบบรูปความต้องการที่สร้างขึ้นจะอาศัยการศึกษาและทำความเข้าใจเพื่อนำประสบการณ์ที่ได้มาใช้ในการสร้างแบบรูปความต้องการ ซึ่งแบบรูปความต้องการที่สร้างขึ้นมีดังต่อไปนี้

5.1.1 แบบรูปโดเมนความต้องการ

แบบรูปโดเมนความต้องการที่สร้างขึ้นได้จากระบบการจำหน่ายตั๋วโดยสารรถไฟ ซึ่งเป็นระบบงานที่ใช้จำหน่ายตั๋วโดยสารตามสถานีรถไฟเพื่อใช้ในการเดินทางระยะสั้นๆ ในตัวเมือง มีสถานีโดยสารและเส้นทางไม่มากและซับซ้อนนัก โดยใช้ชื่อแบบรูปที่สร้างขึ้นว่า “ระบบงานจำหน่ายตั๋วโดยสารที่สถานีโดยสาร”

5.1.2 แบบรูปกระบวนการความต้องการ

แบบรูปกระบวนการความต้องการที่สร้างขึ้น เป็นการพิจารณากระบวนการในส่วนของการกำหนดสภาพแวดล้อม (Define Environments) ซึ่งเป็นส่วนหนึ่งในเทคนิคที่นำมาใช้ในการเก็บความต้องการ ซึ่งจะกำหนดสภาพแวดล้อมให้ชัดเจนเพื่อจะเก็บความต้องการที่ถูกต้องและเป็นไปตามที่กำหนดได้ โดยแบบรูปกระบวนการความต้องการที่สร้างขึ้นใช้ชื่อแบบรูปว่า “การกำหนดสภาพแวดล้อม (Elicitation: Define Environments)”

5.1.3 แบบรูปความต้องการส่วนความปลอดภัย

แบบรูปความต้องการส่วนความปลอดภัยที่สร้างขึ้น ได้มาจากการศึกษาระบบเดินรถอัตโนมัติ (Automatic Train Protection: ATP) โดยพิจารณาการทำงานบางส่วนในระบบ และได้สร้างเป็นแบบรูปความต้องการส่วนความปลอดภัย 3 แบบรูป ซึ่งแบบรูปทั้ง 3 คือ

- 1) “การเว้นระยะห่างระหว่างรถไฟที่ปลอดภัย (ATP: Safety Train separation)” ซึ่งเป็นแบบรูปที่กล่าวเกี่ยวกับการเว้นระยะระหว่างรถไฟแต่ละขบวน
- 2) “การควบคุมการเปิดประตูรถไฟ (ATP: Door Opening Control Protection Interlocks)” ซึ่งเป็นแบบรูปที่กล่าวเกี่ยวกับการควบคุมประตูรถไฟ
- 3) “การกำหนดตำแหน่งและความเร็วของรถไฟ (ATP: Train Location and Speed Determination)” ซึ่งเป็นแบบรูปที่กล่าวเกี่ยวกับการกำหนดตำแหน่งและความเร็วของรถไฟ

5.2 แบบรูประบบงานจำหน่ายตั๋วโดยสารที่สถานีโดยสาร

เป็นระบบจำหน่ายตั๋วโดยสาร โดยเน้นไปที่การจำหน่ายตั๋วแก่ผู้ที่จะเดินทางในขณะนั้น และให้ความสำคัญกับการจำหน่ายตั๋วแบบล่วงหน้าหรือขายแบบชุดรองลงไป การจำหน่ายตั๋วโดยสารจะให้ความสำคัญในแง่ของประสิทธิภาพเพื่อให้ทันต่อความต้องการในการเดินทางในช่วงเวลาหนึ่งๆ ทั้งนี้ข้อมูลการจำหน่ายตั๋วโดยสารยังต้องทำการบันทึกไว้ในระบบเนื่องจากมีความสำคัญเพื่อใช้ในการตรวจสอบยอดจำหน่ายตั๋วโดยสาร และยังสามารถนำไปใช้เพื่อการวิเคราะห์ข้อมูลในด้านต่างๆ ได้อีกด้วย

5.2.1 ชื่อแบบรูป

งานจำหน่ายตั๋วโดยสารที่สถานีโดยสาร

5.2.2 เป้าหมาย

5.2.2.1 เป้าหมายที่ 1

การจำหน่ายตั๋วในแต่ละวันจำเป็นต้องมีการบันทึกข้อมูลเพื่อสามารถใช้อ้างอิงได้ว่ามียอดในการจำหน่ายตั๋วโดยสารเท่าไร มีการจำหน่ายไปในแบบใดบ้าง ตลอดจนต้องบันทึกเจ้าหน้าที่ที่รับผิดชอบในการจำหน่ายตั๋วแต่ละส่วนเอาไว้ โดยจะต้องสามารถนำข้อมูลมาสรุปเป็นรายงานยอดจำหน่ายตั๋วได้

ประเมินโดย มีการบันทึกการขายเก็บไว้ และสรุปเป็นรายงานรายวัน รายเดือนได้

5.2.2.2 เป้าหมายที่ 2

การจำหน่ายตั๋วโดยเฉพาะในช่วงเวลาเร่งด่วนจะต้องสามารถจำหน่ายได้อย่างรวดเร็ว และไม่ควรรจะต้องใช้เวลานาน การจำหน่ายตั๋วไม่ว่าจะเป็นประเภทใดจะต้องไม่ยุ่งยาก การใช้งานระบบควรช่วยให้งานจำหน่ายตั๋วโดยสารมีความผิดพลาดลดลง

ประเมินโดย จำหน่ายตั๋วได้อย่างมีประสิทธิภาพ ถูกต้อง และรวดเร็ว

5.2.3 ปัญหาโดเมน

5.2.3.1 ปัญหาที่ 1 จำหน่ายตัวในลักษณะที่แตกต่างกันหลายรูปแบบ

ประเภทของปัญหา เพิ่มประสิทธิภาพของการทำงาน

มีรายละเอียดคือ ต้องการให้ตัวที่จำหน่ายมีหลากหลายรูปแบบตามพฤติกรรมการใช้งานของผู้โดยสาร การขายต้องสนับสนุนรูปแบบการขายบัตรที่มีในระบบและรูปแบบที่อาจมีเพิ่มเติมในอนาคต

ต้นเหตุคือ รูปแบบของตัวโดยสารมีไม่จำกัด

จาก ฝ่ายวางแผนกลยุทธ์

5.2.3.2 ปัญหาที่ 2 การจำหน่ายตัวโดยสารให้รวดเร็วทันความต้องการ

ประเภทของปัญหา เพิ่มประสิทธิภาพของการทำงาน

มีรายละเอียดคือ ในช่วงโมเมนต์ความต้องการผู้โดยสารมีจำนวนมากส่วนที่จำหน่ายตัวนั้นมีย่อยอย่างจำกัด ทำให้การจำหน่ายตัวทำได้ไม่ทันกับความต้องการ

ต้นเหตุคือ จำนวนผู้โดยสารมีมากและไม่สม่ำเสมอ

จาก ผู้โดยสารที่มาใช้งานพร้อมๆกันจำนวนมาก

ที่ สถานีโดยสารที่เป็นจุดเชื่อมต่อกับระบบขนส่งขนาดใหญ่อื่นๆ จะทำให้จำนวนผู้โดยสารที่มาต่อรถไฟมีมากเป็นช่วงๆ

5.2.4 ปัญหาการพัฒนา

5.2.4.1 ปัญหาที่ 1 การนำเครื่องจำหน่ายตัวอัตโนมัติมาใช้งาน

ประเภทของปัญหา ความสามารถในการพัฒนา

มีรายละเอียดคือ การนำเครื่องจำหน่ายตัวแบบอัตโนมัติมาใช้จะช่วยให้ผู้โดยสารสามารถซื้อตัวได้ด้วยตัวเอง แต่เครื่องจำหน่ายตัวอัตโนมัติที่นำมาใช้อาจไม่เหมาะสมกับผู้โดยสารทุกกลุ่ม ตัวอย่างประเภทมีความยุ่งยากและไม่เหมาะสมในการขายด้วยเครื่องอัตโนมัติ การจำหน่ายตัวด้วยเครื่องอัตโนมัติจึงจะต้องพัฒนาซอฟต์แวร์ขึ้นมาเป็นพิเศษให้รองรับลักษณะการทำงานที่เฉพาะเจาะจง ซึ่งจัดได้ว่าเป็นระบบที่คุ้นเคยแต่เฉพาะผู้ที่เคยพัฒนาซอฟต์แวร์สำหรับระบบฝังตัวลักษณะนี้มาก่อนเท่านั้น ดังนั้นทีมพัฒนาจำเป็นต้องศึกษาลักษณะการทำงานของเครื่องจำหน่ายตัวอัตโนมัติ ตลอดจนต้องศึกษาคำสั่งต่างๆที่ใช้ในการพัฒนาซอฟต์แวร์สำหรับควบคุมเครื่องซึ่งรวมไปถึงลักษณะการทำงานที่เป็นส่วนต่อประสานผู้ใช้ด้วย

5.2.4.2 ปัญหาที่ 2 ไม่สามารถกำหนดประเภทของตัวโดยสารที่จะมีในอนาคต

ประเภทของปัญหา ค่าใช้จ่ายในการพัฒนา

มีรายละเอียดคือ รูปแบบของตั๋วโดยสารเป็นปัจจัยที่ทำให้ระบบในการจำหน่ายตั๋วโดยสารมีความซับซ้อนและเกิดการทำงานที่ยุ่งยาก ซึ่งจากลักษณะของระบบทำให้ไม่สามารถกำหนดประเภทของตั๋วโดยสารและจำกัดลักษณะได้อย่างชัดเจน การจะพัฒนาระบบให้สามารถสนับสนุนการจำหน่ายตั๋วโดยสารได้ทุกประเภทจึงทำได้ยาก อาจมีค่าใช้จ่ายในการพัฒนาสูงกว่าปกติ

5.2.5 ลักษณะโดเมน

5.2.5.1 ลักษณะการซื้อตั๋วโดยสารจากเครื่องจำหน่ายตั๋วอัตโนมัติ

การซื้อจากเครื่องจำหน่ายตั๋วโดยสารอัตโนมัติจะเริ่มจากการเลือกสถานีโดยสาร แล้วจำนวนค่าโดยสารจะแจ้งให้ทราบ ในบางกรณีจะมีให้ระบุจำนวนตั๋วโดยสารที่ซื้อด้วยว่ากี่ใบ รูปแบบการชำระค่าโดยสารของเครื่องจำหน่ายตั๋วโดยสารจะขึ้นอยู่กับคุณสมบัติของเครื่อง ซึ่งอาจรวมไปถึงการชำระค่าโดยสารผ่านบัตรเครดิต

5.2.5.2 ลักษณะการซื้อตั๋วโดยสารจากพนักงาน

การซื้อตั๋วโดยสารจากพนักงาน จะขึ้นอยู่กับผู้โดยสารว่าจะแจ้งซื้อตั๋วโดยสารในลักษณะใด หรือเป็นหน้าที่ของพนักงานจำหน่ายตั๋วโดยสารที่จะช่วยให้คำแนะนำในการซื้อตั๋ว อย่างไรก็ตามตั๋วโดยสารบางประเภทจะไม่มีเครื่องจำหน่ายตั๋วโดยสารไว้ก่อน เช่นตั๋วโดยสารแบบเติมเงิน ตั๋วโดยสารแบบเที่ยวเดียว เป็นต้น ซึ่งจะเป็นการใช้ระบบออกตั๋วโดยสารแบบทันทีในขณะที่ทำการซื้อตั๋วโดยสารผ่านระบบคอมพิวเตอร์

5.2.5.3 รูปแบบของตั๋วโดยสารที่มีการจำหน่าย

รูปแบบตั๋วโดยสารสำหรับการโดยสารในระบบรถไฟมีรูปแบบที่สามารถจำแนกได้หลายประเภท แต่สำหรับการนำมาใช้จริงอาจไม่มีความจำเป็นต้องจำหน่ายตั๋วโดยสารในทุกรูปแบบ โดยมีรูปแบบที่เป็นไปได้ดังนี้

- 1) ตั๋วเที่ยวเดียว(ไปอย่างเดียว)
- 2) ตั๋วไปกลับ(ไปกลับตามสถานีโดยสารที่กำหนด)
- 3) ตั๋วรายวันไม่จำกัด(ใช้ไม่จำกัดใน 1 วัน)
- 4) ตั๋วรายวันจำกัด(ใช้ได้หลายครั้งใน 1 วัน แต่ใช้ได้ไม่เกินเที่ยวที่กำหนด)
- 5) ตั๋วรายสัปดาห์ รายเดือน ฯลฯ (ตามแต่จะกำหนดช่วงอายุของตั๋วโดยสาร)
- 6) ตั๋วตามจำนวนเที่ยว
- 7) ตั๋วแบบเติมเงินหรือเติมเที่ยวได้

5.2.6 สภาพแวดล้อม

5.2.6.1 ความเกี่ยวข้องที่ 1 ลักษณะองค์กรภายนอกที่สัมพันธ์

มีรายละเอียดคือ หน่วยงานที่มีหน้าที่ดูแลระบบขนส่งมวลชนจะเป็นผู้กำหนดกรอบของราคา และประเภทของตั๋วโดยสารที่จะมีการนำมาใช้ ซึ่งอาจกำหนดเป็นตั๋วโดยสารที่ใช้ร่วมกับระบบขนส่งอื่นๆด้วย

5.2.6.2 ความเกี่ยวข้องที่ 2 ลักษณะบุคคลภายนอกที่สัมพันธ์

มีรายละเอียดคือ บุคคลบางกลุ่มจะมีสิทธิพิเศษในการซื้อตั๋วโดยสาร เช่น กลุ่มนักเรียน นักศึกษา ผู้สูงอายุ เป็นต้น

5.2.7 แรงชักจูง

- 1) ตั๋วโดยสารทั่วไปมีราคาตามระยะทางการเดินทาง
- 2) รูปแบบของตั๋วโดยสารมีหลายลักษณะ

5.2.8 ข้อจำกัดและกฎเกณฑ์

ไม่มีข้อจำกัดโดยตรง

5.2.9 ความต้องการ

5.2.9.1 หมายเลข F1

หน้าที่คือ จะต้องสามารถใช้ในการจำหน่ายตั๋วโดยสารได้
โดย ผู้จัดการฝ่ายขาย

5.2.9.2 หมายเลข F2

หน้าที่คือ จะต้องสนับสนุนการจำหน่ายตั๋วโดยสารหลายประเภท
โดย ผู้จัดการฝ่ายขาย

5.2.9.3 หมายเลข F3

หน้าที่คือ จะต้องมีการบันทึกข้อมูลการขายเก็บไว้เสมอ
โดย ผู้จัดการฝ่ายวางแผนการตลาด

5.2.9.4 หมายเลข F4

หน้าที่คือ จะต้องใช้ในการสรุปยอดจำหน่ายตั๋วโดยสารได้
โดย ผู้จัดการฝ่ายขาย

5.2.9.5 หมายเลข NF1

ความต้องการคุณสมบัติของระบบ ข้อมูลจากการจำหน่ายตั๋วโดยสารต้องมีความถูกต้อง
โดย ผู้จัดการฝ่ายขาย

5.2.9.6 หมายเลข NF2

ความต้องการต่อ/ประสาน ข้อมูลการจำหน่ายตั๋วโดยสารต้องสามารถเชื่อมโยงกับ
ฐานข้อมูลเพื่อใช้ในการวิเคราะห์ได้

โดย ผู้จัดการฝ่ายวางแผนและการตลาด

5.2.9.7 หมายเลข NF3

ความต้องการประสิทธิภาพของระบบ การจำหน่ายตั๋วโดยสารต้องทำได้ทันกับความต้องการซื้อตั๋วโดยสาร

โดย ผู้จัดการฝ่ายขาย

5.2.9.8 หมายเลข NF4

ความต้องการคุณสมบัติของระบบ จะต้องรองรับจำนวนผู้โดยสารที่มาใช้งานได้อย่าง
เพียงพอ

โดย ผู้จัดการฝ่ายขาย

5.2.10 ความเชื่อมโยงและข้อจำกัดของความต้องการ

5.2.10.1 ประเด็นที่ 1 การรองรับผู้โดยสารที่มีความหลากหลายเป็นจำนวนมาก

กับความต้องการ NF4 มีความสำคัญสูงสุด

กับความต้องการ F2 มีความสำคัญรองลงมา

มีรายละเอียดคือ การจำหน่ายตั๋วโดยสารจำเป็นต้องทันต่อความต้องการ แม้ว่าตั๋วโดยสาร
อาจมีหลายลักษณะ การขายตั๋วโดยสารแต่ละรูปแบบอาจมีความแตกต่างกันออกไป ระบบ
จำหน่ายตั๋วโดยสารก็ต้องยืดหยุ่นและสนับสนุนการจำหน่ายตั๋วโดยสารที่มีนัยของจำนวนตั๋ว
โดยสารที่ขายเป็นสำคัญ

5.2.10.2 ประเด็นที่ 2 การบันทึกข้อมูลและสรุปรายงาน

กับความต้องการ F3 โดยมีความสำคัญระดับเดียวกับความต้องการ F4

มีรายละเอียดคือ ข้อมูลที่ทำการบันทึกและรายงานสรุปผลต้องมีความสอดคล้องกัน ซึ่ง
ความต้องการเหล่านี้ต้องพิจารณาไปคู่กัน เพื่อสามารถเก็บข้อมูลให้สอดคล้องกับที่จะนำไปใช้ใน
รายงาน

5.2.11 ผู้เกี่ยวข้องและมุมมองต่างๆ

5.2.11.1 บทบาทที่ 1 ผู้โดยสาร

มีความเห็นเกี่ยวกับ F1 ว่าไม่ต้องการที่จะต่อแถวซื้อตั๋วโดยสารนาน และหากเป็นไปได้ก็ต้องการใบเสร็จในการซื้อตั๋วโดยสารโดยสารด้วย

มีความเห็นเกี่ยวกับ F2 ว่าการซื้อตั๋วโดยสารทุกประเภทน่าจะทำได้ผ่านจุดจำหน่ายตั๋วโดยสารจุดเดียวกัน

5.2.11.2 บทบาทที่ 2 พนักงานจำหน่ายตั๋วโดยสารที่สถานีโดยสาร

มีความเห็นเกี่ยวกับ F1 ว่าระบบในการจำหน่ายตั๋วโดยสารควรจะใช้งานง่าย และควรสามารถแก้ปัญหาเมื่อมีการยกเลิกการขายที่ผิดพลาดหรือคืนตั๋วโดยสารได้

มีความเห็นเกี่ยวกับ F2 ว่าการที่มีตั๋วโดยสารหลายประเภท ระบบควรจะนำมาช่วยในจุดนี้เพื่อลดความสับสนในการจำหน่ายตั๋วโดยสารได้ และไม่ว่าจะเป็นการจำหน่ายตั๋วโดยสารแบบใดการใช้งานก็ไม่ควรจะต้องต่างกันมากนัก

มีความเห็นเกี่ยวกับ F3 ว่าในการบันทึกข้อมูลจากการจำหน่ายตั๋วโดยสาร การบันทึกและผลของการบันทึกน่าจะแสดงออกมาให้ทราบด้วย อาจแสดงสรุปยอดขายจากข้อมูลที่บันทึกระหว่างการจำหน่ายตั๋วโดยสารด้วย

5.2.11.3 บทบาทที่ 3 เจ้าหน้าที่วางแผนระบบขนส่ง

มีความเห็นเกี่ยวกับ F3 ว่าการบันทึกข้อมูลต้องถูกต้อง หากระบบมีปัญหาข้อมูลที่มีการบันทึกควรจะคงอยู่

มีความเห็นเกี่ยวกับ NF4 ว่าระบบจะต้องรองรับผู้โดยสารในช่วงเวลาการใช้งานปกติได้โดยไม่มีปัญหา และในช่วงที่มีผู้โดยสารมาใช้งานมากเป็นพิเศษการทำงานของระบบจะต้องเพียงพอ และควรมีการล่าช้าได้เล็กน้อยเท่านั้น หรือมีมาตรการรองรับปัญหาที่เกิดขึ้น

มีความเห็นเกี่ยวกับ F4 ว่าการสรุปยอดการจำหน่ายตั๋วโดยสารจะต้องเป็นไปตามมุมมองที่สามารถนำมาใช้วิเคราะห์ระบบขนส่งได้ โดยข้อมูลที่ได้ต้องมีความถูกต้องครบถ้วน

5.2.12 วิธีการเพื่อแก้ปัญหาโดเมน

5.2.12.1 ประเด็นที่ 1 จำนวนตั๋วโดยสารมีหลายรูปแบบ

มีรายละเอียดคือ จากปัญหาที่ตั๋วโดยสารมีด้วยกันหลายประเภท และอาจมีการปรับเปลี่ยนหรือเพิ่มรูปแบบตั๋วโดยสารสำหรับอนาคตได้ ทำให้เกิดปัญหาด้านในหลายลักษณะ

แนวทางการแก้ปัญหา จำกัดรูปแบบที่เป็นไปได้ โดยจัดกลุ่มตามลักษณะคุณสมบัติของตัวโดยสาร ซึ่งหากบัตรประเภทใดมีลักษณะการใช้งานคล้ายกันก็จะจัดให้อยู่ในกลุ่มเดียวกัน โดยชนิดของตัวโดยสารจะยอมให้มีได้ไม่จำกัดแต่จะต้องอยู่ภายใต้กลุ่มของรูปแบบตัวโดยสารที่กำหนดไว้เท่านั้น

ข้อสรุปที่ได้ จำกัดกลุ่มของตัวโดยสารซึ่งแยกประเภทตามคุณสมบัติของตัวโดยสารที่คล้ายกัน

สัมพันธ์กับความต้องการ F2

5.2.12.2 ประเด็นที่ 2 ผู้โดยสารมีจำนวนมากทำให้งานจำหน่ายตัวโดยสารอาจไม่เพียงพอและทันต่อความต้องการ

มีรายละเอียดคือ ตามที่มีบางช่วงเวลาและบางเหตุการณ์ทำให้ผู้ที่มาซื้อตัวโดยสารมีจำนวนมาก และเกรงว่างานจำหน่ายตัวโดยสารอาจไม่เพียงพอต่อความต้องการ คงไม่สามารถหาวิธีการที่มีประสิทธิภาพร้อยเปอร์เซ็นต์มาแก้ปัญหาในลักษณะนี้ แต่ก็น่าที่จะประมาณการ มีแผนหรือระบบรองรับลักษณะที่จะเกิดขึ้น

แนวทางการแก้ปัญหา เป็นไปได้ที่จะใช้แบบจำลองการทำงานมาช่วยในการประเมิน รวมทั้งควรเอาสถิติผู้เข้ามาซื้อตัวโดยสารแต่ละลักษณะมาพิจารณา ซึ่งการจะพัฒนาระบบให้สามารถรองรับการทำงานเต็มร้อยเปอร์เซ็นต์คงไม่สามารถเป็นไปได้ แต่ระดับที่น่าจะยอมรับได้ซึ่งควรใช้กำหนดเป็นคุณสมบัติของงานจำหน่ายตัวโดยสารคือ 60 ถึง 70 เปอร์เซ็นต์ ของผู้มาซื้อตัวโดยสารจะต้องใช้เวลาไม่เกิน 5 นาที และอย่างช้าที่สุดที่ผู้โดยสารจะใช้เวลาในการซื้อตัวโดยสารคือ 20 นาที โดยเพิ่มปริมาณเครื่องจำหน่ายตัวโดยสารอัตโนมัติและพนักงานจำหน่ายตัวโดยสารเป็นการแก้ปัญหา และอาจอาศัยลักษณะตัวโดยสารที่มีด้วยกันหลายประเภทมาใช้ลดเวลาที่จะต้องซื้อตัวโดยสารได้ เช่น ตัวแบบรายวัน ตัวแบบรายเดือน เป็นต้น

ข้อสรุปที่ได้ นำแบบจำลองการทำงานมาช่วย เพื่อหาวิธีให้สามารถจัดจำหน่ายตัวโดยสารได้อย่างเพียงพอ ด้วยการประกันผลที่ได้

สัมพันธ์กับความต้องการ NF4

5.2.12.3 ประเด็นที่ 3 การบันทึกข้อมูลการจำหน่ายตัวโดยสาร

มีรายละเอียดคือ ข้อมูลที่ได้จากการจำหน่ายตัวโดยสารจำเป็นต้องมีการบันทึกเก็บเอาไว้ แต่เนื่องจากรูปแบบของการจำหน่ายตัวโดยสารมีหลายลักษณะ ไม่ว่าจะเป็นการขายโดยปกติ การขายด้วยเครื่องจำหน่ายตัวโดยสารอัตโนมัติ รวมทั้งตัวโดยสารที่มีจำหน่ายนอกเหนือจากในสถานีโดยสาร ข้อมูลที่จะต้องบันทึกนอกจากจำนวนเงิน และจำนวนตัวโดยสารที่ขาย อาจจะต้องทำการ

บันทึกข้อมูลอื่นๆเพิ่มเติมอีก เช่น พนักงานจำหน่ายตั๋วโดยสาร ช่วงเวลาที่มีการจำหน่ายตั๋วโดยสาร

แนวทางการแก้ปัญหา การบันทึกข้อมูลควรบันทึกเท่าที่จำเป็นเท่านั้น ไม่เช่นนั้นข้อมูลที่จัดเก็บจะมีจำนวนมากเกินไป เช่นการเก็บบันทึกเวลาที่ทำการจำหน่ายตั๋วโดยสารซึ่งไม่มีความจำเป็นเพียงพอ ซึ่งหนทางที่ดีและน่าจะยอมรับได้ คือการเก็บข้อมูลโดยแจกแจงตามประเภทของรูปแบบการจำหน่าย โดยบันทึกผลลัพธ์เชิงสรุป เช่น จำนวนตั๋วโดยสารที่จำหน่ายผ่านเครื่องจำหน่ายตั๋วโดยสารอัตโนมัติในช่วงเวลา 8.00 - 8.30น. เป็นต้น

ข้อสรุปที่ได้ ทำการบันทึกข้อมูลเพียงบางส่วน จะไม่บันทึกข้อมูลทั้งหมดโดยละเอียด แต่ก็ยังคงแจกแจงข้อมูลที่มีที่มาต่างกันถือเป็นข้อมูลต่างชุดกัน

สัมพันธ์กับความต้องการ F3

5.2.13 สภาวะผลลัพธ์

ระบบจำหน่ายตั๋วโดยสารที่รองรับรูปแบบของตั๋วโดยสารได้หลายรูปแบบ โดยจำกัดเป็นกลุ่มประเภทของตั๋วโดยสารที่จะมีใช้ในระบบ ระบบจำหน่ายตั๋วโดยสารที่หน้าสถานีโดยสารโดยใช้เครื่องจำหน่ายตั๋วโดยสารแบบอัตโนมัติร่วมด้วย ฐานข้อมูลจากการจำหน่ายตั๋วโดยสารที่นำไปใช้สรุปรายงาน

5.2.14 แบบรูปที่เกี่ยวข้อง

1) การใช้ตั๋วโดยสารแบบตัดค่าเดินทางอัตโนมัติ

เป็นรูปแบบหนึ่งของตั๋วโดยสาร ซึ่งมูลค่าจากการขายและการใช้งานจะต้องนำมาพิจารณาพร้อมกับตั๋วโดยสารประเภทอื่นๆด้วย

2) การเก็บและวิเคราะห์ข้อมูลการใช้บริการรถไฟฟ้า

ข้อมูลที่จัดเก็บเพื่อการวิเคราะห์การใช้งานบริการนั้น ส่วนหนึ่งจะมาจากข้อมูลการใช้งานและการจำหน่ายตั๋วโดยสาร

5.2.15 แบบรูปกระบวนการ

ไม่มีกระบวนการที่เกี่ยวข้องหรือเหมาะสมกับระบบนี้เป็นพิเศษ

5.2.16 การนำไปใช้ที่ทราบ

ยังไม่มีการนำไปใช้

5.2.17 ตัวอย่าง

ยังไม่มีตัวอย่าง

5.3 แบบรูปการกำหนดสภาพแวดล้อม

เป็นกระบวนการที่ใช้กำหนดสภาพแวดล้อมของระบบในกรณีที่ขาดความชัดเจนหรือยังไม่มี การกำหนดสภาพแวดล้อมที่ระบบจะดำเนินการไว้ เมื่อมีการกำหนดสภาพแวดล้อมจะช่วยให้สามารถจัดการ กับความต้องการ ได้อย่างถูกต้อง

5.3.1 ชื่อแบบรูป

การกำหนดสภาพแวดล้อม (Elicitation: Define Environments)

5.3.2 จุดประสงค์

- 1) เพื่อกำหนดสภาพแวดล้อม ซึ่งยังไม่ได้มีการกำหนดไว้
- 2) เพื่อลดความเปลี่ยนแปลงของคุณลักษณะของสภาพแวดล้อม
- 3) เพื่อกำหนดสภาพแวดล้อมที่เหมาะสมในการพัฒนา

5.3.3 ปัญหาการทำงาน

5.3.3.1 ปัญหาที่ 1 จะกำหนดสภาพแวดล้อมได้อย่างไร

มีรายละเอียดคือ เนื่องจากสภาพแวดล้อมอาจยังไม่ได้มีการกำหนดไว้เมื่อเริ่มต้นการพัฒนา การกำหนดสภาพแวดล้อมของระบบจึงตกเป็นหน้าที่ของผู้พัฒนาในการเก็บรายละเอียดของ ความต้องการเพื่อกำหนดเป็นสภาพแวดล้อมที่เหมาะสม ซึ่งการที่จะกำหนดสภาพแวดล้อมขึ้นมานั้น อาจได้รับสิทธิ์เต็มจากผู้สนับสนุนโครงการซึ่งก็จะเป็นเรื่องง่ายของทีมงาน แต่หากบางส่วน ผู้สนับสนุน โครงการหรือผู้ใช้งานมีการกำหนดสภาพแวดล้อมบางอย่างเอาไว้ การจะสรุป รายละเอียดของสภาพแวดล้อมสำหรับระบบจึงเป็นปัญหาที่กระทบไปยังส่วนอื่นๆตลอดการ พัฒนาระบบ

ต้นเหตุคือ ยังไม่มีความชัดเจนในการกำหนดสภาพแวดล้อม และอาจไม่ได้รับสิทธิ์และ อำนาจเต็มในการตัดสินใจเลือกสภาพแวดล้อม

5.3.3.2 ปัญหาที่ 2 จะต้องพิจารณาสภาพแวดล้อมในด้านใดบ้าง

มีรายละเอียดคือ การพัฒนาระบบจะมีส่วนของสภาพแวดล้อมที่มีส่วนที่เกี่ยวข้องในด้าน ต่างๆจำนวนมาก ทำให้เกิดปัญหามีจุดใดบ้างของสภาพแวดล้อมที่ควรจะต้องกำหนดเพื่อนำมาใช้ การเก็บความต้องการอาจไม่ได้รับรายละเอียดที่ครบถ้วนหากผู้ให้รายละเอียดไม่ได้พิจารณา ครอบคลุมปัจจัยของสภาพแวดล้อมทั้งหมดที่มีผลเกี่ยวเนื่อง

ต้นเหตุคือ ไม่สามารถแจกแจงให้ครอบคลุมสภาพแวดล้อมที่จะทำการกำหนดได้

5.3.3.3 ปัญหาที่ 3 จะต้องกำหนดสภาพแวดล้อมให้ครอบคลุมระดับในระดับใด เพื่อประโยชน์ในแง่ใดบ้าง

มีรายละเอียดคือ การกำหนดสภาพแวดล้อมมักจะพิจารณาเพียงแค่เหตุการณ์เฉพาะหน้า และช่วงต้นของการดำเนินการพัฒนาระบบ ทำให้การกำหนดสภาพแวดล้อมมีปัญหาหลังจากผ่านไประยะเวลาหนึ่ง การกำหนดสภาพแวดล้อมในหลายๆครั้งก็ไม่ก่อให้เกิดประโยชน์ต่อการพัฒนา บ่อยครั้งที่การกำหนดสภาพแวดล้อมไม่ได้สังเกตเห็นผลประโยชน์ในการนำไปใช้ ขอบเขตที่กำหนดก็มักจะครอบคลุมแค่สิ่งผู้เก็บความต้องการนั้นมองเห็นเท่านั้น เช่น การละเลยการกำหนดสภาพแวดล้อมเพื่อให้เอื้อไปถึงการพัฒนาและทดสอบระบบด้วย

ต้นเหตุคือ ขาดมุมมองในวงกว้างในการกำหนดสภาพแวดล้อมเพื่ออำนวยความสะดวกไปยังส่วนอื่นๆของการพัฒนาซอฟต์แวร์

5.3.4 สถานการณ์

- 1) ระบบไม่มีการกำหนดสภาพแวดล้อมที่ชัดเจนในเบื้องต้น
- 2) ระบบมีความเป็นไปได้ที่จะมีความเปลี่ยนแปลงเกิดขึ้นในอนาคต
- 3) ไม่ทราบรายละเอียดของระบบหรือส่วนประกอบอื่นๆที่ระบบจะต้องทำงานร่วมกับ

5.3.5 กลุ่มประเภทระบบที่นำไปใช้

ระบบทั่วไปที่สามารถพัฒนาในสภาพแวดล้อมที่มีให้เลือกหรือเป็นไปได้หลายรูปแบบ

5.3.6 แรงชักจูง

- 1) ไม่มีการกำหนดระบบหรือสภาพแวดล้อมที่เกี่ยวข้องกับระบบอย่างแท้จริง
- 2) มีความเป็นไปได้ว่าจะมีการเปลี่ยนแปลงในอนาคต

5.3.7 วิธีการ

- 1) กำหนดคุณลักษณะของฮาร์ดแวร์
- 2) กำหนดระบบปฏิบัติการ
- 3) กำหนดคลังโปรแกรม (Library) ของซอฟต์แวร์
- 4) กำหนดเวอร์ชัน
- 5) ตรวจสอบความเข้ากันได้และความเหมาะสม
- 6) พิจารณาและเลือกส่วนต่อประสานที่เกี่ยวข้อง เช่น ไดรฟ์เวอร์ฐานข้อมูล(Database Driver) โปรโตคอล เป็นต้น
- 7) จัดแจงส่วนต่อประสานให้สามารถทำงานร่วมกันได้

- 8) กำหนดซอฟต์แวร์ที่จะมีการใช้งานเป็นพื้นฐานของระบบ เช่น ระบบต้องมีซอฟต์แวร์อ่านเอกสารนามสกุลพีดีเอฟ เอกสารตารางคำนวณ เป็นต้น
- 9) กำหนดเวอร์ชันของซอฟต์แวร์ที่นำมาใช้ให้เหมาะสมกับระบบ
- 10) ตรวจสอบข้อมูลระบบส่วนอื่นๆที่เกี่ยวข้องกับการทำงาน
- 11) กำหนดสภาพการทำงาน (ในระบบที่เกี่ยวข้องกับลักษณะการทำงานด้วย)

5.3.8 ค่าใช้จ่าย

ค่าใช้จ่ายในเชิงมูลค่า มีค่าใช้จ่ายน้อย

ค่าใช้จ่ายในเชิงกำลังคน ต้องใช้ทีมงานที่วางโครงสร้างสถาปัตยกรรมระบบเป็นผู้กำหนด

ค่าใช้จ่ายในเชิงองค์ความรู้ ใช้ความสามารถในการวางโครงสร้างสถาปัตยกรรมระบบ

ค่าใช้จ่ายในเชิงเวลา ใช้เวลาไม่มากนัก

5.3.9 ประสิทธิภาพการทำงาน

การกำหนดสภาพแวดล้อมต้องพิจารณาไปถึงปัจจัยที่จะเข้ามามีบทบาททำให้ระบบมีการเปลี่ยนแปลงด้วย การกำหนดสภาพแวดล้อมในตอนต้นของการพัฒนามีจุดประสงค์หลักเพื่อให้สามารถพัฒนาระบบได้ง่ายและมีความชัดเจน โดยมักจะกำหนดสภาพแวดล้อมให้มีความยืดหยุ่นในระดับหนึ่ง เพื่อเอาไว้สำหรับการเปลี่ยนแปลงระบบไปตามลักษณะที่เป็นไปได้ที่จะเกิดขึ้น สภาพแวดล้อมที่ได้พิจารณาเอาไว้แล้วจะจะสามารถรองรับได้โดยไม่มีการเปลี่ยนแปลง ปัญหาสำคัญในการกำหนดสภาพแวดล้อมคือต้องคุยกับเจ้าของโครงการให้เข้าใจว่าเหตุใดต้องการสภาพแวดล้อมแบบนี้และเหตุใดจึงกำหนดแบบนี้ไม่ได้ ทั้งนี้บางครั้งการกำหนดบางอย่างอาจถูกกำหนดไว้ตั้งแต่การเริ่มโครงการแล้ว ซึ่งควรที่จะกลับไปแก้ไขการกำหนดตอนต้นเสีย แม้อาจจะดูไม่สมควรแต่ด้วยเหตุผลก็ควรจะสามารถทำได้เพื่อให้เกิดประโยชน์รวมทั้งสองฝ่ายและในระยะยาว

5.3.10 สถานะผลลัพธ์

ระบบที่พัฒนามีความชัดเจนมากขึ้นในลักษณะขอบเขตและข้อจำกัดในการพัฒนา สามารถประเมินความต้องการตามลักษณะสภาพแวดล้อมที่กำหนดขึ้น มีความชัดเจนและที่นำไปในเรื่องของผู้กำหนดและจุดประสงค์ของสภาพแวดล้อมของระบบ และระบบที่สร้างขึ้นตามสภาพแวดล้อมที่กำหนดจะสามารถทำงานเข้ากันได้กับส่วนอื่นๆหรือระบบอื่นๆที่เกี่ยวข้อง

5.3.11 การเข้าใจการใช้งาน

ยังไม่มีนำไปใช้

5.3.12 ตัวอย่าง

ยังไม่มีตัวอย่าง

5.3.13 แบบรูปที่เกี่ยวข้อง

ไม่มีกระบวนการที่เกี่ยวข้องหรือสนับสนุน

5.4 แบบรูปการเว้นระยะห่างระหว่างรถไฟที่ปลอดภัย

การเว้นระยะห่างระหว่างรถไฟที่ปลอดภัยเป็นการกำหนดระยะห่างหรือช่องว่างที่สามารถป้องกันหรือลดความเสี่ยงจากอุบัติเหตุที่เกิดขึ้น โดยเป็นการตั้งค่าเพื่อกำหนดการเว้นระยะขบวนที่รองรับขบวนรูปแบบต่างๆ ซึ่งจะทำงานสอดคล้องกับระบบที่เกี่ยวข้องจากภายนอกหรือแม้แต่น้ำไฟที่ไม่อยู่ในการควบคุมด้วยระบบอัตโนมัติ โดยการทำงานจะอิงตามรูปแบบการหยุดอย่างปลอดภัย

5.4.1 ชื่อแบบรูป

การเว้นระยะห่างระหว่างรถไฟที่ปลอดภัย (ATP: Safety Train separation)

5.4.2 เป้าหมาย

5.4.2.1 เป้าหมายที่ 1

การเว้นระยะห่างที่เหมาะสมที่จะทำให้เกิดความปลอดภัยจากระยะที่จะสามารถหยุดรถหรือหลีกเลี่ยงการชนของรถไฟได้ โดยรถไฟจะต้องสามารถไปอยู่ในตำแหน่งปลอดภัยได้เมื่อเกิดเหตุการณ์ใดๆขึ้น โดยอาศัยการกำหนดควบคุมระยะห่างที่จะหลีกเลี่ยงจากเหตุร้ายที่ตามมา

ประเมินโดย ระยะห่างระหว่างรถจะถูกกำหนดให้อยู่ในช่วงที่ปลอดภัยเสมอ

5.4.2.2 เป้าหมายที่ 2

เมื่อระบบเกิดเหตุการณ์ใดๆขึ้น ซึ่งเป็นผลให้รถไฟต้องหยุดเพื่อความปลอดภัย เช่น มีรถจอดเสีย หรือเข้าสู่ทางที่ไขว้ร่วม ระบบจะต้องคำนวณระยะปลอดภัยในการหยุดรถ และเมื่อเข้าสู่ระยะที่วิกฤตเข้าสู่อันตรายได้ระบบก็จะต้องดำเนินการในส่วนของการหยุดรถโดยทันที

ประเมินโดย ระบบจะนำไปสู่การหยุดรถได้อย่างทันท่วงที

5.4.3 ปัญหาโดเมน

5.4.3.1 ปัญหาที่ 1 ระยะห่างที่เหมาะสมจะพิจารณาจากอะไร

ประเภทของปัญหา เพิ่มประสิทธิภาพของการทำงาน

มีรายละเอียดคือ การพิจารณาระยะห่างที่เหมาะสมต้องพิจารณาไปตามลักษณะอื่นๆที่เกี่ยวข้อง เช่น รถไฟด้วยกัน สุดทาง ทางชำรุด เป็นต้น และยังมีปัจจัยที่ขึ้นกับตัวรถไฟ ณ ระยะเวลา ความเร็ว และตำแหน่งที่แตกต่างกันออกไปด้วย

ต้นเหตุคือ การกำหนดระยะห่างมีปัจจัยมาจากหลายส่วน

5.4.4 ปัญหาการพัฒนา

5.4.4.1 ปัญหาที่ 1 การคำนวณระยะห่างที่ปลอดภัยควรทำในขณะใด

ประเภทของปัญหา ประสิทธิภาพของระบบ

มีรายละเอียดคือ การคำนวณหาระยะห่างมีปัจจัยที่เกี่ยวข้องหลายส่วน จึงเป็นเรื่องยากที่จะกำหนดเป็นระยะห่างที่คงตัว ดังนั้นระยะห่างที่ปลอดภัยจึงเกิดจากการคำนวณตามปัจจัยที่ถูกต้องในช่วงเวลานั้นๆ ซึ่งเป็นปัญหาว่าการคำนวณควรจะเกิดขึ้นเมื่อใดหรือต้องคำนวณตลอดเวลา

5.4.4.2 ปัญหาที่ 2 ระบบข้อมูลและอุปกรณ์ตรวจจับสัญญาณต่างๆจะทำงานถูกต้องเพียงพอหรือไม่

ประเภทของปัญหา ข้อจำกัดของเทคโนโลยี

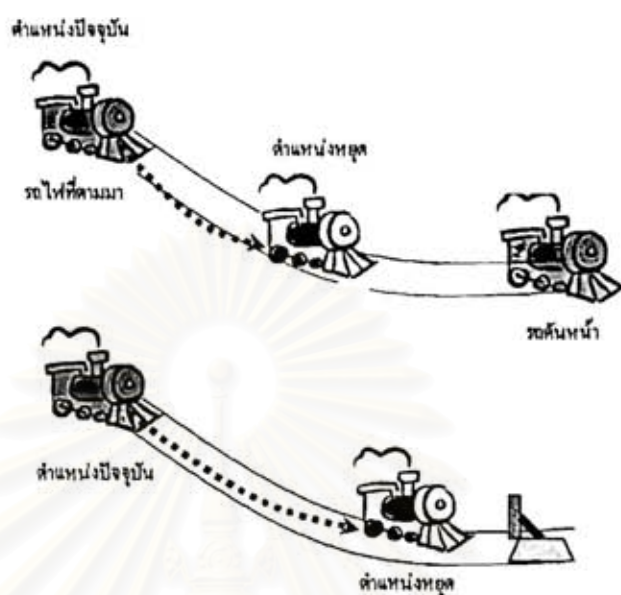
มีรายละเอียดคือ การคำนวณเกี่ยวกับระยะจะต้องอาศัยข้อมูลจากหลายๆส่วนประกอบกัน เพื่อคำนวณระยะห่างที่จะปลอดภัย ข้อมูลที่จะนำมาใช้ในการคำนวณจึงมีความสำคัญมาก และมีความเป็นไปได้ที่จะได้ข้อมูลไม่ครบหรือไม่ถูกต้อง ในการพัฒนาต้องเอาประเด็นเหล่านี้มาใช้ประกอบด้วย ต้องวิเคราะห์ถึงลักษณะของอุปกรณ์และความแม่นยำประกอบด้วย โดยอุปกรณ์ในตรวจวัดค่าบางประเภทอาจต้องการความแม่นยำมากกว่าปกติ

5.4.5 ลักษณะโดเมน

5.4.5.1 การพิจารณาระยะหยุดรถ

หยุดรถจะพิจารณาโดยอ้างอิงตามรูปแบบการหยุดรถอย่างปลอดภัย(Safety Braking Model)

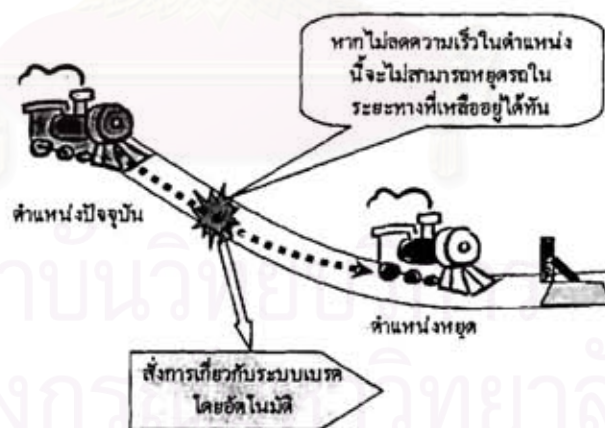
5.4.5.2 แผนภาพที่ 1 ลักษณะของระยะห่างที่ปลอดภัย รูปที่ 5.1



รูปที่ 5.1 ลักษณะของระบบที่ปลอดภัย

คำอธิบาย แสดงระบบที่รถไฟจะสามารถหยุดได้อย่างปลอดภัยโดยไม่ชน (มีระยะเพียงพอในการหยุดรถ)

5.4.5.3 แผนภาพที่ 2 การสั่งการหยุดรถไฟเมื่อถึงจุดเสี่ยง รูปที่ 5.2



รูปที่ 5.2 การสั่งการหยุดรถไฟเมื่อถึงจุดเสี่ยง

คำอธิบาย ระบบจะดำเนินการส่วนของระบบหยุดโดยอัตโนมัติเมื่อรถไฟเข้าสู่จุดที่ ระยะห่างที่เหลืออาจเกิดอันตราย (ระยะไม่เพียงพอในการหยุดรถ)

5.4.6 สภาพแวดล้อม

5.4.6.1 ความเกี่ยวข้องที่ 1 ลักษณะระบบภายนอกที่สัมพันธ์

มีรายละเอียดคือ ระบบกำกับการเดินรถไฟ(Train Supervisions) ซึ่งจะเป็นตัวกำหนดการใช้เส้นทางและการปล่อยรถ ซึ่งมีผลต่อระยะห่างและการใช้รางของรถไฟแต่ละขบวน

5.4.7 แรงชักจูง

- 1) ระบบรถไฟที่ควบคุมตัวรถได้แบบอัตโนมัติหรือกึ่งอัตโนมัติ
- 2) เป็นระบบที่ใช้การบริหารรางโดยการอิงระยะห่างของราง (ไม่ใช่รางแบบแบ่งช่วงของการใช้งาน)

5.4.8 ข้อจำกัดและกฎเกณฑ์

- 1) การทำงานจะสามารถนำไปสู่การหยุดรถได้อย่างทันที่
- 2) ขกเลิกการทำงานในกรณีดำเนินการเพื่อทำให้ระบบที่เสียกลับมาใช้งานได้
- 3) ค่าความไม่แน่นอนของตำแหน่ง ค่าความเร็ว ความคลาดเคลื่อน จะต้องนำมาพิจารณาด้วย

5.4.9 ความต้องการ

5.4.9.1 หมายเลข F1

หน้าที่คือ จะต้องคำนวณระยะปลอดภัยได้อย่างถูกต้อง
โดย ผู้จัดการระบบ

5.4.9.2 หมายเลข F2

หน้าที่คือ จะต้องครอบคลุมจุดที่มีรถเสียอยู่
โดย ผู้จัดการระบบ

5.4.9.3 หมายเลข F3

หน้าที่คือ จะต้องทำงานร่วมกับรถไฟที่ไม่อยู่ในระบบได้
โดย ผู้จัดการระบบ

5.4.9.4 หมายเลข F4

หน้าที่คือ จะต้องควบคุมในจุดที่มีการใช้รางร่วมกันได้
โดย ผู้จัดการระบบ

5.4.9.5 หมายเลข F5

หน้าที่คือ จะต้องนำไปสู่การหยุดรถหากมีการเคลื่อนที่เกินระยะปลอดภัย
โดย ผู้จัดการระบบ

5.4.9.6 หมายเลข NF1

ความต้องการคุณสมบัติของระบบ จะต้องทำงานได้ทั้งกับรถไฟที่มีความยาวแบบตายตัว และแบบไม่แน่นอน

โดย ผู้จัดการระบบ

5.4.9.7 หมายเลข NF2

ความต้องการในการต่อประสาน จะต้องสอดคล้องกับรูปแบบการหยุดอย่างปลอดภัย

โดย ผู้จัดการระบบ

5.4.10 การเสนอวิธีการเพื่อแก้ปัญหาโดเมน

5.4.10.1 ประเด็นที่ 1 การคำนวณระยะห่าง

มีรายละเอียดคือ ระยะทางที่ปลอดภัยสำหรับการหยุดจะขึ้นอยู่กับตำแหน่งของรถกับวัตถุ ลักษณะราง ความเร็วและความสามารถในการหยุดของรถ ซึ่งการคำนวณหาระยะที่ปลอดภัยจะเป็นการพิจารณาระยะที่เพียงพอในการหยุดในขณะนั้น หรือการประมาณระยะที่ไม่ควรเข้าไปใกล้รถขบวนหน้าจนเกินไป

แนวทางการแก้ปัญหา วิธีการหนึ่งที่มีถูกนำมาใช้คือการกำหนดระยะของเส้นทางเป็นช่วงๆ เช่น ช่วงละ 100 เมตร โดยการคำนวณจะคำนวณเมื่อรถไฟเข้าสู่ช่วงถัดไปของทาง ระยะทางจะถูกคำนวณจากความเร็วสูงสุด (ตามที่กำหนดให้วิ่งได้) หรือความเร็วรถไฟขณะนั้น เพื่อคำนวณระยะหยุด และเมื่อเปรียบเทียบกับระยะทางที่ใช้กับรางก็จะได้ช่วงของรางที่รถจะหยุดได้ทัน โดยหากเกิดกรณีช่วงของรางนั้นทับกับช่วงการหยุดของรถไฟขบวนอื่น หรือไม่อยู่ในช่วงที่ต้องการก็จะถือว่าเกิดกรณีที่จะต้องดำเนินการแจ้งเพื่อดำเนินการหยุดรถทั้งแบบปกติและแบบฉุกเฉิน ในทางปฏิบัติการแบ่งช่วงจะพิจารณาตาม Track Circuit

ข้อสรุปที่ได้ พิจารณาโดยแบ่งรางเป็นช่วง คำนวณระยะขณะนั้นเพื่อตรวจสอบ

สัมพันธ์กับความต้องการ F1

5.4.10.2 ประเด็นที่ 2 การกำหนดและควบคุมการตอบสนอง

มีรายละเอียดคือ เมื่อรถไฟไม่อยู่ในขีดจำกัดของการกำหนดระยะ นั่นคือระยะและความเร็วเลขขอบเขตที่กำหนดเป็นระยะที่มีความปลอดภัย ระบบจะต้องทำการตอบสนองเพื่อป้องกันผลที่ตามมา

แนวทางการแก้ปัญหา ระบบจะต้องทำการตอบสนองเมื่อ ระยะหยุดใกล้จะเหลือน้อยกว่าระยะปลอดภัย(ไม่ยอมให้น้อยกว่า) ความเร็วเกินกว่าความเร็วที่ใช้คำนวณตามระยะปลอดภัย และเกิดการกีดขวางฉับพลันในระยะที่ปลอดภัย โดยระบบจะต้องดำเนินการแจ้งให้ผู้ควบคุมรถไฟ

ทราบในทันที โดยการเตรียมระบบหยุดรถให้พร้อมทำงาน ไม่ยอมให้เพิ่มความเร็วดำเนินการลดความเร็ว รวมไปถึงดำเนินการหยุดรถแบบฉุกเฉิน

ข้อสรุปที่ได้ กำหนดปัจจัยที่ต้องตอบสนองและวิธีการตอบสนองที่จะดำเนินการตามสัมพันธภาพความต้องการ F5

5.4.11 ความเชื่อมโยงและข้อจำกัดของความต้องการ

5.4.11.1 ประเด็นที่ 1 การหยุดรถเมื่อไม่อยู่ในขอบเขตตามรูปแบบการหยุดรถอย่างปลอดภัยกับความต้องการ F5 โดยมีความสำคัญเทียบเท่ากับความต้องการ

มีรายละเอียดคือ การทำงานของระบบจะอิงกับรูปแบบการหยุดรถอย่างปลอดภัย ซึ่งจะเป็ยรายละเอียดของระยะที่จะต้องทำการหยุดรถตามความเร็วในขณะนั้น เมื่อเห็นว่าอยู่ในระยะที่ไม่ปลอดภัยแล้ว การหยุดรถก็จะถูกดำเนินการขึ้นในทันที โดยบวกค่าระยะที่เผื่อไว้ตามที่มีการกำหนดด้วย การหยุดรถและการควบคุมตามรูปแบบการหยุดอย่างปลอดภัยจะต้องพิจารณาควบคู่กันไป

5.4.11.2 ประเด็นที่ 2 การคำนวณระยะที่ปลอดภัยเกี่ยวกับรถไฟที่อยู่และไม่อยู่ในระบบกับความต้องการ F2 ความสำคัญเทียบเท่ากับ F3 โดยมีความสำคัญสูงสุด

กับความต้องการ F1 ความสำคัญเทียบเท่ากับ F4 โดยมีความสำคัญรองลงมา

มีรายละเอียดคือ เมื่อรางในระบบมีการใช้งานร่วมกับรถไฟขบวนอื่นๆที่ไม่ได้ควบคุมโดยระบบ เช่น รถไฟที่เคลื่อนที่เป็นกรณีพิเศษ รถไฟที่ไม่มีระบบอัตโนมัติในการควบคุม เมื่อเป็นรถไฟในระบบข้อมูลต่างๆจะมีความสมบูรณ์กว่าและสามารถควบคุมและกำหนดการทำงานของรถไฟได้ แต่เมื่อทำงานร่วมกับรถไฟที่ไม่อยู่ในระบบอาจทำให้เกิดปัญหาในการควบคุมรถไฟที่อยู่นอกระบบ การใช้รางร่วมกันจึงต้องให้ความสำคัญกับรถไฟที่อยู่นอกระบบด้วย และเนื่องจากขาดความสามารถในการจัดการอย่างสมบูรณ์ ในการควบคุมระยะห่างซึ่งไม่สามารถกำหนดที่รถไฟที่อยู่นอกระบบได้ รถไฟในระบบจึงต้องได้รับการควบคุมอย่างเคร่งครัดและให้การเว้นระยะเป็นไปตามรถไฟนอกระบบเมื่อมีการใช้รางร่วมกัน

5.4.12 มุมมองจากผู้เกี่ยวข้อง

5.4.12.1 บทบาทที่ 1 พนักงานควบคุมไฟ

มีความเห็นต่อความต้องการ F3 ดังนี้

หากมีรถไฟขบวนอื่นๆที่นอกเหนือจากที่ถูกควบคุมอยู่โดยระบบ ระบบควรสามารถบอกให้ทราบ เนื่องจากการควบคุมรถไฟเหล่านี้เป็นลักษณะการควบคุมด้วยมือ จึงอาจจะต้องติดต่อกับศูนย์ควบคุมและอาจต้องระมัดระวังเป็นพิเศษ

มีความเห็นต่อความต้องการ NF2 ดังนี้

เมื่อรถไฟกำลังจะเข้าสู่ระยะที่ทำการจำกัดระยะและความเร็วควรจะเตือนในทันที เช่น หากวิ่งด้วยความเร็วปัจจุบันต่อไปอีก 5 นาที จะทำให้รถชนกับรถขบวนข้างหน้า และหากเข้าสู่ระยะที่วิกฤติแล้วระบบก็ควรจะสามารถชะลอความเร็วเองได้ หรือทำการหยุดอย่างกะทันหันโดยอัตโนมัติเมื่อเกิดเหตุการณ์สุดวิสัย

5.4.13 สภาวะผลลัพธ์

ระบบที่เว้นระยะรถไฟตามความเร็วและตำแหน่งในขณะเวลานั้น ทำให้ใช้รางได้อย่างมีประสิทธิภาพ ระบบซึ่งสามารถหยุดหรือชะลอความเร็วของรถไฟโดยอัตโนมัติเพื่อให้เกิดความปลอดภัยในระยะการเคลื่อนที่

5.4.14 ระดับการยอมรับความปลอดภัย

5.4.14.1 เงื่อนไขการยอมรับที่ 1 ระยะห่างอยู่ในช่วงปลอดภัย

ระดับความสำคัญ สำคัญยิ่ง

จัดการและรับผิดชอบโดย ผู้บริหาร

5.4.14.2 เงื่อนไขการยอมรับที่ 2 การหยุดหรือชะลอความเร็วโดยอัตโนมัติ

ระดับความสำคัญ สูง

จัดการและรับผิดชอบโดย หัวหน้าโครงการ

5.4.15 ความต้องการความปลอดภัย

SR1: ทำการป้องกัน ระยะห่างระหว่างรถไฟน้อยเกินไป

SR2: ทำการป้องกัน ระยะจากรถไฟถึงสิ่งกีดขวางหรือจุดสิ้นสุดของรางน้อยเกินไป

SR3: ทำการตรวจสอบ การเพิ่มความเร็วที่ทำให้ระยะห่างระหว่างรถไฟน้อยเกินไป

SR4: ตอบสนองต่อ ระยะห่างระหว่างรถไฟน้อยเกินไป

SR5: ตอบสนองต่อ การเพิ่มความเร็วที่ทำให้ระยะห่างระหว่างรถไฟน้อยเกินไป

5.4.16 นัยความปลอดภัย

5.4.16.1 สถานการณ์ที่ 1 การคำนวณระยะห่างผิดพลาด

ระดับความเป็นไปได้ มีความเป็นไปได้ (Probable)

ระดับความอันตราย ช่วงอันตราย (Critical)

5.4.16.2 สถานการณ์ที่ 2 ความคลาดเคลื่อนของการวัดความเร็วและตำแหน่ง

ระดับความเป็นไปได้ มีความเป็นไปได้ (Probable)

ระดับความอันตราย เล็กน้อย (Marginal)

5.4.17 เหตุการณ์นำไปสู่อุบัติเหตุ

5.4.17.1 เหตุการณ์ที่ 1 ตัวตรวจจับความเร็วของรถทำงานผิดพลาด

เนื่องจาก ความผิดพลาดจากการทำงานของฮาร์ดแวร์

5.4.17.2 เหตุการณ์ที่ 2 ตัวตรวจจับตำแหน่งของรถทำงานผิดพลาด

เนื่องจาก ความผิดพลาดจากการทำงานของฮาร์ดแวร์

5.4.17.3 เหตุการณ์ที่ 3 การคำนวณระยะห่างไม่ถูกต้อง

เนื่องจาก ความผิดพลาดจากการทำงานของซอฟต์แวร์

5.4.17.4 เหตุการณ์ที่ 4 ทำการควบคุมรถโดยไม่ระมัดระวัง

เนื่องจาก ความผิดพลาดจากการทำงานของผู้ใช้งาน

5.4.17.5 เหตุการณ์ที่ 5 ระบบเตือนการหยุดรถหรือการชะลอรถไม่ทำงาน

เนื่องจาก ความผิดพลาดจากการทำงานของซอฟต์แวร์

5.4.18 อันตรายที่เกิดขึ้นได้

5.4.18.1 อันตรายที่ 1 รถไฟชนกัน

ความเสียหายที่เกิดขึ้น รถไฟเสียหายและอาจทำให้รถไฟตกราง ทำให้ผู้โดยสารได้รับอันตรายโดยเฉพาะการชนกันที่มีความเร็วหรือในลักษณะประสานงาน ทำให้ระบบหรือเครือข่ายหยุดชะงัก และอาจก่อให้เกิดอุบัติเหตุอื่นๆตามมา

5.4.18.2 อันตรายที่ 2 รถไฟชนสิ่งกีดขวางหรือวิ่งเลยจุดสิ้นสุดของราง

ความเสียหายที่เกิดขึ้น รถไฟเสียหายหรือตกราง ทำให้ผู้โดยสารได้รับอันตราย

5.4.19 การเข้าใจการใช้งาน

ยังไม่มีให้นำไปใช้

5.4.20 ตัวอย่าง

ยังไม่มีตัวอย่าง

5.4.21 แบบรูปที่เกี่ยวข้อง

1) ATP: Restricted Route

การกำหนดการใช้รางและระยะห่างจะขึ้นอยู่กับเส้นทางการเดินรถ การเปลี่ยนแปลงเส้นทางจะทำให้การกำหนดการใช้รางและระยะห่างเปลี่ยนไป

2) ATP: End-of-track Protection

พิจารณาระยะห่างไปยังจุดสิ้นสุดของราง

3) ATP: Train Speed and Location Determination

ความเร็วและตำแหน่งมีผลต่อการคำนวณและการกำหนดระยะห่าง

4) ATP: Route Interlocking

การเปลี่ยนรางหรือเส้นทางในการวิ่งจะต้องพิจารณาระยะห่างที่ปลอดภัยประกอบ

5) ATP: Safe Breaking Model

การเว้นระยะห่างจะคำนวณและกำหนดโดยอาศัยรูปแบบการหยุดอย่างปลอดภัยในการพิจารณาและวิเคราะห์

5.4.22 แบบรูปกระบวนการ

1) Safety Requirements Checklist

นำมาใช้ในการตรวจสอบเงื่อนไขของระยะปลอดภัยในทุกกรณีที่เป็นไปได้

2) HA: Event Tree Analysis

นำมาใช้วิเคราะห์เหตุการณ์ดำเนินการเมื่อรถไฟเข้าสู่ลักษณะที่ไม่อยู่ในระยะที่ปลอดภัยในรูปแบบต่างๆ

5.5 แบบรูปการควบคุมการเปิดประตูรถไฟ

เป็นการควบคุมการเปิดประตูโดยพิจารณาการทำงานของอุปกรณ์และสถานการณ์ต่างๆเพื่อให้สามารถเปิดประตูไฟได้อย่างปลอดภัย การเปิดประตูจะถูกกำหนดและควบคุมโดยพนักงานควบคุมรถ โดยการจะสั่งงานเพื่อเปิดประตูจะต้องเป็นไปตามเงื่อนไขของความปลอดภัยที่กำหนดตามสภาพแวดล้อมในขณะนั้น ทั้งนี้ทั้งนั้นจะต้องพิจารณาการเปิดประตูสำหรับกรณีเหตุฉุกเฉินด้วย

5.5.1 ชื่อแบบรูป

การควบคุมการเปิดประตูรถไฟ (ATP: Door Opening Control Protection)

5.5.2 เป้าหมาย

5.5.2.1 เป้าหมายที่ 1

มีส่วนที่สามารถควบคุมระบบการเปิดประตูรถไฟ สำหรับพนักงานควบคุม หรือผู้ได้รับอนุญาต

ประเมินโดย ระบบที่สามารถควบคุมการเปิดประตูโดยเจ้าหน้าที่ควบคุมรถไฟ เจ้าหน้าที่ประจำสถานีโดยสาร หรือเจ้าหน้าที่ที่ศูนย์ควบคุมรถไฟ

5.5.2.2 เป้าหมายที่ 2

การเปิดประตูรถไฟต้องกระทำได้อย่างปลอดภัย โดยต้องพิจารณาองค์ประกอบอื่นๆของระบบที่มีความสัมพันธ์กัน เช่น ตำแหน่งการจอดรถ การหยุดนิ่งของรถไฟ

ประเมินโดย ประตูจะเปิดได้เมื่ออยู่ในสภาพที่ปลอดภัยเท่านั้น

5.5.2.3 เป้าหมายที่ 3

เมื่อเกิดกรณีฉุกเฉินที่ทำให้รถไฟไม่สามารถดำเนินการ โดยปกติ การสั่งการเปิดประตูอาจต้องทำภายใต้เงื่อนไขพิเศษ เช่น ขอมให้เปิดประตูได้แม้ว่ารถไฟไม่ได้จอดในตำแหน่งจอดรถที่กำหนด

ประเมินโดย อนุญาตให้เปิดประตูโดยคัตบวงเงื่อนไขทั้งเมื่อเกิดกรณีฉุกเฉิน

5.5.3 ปัญหาโดเมน

5.5.3.1 ปัญหาที่ 1 การเปิดประตูจะต้องพิจารณาเงื่อนไขที่เกี่ยวข้องกับความปลอดภัยอะไรบ้าง

ประเภทของปัญหา แก้ปัญหาของงาน

มีรายละเอียดคือ เมื่อจะทำการสั่งเปิดประตู จำเป็นต้องพิจารณารายละเอียดที่เกี่ยวข้องกับความปลอดภัยในหลายๆส่วน ก่อนที่จะเปิดประตูได้ เช่นต้องตรวจสอบว่ารถจอดแล้วหรือยัง ประตูอยู่ในตำแหน่งที่ถูกต้องหรือไม่ เป็นต้น โดยภาระเหล่านี้ควรเป็นหน้าที่ที่ต้องจัดการโดยระบบ

ต้นเหตุคือ มีปัจจัยที่เกี่ยวข้องกับการควบคุมการเปิดประตูจำนวนมาก

จาก คณะกรรมการดูแลด้านความปลอดภัย

5.5.4 ปัญหาการพัฒนา

5.5.4.1 ปัญหาที่ 1 การพิจารณากำหนดส่วนสัมพันธ์กับอุปกรณ์ที่เกี่ยวข้องกับการเปิดประตู

ประเภทของปัญหา ความสามารถในการพัฒนา

มีรายละเอียดคือ เงื่อนไขของการเปิดประตูเกี่ยวข้องกับการตรวจสอบและควบคุมในส่วนที่ใกล้ชิดกับฮาร์ดแวร์ การจะทราบสถานะภาพของระบบเพื่อนำมาใช้เป็นเงื่อนไข จึงต้องอาศัยข้อมูลจากติดต่อกับส่วนของอุปกรณ์กลไกการควบคุมประตู ดังนั้นทีมพัฒนาต้องทำความเข้าใจสัญญาณจากอุปกรณ์ที่ต้องพิจารณาและนำมาใช้กำหนดเป็นเงื่อนไขของการควบคุมการเปิดประตูรถไฟ ในกรณีที่เป็นระบบจอร์ดตามตำแหน่งก็ต้องสนใจการทำงานของอุปกรณ์ที่อยู่บริเวณสถานีโดยสารด้วย

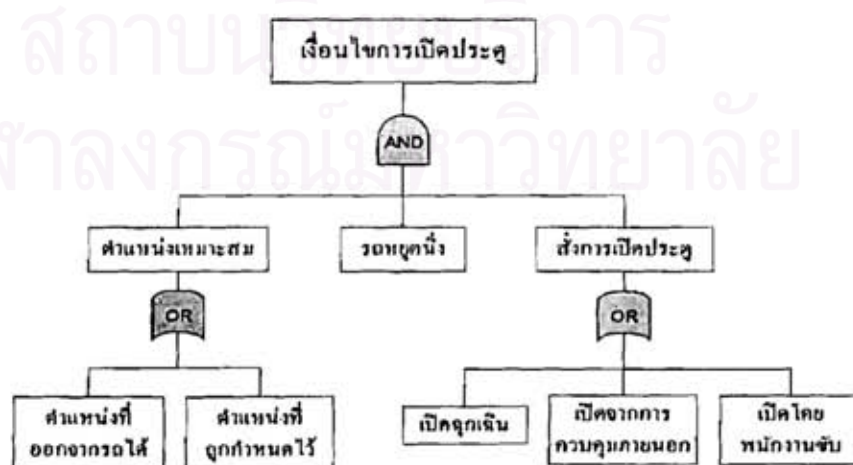
5.5.5 ลักษณะโดเมน

5.5.5.1 สิทธิการควบคุมประตู

สิทธิในการควบคุมการเปิดประตูอาจมีการกำหนดเพิ่มเติมได้ แต่โดยพื้นฐานแล้วจะมีการเปิดอยู่ 3 ลักษณะดังนี้

- 1) พนักงานควบคุม ซึ่งเป็นการสั่งการเปิดประตูในกรณีทั่วไป
- 2) พนักงานควบคุมภายนอกห้องขับ จะเป็นการสั่งการในกรณีพิเศษที่พนักงานขับไม่อยู่ในการควบคุม
- 3) ผู้โดยสาร จะเป็นการเปิดโดยใช้ประตูฉุกเฉินเท่านั้น

5.5.5.2 แผนภาพที่ 1 เงื่อนไขการเปิดประตู รูปที่ 5.3



รูปที่ 5.3 เงื่อนไขการเปิดประตู

ก่าอธิบาย การเปิดประตูจะเกิดขึ้นได้จะต้องเป็นไปตามเงื่อนไขอย่างน้อย 3 ข้อ ตามที่แสดงในรูป คือตำแหน่งการจอด สถานการณ์หยุดของรถ และคำสั่งเพื่อเปิดประตู หากขาดเงื่อนไขใดเงื่อนไขหนึ่งไป การเปิดประตูจะต้องไม่เกิดขึ้นอย่างเด็ดขาด

5.5.6 สภาพแวดล้อม

5.5.6.1 ความเกี่ยวข้องที่ 1 ลักษณะระบบภายนอกที่สัมพันธ์

มีรายละเอียดคือ ระบบสื่อสารระหว่างผู้โดยสารกับผู้ควบคุมรถหรือกับศูนย์ควบคุม

5.5.7 แรงชักจูง

- 1) การควบคุมประตูเป็นไปตามผู้มีสิทธิหรือได้รับอำนาจในการเปิด
- 2) มีอุปกรณ์ตรวจสอบสถานะปัจจุบันของรถไฟ

5.5.8 ข้อจำกัดและกฎเกณฑ์

- 1) ผู้ควบคุมการเปิดประตูได้คือผู้ที่ได้รับสิทธิเท่านั้น
- 2) การกำหนดตำแหน่งของการเปิดประตู

5.5.9 ความต้องการ

5.5.9.1 หมายเลข F1

หน้าที่คือ ต้องควบคุมการเปิดประตูโดยพนักงานควบคุมได้
โดย พนักงานควบคุม

5.5.9.2 หมายเลข F2

หน้าที่คือ ต้องควบคุมการเปิดประตูจากผู้ซึ่งได้รับสิทธิจากภายนอกได้
โดย ผู้จัดการระบบ

5.5.9.3 หมายเลข F3

หน้าที่คือ ต้องตรวจสอบความปลอดภัยก่อนจะทำการเปิดประตู
โดย ผู้จัดการระบบ

5.5.9.4 หมายเลข F4

หน้าที่คือ ต้องสามารถเปิดประตูในกรณีฉุกเฉินได้
โดย ผู้จัดการระบบ

5.5.10 วิธีการเพื่อแก้ปัญหาโดเมน

5.5.10.1 ประเด็นที่ 1 ทำให้ระบบควบคุมได้จากผู้มีสิทธิออกรถไฟ

มีรายละเอียดคือ การควบคุมการเปิดประตูรถอยู่ภายใต้ข้อกำหนดให้สามารถเปิดได้จากผู้ที่มีสิทธิ์ ซึ่งหมายถึงนายสถานีโดยสาร หรือบุคคลอื่นที่ไม่ใช่พนักงานควบคุม การควบคุมการเปิดประตูจึงควรจะสนับสนุนให้เปิดได้หลายกรณี

แนวทางการแก้ปัญหา พัฒนาส่วนของการควบคุมประตูให้สามารถรองรับการสั่งงานเปิดประตูได้จากส่วนอื่นนอกจากห้องควบคุมภายในตัวรถ โดยควรจะเป็นตัวควบคุมและสั่งการเปิดประตูรูปแบบเดียวกัน และเพื่อให้สามารถจัดการกับส่วนควบคุมได้เหมือนกันและง่ายต่อการแก้ไขและดูแลจึงควรพัฒนาเป็นส่วนของการทำงานส่วนเดียวกันที่รองรับการสั่งงานได้หลายทาง นั่นคือกำหนดให้มีชุดคำสั่งควบคุมเป็นตัวเดียวกันในการสั่งงานควบคุมการเปิดประตู

ข้อสรุปที่ได้ พัฒนาชุดคำสั่งควบคุมในการควบคุมการเปิดประตู ที่เรียกใช้ผ่านการเชื่อมต่อที่ต่างกันได้หลายวิธี

สัมพันธ์กับความต้องการ F2

5.5.10.2 ประเด็นที่ 2 กำหนดเงื่อนไขประกอบสถานการณ์ที่จะเปิดประตู

มีรายละเอียดคือ การเปิดประตูแต่ละครั้งมีสภาพแวดล้อมหลายส่วนที่เกี่ยวข้องกับความปลอดภัยในการเปิดประตู ซึ่งในหลายๆเงื่อนไขอาจลืมนำมาพิจารณาด้วย ในขณะที่บางเงื่อนไขอาจมีความเกี่ยวข้องกับความปลอดภัยเพียงเล็กน้อย ดังนั้นจึงเป็นปัญหาที่ในการที่ผู้พัฒนาจะนำปัจจัยทั้งหมดมาทำการตรวจสอบจนครบถ้วน

แนวทางการแก้ปัญหา จะต้องมีการลำดับรายการปัจจัยที่มีส่วนเกี่ยวข้อง โดยผู้ใช้งานหรือผู้ที่มีความเชี่ยวชาญในโดเมนจะสามารถให้รายละเอียดของปัจจัยที่ควรนำมาพิจารณาได้

ข้อสรุปที่ได้ ทำรายการตรวจสอบจากผู้ใช้หรือผู้เชี่ยวชาญใน โดเมน

สัมพันธ์กับความต้องการ F3

5.5.11 ความเชื่อมโยงและข้อจำกัดของความต้องการ

5.5.11.1 ประเด็นที่ 1 สิทธิในการเปิดประตู

กับความต้องการ F1 โดยมีความสำคัญสูงสุด

กับความต้องการ F2 มีความสำคัญเทียบเท่า F4 โดยมีความสำคัญเป็นอันดับรองลงมา

มีรายละเอียดคือ สิทธิในการเปิดประตูจะถูกกำหนดขึ้นในระบบเพื่อให้มีความชัดเจนในการดำเนินงาน โดยการเปิดประตูเป็นหน้าที่พื้นฐานของผู้ควบคุมรถ แต่เพื่อเป็นการป้องกันกรณีที่มีการเปิดประตูไม่สามารถกระทำได้โดยพนักงานควบคุม หรือพนักงานควบคุมไม่อยู่ในสถานะ

พร้อมที่จะมีการสั่งงาน การควบคุมประตูดึงกำหนดให้สามารถควบคุมได้จากภายนอก ห้องควบคุมบรรดไฟ ซึ่งอาจเป็นส่วนกลางหรือห้องควบคุมในแต่ละสถานีโดยสาร และนอกจากผู้ควบคุมที่เป็นพนักงานหรือได้รับหน้าที่แล้ว ผู้โดยสารหรือพนักงานภาคสนามยังควรสามารถเปิดประตูในกรณีที่ระบบส่วนอื่นไม่สามารถทำงานอย่างปกติ โดยหน้าที่หลักในการควบคุมจะเป็นในส่วนของพนักงานควบคุมบรรดไฟ

5.5.12 มุมมองจากผู้เกี่ยวข้อง

5.5.12.1 บทบาทที่ 1 พนักงานควบคุม

มีความเห็นต่อความต้องการ F1 ดังนี้

การสั่งเปิดประตูไม่จำเป็นต้องยุ่งยากหรือมีขั้นตอนการทำงานที่ซับซ้อน เพราะเป็นแค่การสั่งให้เปิดประตูเท่านั้น ควรควบคุมได้ด้วยคำสั่ง ปุ่ม หรือ สวิตช์ เพียงตัวเดียว หากการเปิดประตูมีปัญหาหรือไม่สามารถทำงานได้โดยสมบูรณ์จะต้องแจ้งผลและให้คำแนะนำ หรือมีวิธีแก้ปัญหาเฉพาะหน้า

มีความเห็นต่อความต้องการ F2 ดังนี้

จะเกิดปัญหาหากสามารถควบคุมจากบุคคลภายนอก การเปิดประตูอาจเกิดความสับสน การเปิดประตูได้หรือไม่ควรเป็นสิทธิโดยสมบูรณ์ของพนักงานควบคุม เว้นแต่กรณีที่มีปัญหา หรือ ไม่สามารถควบคุมการเปิดประตูได้ การเปิดจึงจะมีสิทธิควบคุมจากภายนอก

มีความเห็นต่อความต้องการ F3 ดังนี้

การตรวจสอบความปลอดภัยในส่วนต่างๆก่อนจะสามารถเปิดประตูได้เป็นสิ่งที่ดี และควรเป็นหน้าที่ของระบบในการตรวจสอบเงื่อนไขทั้งหมด แต่เมื่อเงื่อนไขไม่สมบูรณ์ทำให้การสั่งงานเปิดประตูไม่สามารถทำได้ จะต้องแจ้งด้วยว่าเงื่อนไขใดที่ทำให้ไม่สามารถเปิดประตูได้

5.5.12.2 บทบาทที่ 2 ผู้โดยสาร

มีความเห็นต่อความต้องการ F4 ดังนี้

การเปิดประตูในกรณีฉุกเฉินควรทำได้โดยง่าย ไม่จำเป็นต้องเข้าใจเงื่อนไขของการเปิดประตู แต่ควรรองรับปัญหาในการเปิดประตูด้วยระบบสนับสนุนหรือแก้ปัญหาอื่นๆแทน เช่น โทรศัพท์ติดต่อ เป็นต้น

5.5.12.3 บทบาทที่ 3 พนักงานควบคุมบรรดไฟนอกกรด

มีความเห็นต่อความต้องการ F2 ดังนี้

การเปิดประตูโดยควบคุมจากภายนอกหากทำได้ควรที่จะแสดงการควบคุมโดยละเอียด โดยใช้ประกอบกับข้อมูลสถานะของรถไฟขบวนนั้นๆ และระบบควรช่วยในการทำงานที่สอดคล้องกับผู้ควบคุมรถไฟ เพื่อให้สามารถติดต่อและทำงานสัมพันธ์ได้โดยไม่สับสน

มีความเห็นต่อความต้องการ F3 ดังนี้

หากการสั่งเปิดประตูไม่ผ่านเงื่อนไขจะต้องอธิบายปัญหาที่ทำให้ไม่สามารถเปิดประตูได้

5.5.13 สถานะผลลัพธ์

ระบบควบคุมการเปิดประตูที่พิจารณาความปลอดภัยเป็นอันดับแรก ระบบควบคุมการเปิดประตูที่สามารถเปิดได้ในหลายลักษณะ ทั้งจากพนักงานควบคุม ผู้ควบคุมระบบ และผู้โดยสาร

5.5.14 ระดับการยอมรับความปลอดภัย

5.5.14.1 เงื่อนไขการยอมรับที่ 1 ประตูเปิดโดยเป็นผลให้เกิดอันตราย

ระดับความสำคัญ สำคัญยิ่ง

จัดการและรับผิดชอบโดย ผู้บริหาร

5.5.14.2 เงื่อนไขการยอมรับที่ 2 ประตูเปิดด้วยเงื่อนไขที่ไม่สมบูรณ์ แต่ยังคงสภาพที่น่าเชื่อถือได้ว่ามีความปลอดภัย

ระดับความสำคัญ ต่ำ

จัดการและรับผิดชอบโดย หัวหน้าโครงการ

5.5.15 ความต้องการความปลอดภัย

SR1: ทำการป้องกันการ การเปิดประตูขณะที่รถไฟยังไม่หยุดสมบูรณ์

SR2: ตอบสนองต่อ กรณีที่ทำการเปิดประตูโดยเงื่อนไขความปลอดภัยไม่สมบูรณ์

SR3: ทำการตรวจสอบ การเปิดประตูลงในกรณีฉุกเฉิน

SR4: ทำการป้องกัน การพยายามเปิดประตูแบบ manual ของผู้โดยสารในกรณีที่ไม

สมควร

5.5.16 นัยความปลอดภัย

5.5.16.1 สถานการณ์ที่ 1 ประตูเปิดขณะที่รถไฟยังไม่หยุด

ระดับความเป็นไปได้ มีความเป็นไปได้ (Probable)

ระดับความอันตราย หายนะ (Catastrophic)

5.5.16.2 สถานการณ์ที่ 2 ประตูเปิดไม่ตรงตำแหน่ง

ระดับความเป็นไปได้ มีความเป็นไปได้ (Probable)

ระดับความอันตราย เล็กน้อย (Marginal)

5.5.16.3 สถานการณ์ที่ 3 ประตูถูกเปิดแบบ manual ทั้งที่ไม่ปลอดภัย

ระดับความเป็นไปได้ นานๆครั้ง (Remote)

ระดับความอันตราย ชั่วอันตราย (Critical)

5.5.17 เหตุการณ์นำไปสู่อุบัติเหตุ

5.5.17.1 เหตุการณ์ที่ 1 อุปกรณ์ตรวจจับการเคลื่อนที่ของรถทำงานผิดพลาด

เนื่องจาก ความผิดพลาดจากการทำงานของฮาร์ดแวร์

5.5.17.2 เหตุการณ์ที่ 2 การตรวจสอบเงื่อนไขการเปิดประตูทำได้ไม่สมบูรณ์

เนื่องจาก ความผิดพลาดจากการทำงานของซอฟต์แวร์

5.5.17.3 เหตุการณ์ที่ 3 ทำการสั่งเปิดประตูทั้งที่อยู่ในลักษณะที่เกิดอันตรายได้

เนื่องจาก ความผิดพลาดจากการทำงานของผู้ใช้งาน

5.5.17.4 เหตุการณ์ที่ 4 มีการเปิดประตูด้วยมือ

เนื่องจาก ความผิดพลาดจากการทำงานของผู้ใช้งาน

5.5.18 อันตรายที่เกิดขึ้นได้

5.5.18.1 อันตรายที่ 1 ประตูเปิดขณะรถไฟยังวิ่งอยู่

ความเสียหายที่เกิดขึ้น ผู้โดยสารที่อยู่บริเวณประตูอาจพลัดตกจากรถไฟซึ่งส่งผลถึงชีวิตได้

5.5.18.2 อันตรายที่ 2 ประตูเปิดขึ้นในตำแหน่งที่ไม่ปลอดภัย

ความเสียหายที่เกิดขึ้น การออกจากรถไฟอาจไม่สะดวกหรือเกิดอันตรายตามมา

5.5.19 การเข้าใจการใช้งาน

ยังไม่มีการนำไปใช้จริง

5.5.20 ตัวอย่าง

ยังไม่มีตัวอย่าง

5.5.21 แบบรูปที่เกี่ยวข้อง

1) ATP: Zero Speed Detection

การหยุดรถที่สมบูรณ์แล้วเท่านั้นที่เป็นเงื่อนไขขอมให้เปิดประตูได้

5.5.22 แบบรูปกระบวนการ

1) HA: Fault Tree Analysis

การวิเคราะห์ความผิดพลาดของสัญญาณควบคุมต่างๆจะนำมาใช้ประเมินเพื่ออนุญาตเปิดประตู

5.6 แบบรูปการกำหนดตำแหน่งและความเร็วของรถไฟ

เป็นการกำหนดและติดตามตำแหน่ง ความเร็ว อัตราเร่ง และทิศทางของรถไฟให้อยู่ในขอบเขตที่ปลอดภัย ข้อมูลเกี่ยวกับตำแหน่ง ทิศทาง และความเร็ว จะถูกประเมินเพื่อหาตำแหน่งที่ถูกต้อง โดยกำหนดตำแหน่งและความเร็วที่เป็นไปในทางปลอดภัยเพื่อใช้เป็นขอบเขตในการควบคุมรถไฟ

5.6.1 ชื่อแบบรูป

การกำหนดตำแหน่งและความเร็วของรถไฟ (Train Location and Speed Determination)

5.6.2 เป้าหมาย

5.6.2.1 เป้าหมายที่ 1

รถไฟจะต้องสามารถกำหนดตำแหน่ง ความเร็ว อัตราเร่ง และทิศทางที่รถไฟเคลื่อนที่
ประเมินโดย สามารถกำหนดตำแหน่งและความเร็วของรถไฟ

5.6.2.2 เป้าหมายที่ 2

รถไฟที่อยู่ในการควบคุมจะต้องสามารถคำนวณตำแหน่งและความเร็วได้อย่างความถูกต้อง และขอมให้เกิดความคลาดเคลื่อนได้เฉพาะช่วงความปลอดภัยที่ขอมรับเท่านั้น
ประเมินโดย การคำนวณตำแหน่ง ความเร็ว อัตราเร่ง และทิศทางของรถไฟเป็นไปอย่างถูกต้องไม่คลาดเคลื่อนเกินระดับการขอมรับ

5.6.3 ปัญหาโดเมน

5.6.3.1 ปัญหาที่ 1 การกำหนดค่าความปลอดภัยที่เกี่ยวข้อง

ประเภทของปัญหา กำหนดเกี่ยวกับงาน

มีรายละเอียดคือ การกำหนดตำแหน่งและความเร็วรถไฟ จะมีความสำคัญในด้านความปลอดภัย โดยจะมีค่าที่จะต้องทำการกำหนดเพื่อควบคุม หรือข้อมูลเกี่ยวกับตำแหน่งและความเร็วที่จะมีการกำหนด ซึ่งข้อมูลเหล่านี้ผู้ที่เกี่ยวข้องอาจไม่ทราบหรือไม่รู้จริงเกี่ยวกับค่าที่ควรกำหนด

ต้นเหตุคือ ต้องศึกษาและเข้าใจในลักษณะของรางและรถไฟโดยละเอียดจึงจะสามารถกำหนดค่าที่เกี่ยวข้องได้อย่างถูกต้อง

จาก คณะกรรมการกำหนดและดูแลทางด้านความปลอดภัย

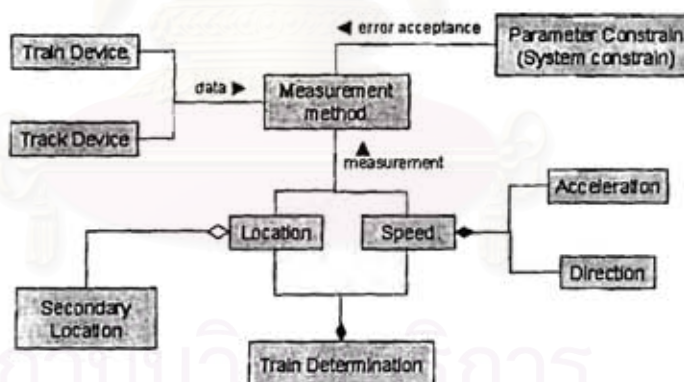
เมื่อ ทำการให้ข้อมูลเชิงเทคนิคเพื่อกำหนดทางด้านความปลอดภัย

5.6.4 ปัญหาการพัฒนา

ไม่มีการแสดงความเห็นต่อสิ่งที่จะเป็นปัญหาในการพัฒนาระบบให้เป็นไปตามความต้องการ

5.6.5 ลักษณะโดเมน

5.6.5.1 แผนภาพที่ 1 โครงสร้างการกำหนดเกี่ยวกับตำแหน่งและความเร็วของรถไฟ รูปที่ 5.4



รูปที่ 5.4 โครงสร้างการกำหนดเกี่ยวกับตำแหน่งและความเร็วของรถไฟ

คำอธิบาย แผนภาพแสดงรายละเอียดของส่วนที่เกี่ยวข้องกับการกำหนดตำแหน่งและความเร็ว รายละเอียดสำคัญที่จะต้องทำการกำหนด และแสดงการคำนวณหาตำแหน่งและความเร็ว โดยอธิบายไปยังองค์ประกอบข้อมูลต่างๆที่ต้องนำมาใช้

5.6.6 สภาพแวดล้อม

ไม่มีสภาพแวดล้อมภายนอกที่ส่งผลต่อการทำงานของระบบ

5.6.7 แรงชักจูง

- 1) รถไฟสามารถควบคุมตัวรถได้แบบอัตโนมัติหรือกึ่งอัตโนมัติ
- 2) ระบบมีการตรวจสอบตำแหน่งและความเร็วของรถไฟ

5.6.8 ข้อจำกัดและกฎเกณฑ์

- 1) ผลลัพธ์ในการประเมินตำแหน่งรถไฟ (+/- 0.25 ถึง +/- 6.25 เมตร)
- 2) ความแม่นยำในการวัดตำแหน่งรถไฟในกรณีปกติ (+/- 5 ถึง +/- 10 เมตร)
- 3) ผลลัพธ์ในการประเมินความเร็วรถไฟ (+/- 0.5 km/h ถึง +/- 2 km/h)
- 4) ความแม่นยำในการวัดความเร็วรถไฟ (+/- 3 km/h)

5.6.9 ความต้องการ

5.6.9.1 หมายเลข F1

หน้าที่คือ จะต้องควบคุมตำแหน่ง ความเร็ว และทิศทางได้

โดย พนักงานควบคุม

5.6.9.2 หมายเลข F2

หน้าที่คือ จะต้องแสดงตำแหน่ง ความเร็ว และทิศทางได้

โดย พนักงานควบคุม

5.6.9.3 หมายเลข F3

หน้าที่คือ จะต้องเริ่มทำงานโดยอัตโนมัติเมื่อรถไฟเข้าสู่ระบบ

โดย ผู้จัดการระบบ

5.6.9.4 หมายเลข F4

หน้าที่คือ จะต้องเริ่มทำงานโดยอัตโนมัติเมื่อรถไฟหรือระบบกลับคืนสู่สถานะปกติ

หลังจากเกิดเหตุขัดข้อง

โดย ผู้จัดการระบบ

5.6.9.5 หมายเลข NF1

ความต้องการคุณสมบัติของระบบ จะต้องพิจารณาความคลาดเคลื่อนที่เกิดขึ้นได้
ประกอบการคำนวณด้วย

โดย ผู้จัดการระบบ

5.6.9.6 หมายเลข NF2

ความต้องการคุณสมบัติของระบบ จะต้องกำหนดตำแหน่งทั้งหน้าและหลังรถไฟอย่างถูกต้อง
โดย ผู้จัดการระบบ

5.6.10 การเสนอวิธีการเพื่อแก้ปัญหาโดเมน

5.6.10.1 ประเด็นที่ 1 รถไฟที่ต้องมีการกำหนดตำแหน่งและความเร็ว

มีรายละเอียดคือ นอกจากรถไฟที่มีการควบคุมติดตามการทำงานโดยสมบูรณ์ ยังมีรถไฟรวมไปถึงวัตถุขับเคลื่อนบนรางประเภทอื่นๆอีกที่ไม่ได้มีการควบคุมติดตามการทำงาน ซึ่งการกำหนดตำแหน่งและความเร็วจะใช้กับรถไฟที่ถูกติดตามควบคุมการทำงานโดยสมบูรณ์เท่านั้น ดังนั้นการติดตามควบคุมในการกำหนดตำแหน่งและความเร็วจึงจะเริ่มต้นเมื่อรถไฟมีความพร้อมและสามารถเข้าสู่ระบบโดยไม่รบกวนการทำงานที่มีอยู่เดิม

แนวทางการแก้ปัญหา ใช้ลักษณะการทำงานแบบเซต โดยพิจารณารถไฟในเซตของการติดตามและควบคุม ทำการกำหนดเงื่อนไขของการเพิ่มสมาชิกเข้ามาและการนำออกจากเซตในการติดตามและควบคุม โดยรถไฟที่จัดอยู่ในเซตจะมีการกำหนดตำแหน่งและความเร็วซึ่งระบบจะทำการคำนวณค่าเฉพาะในเซตอย่างอัตโนมัติ

ข้อสรุปที่ได้ พิจารณาการเพิ่มและนำออกจากการกำกับ โดยระบบทำการคำนวณปรับปรุงค่าเฉพาะที่ถูกกำกับอย่างอัตโนมัติ

สัมพันธ์กับความต้องการ F3 F4

5.6.10.2 ประเด็นที่ 2 ตำแหน่ง ความเร็ว อัตราเร่ง และทิศทาง ในขณะที่ใดขณะหนึ่ง

มีรายละเอียดคือ การคำนวณตำแหน่ง ความเร็ว อัตราเร่ง และทิศทาง สามารถทำได้ในขณะเวลาหนึ่ง ซึ่งในการติดตามค่าเหล่านี้ก็จำเป็นต้องมีการปรับปรุงให้ค่าเป็นปัจจุบันที่สุด ซึ่งมีปัญหาในการคำนวณค่าจากเวลาที่เหมาะสม ความละเอียดและความแม่นยำที่ต้องการ

แนวทางการแก้ปัญหา การคำนวณค่าที่นำมาใช้จะต้องเป็นไปตามที่มีการกำหนดไว้ ซึ่งเมื่อรถไฟเปลี่ยนตำแหน่งจากเดิมเกินกว่าช่วงของค่าที่กำหนดจะต้องมีการคำนวณค่าในเวลานั้นใหม่ เช่นเดียวกับความเร็วก็จะพิจารณาในลักษณะเดียวกัน ส่วนค่าอื่นๆก็จะมีการคำนวณใหม่ควบคู่กันไปด้วยเสมอ ซึ่งการทำเช่นนี้จะทำให้ไม่ต้องคำนวณค่าใหม่ตลอดเวลาหากไม่มีการเปลี่ยนแปลง

ข้อสรุปที่ได้ ใช้การคาดการณ์ค่าในช่วงที่กำหนดเป็นกลไกในการคำนวณค่าใหม่ โดยพิจารณาความคลาดเคลื่อนประกอบด้วย

สัมพันธ์กับความต้องการ F1 NF1

5.6.11 ความเชื่อมโยงและข้อจำกัดของความต้องการ

5.6.11.1 ประเด็นที่ 1 การเริ่มติดตามและกำกับตำแหน่งและความเร็วรถไฟ

กับความต้องการ F1 มีความสำคัญเทียบเท่า F2 โดยมีความสำคัญสูงสุด

กับความต้องการ F3 โดยมีความสำคัญเป็นอันดับรองลงมา

มีรายละเอียดคือ รถไฟจะถูกติดตามและกำกับเมื่อเข้าสู่ส่วนควบคุมแล้วเท่านั้น หากยังไม่มีการนำเข้าสู่ระบบการควบคุมก็จะยังไม่ดำเนินการ

5.6.11.2 ประเด็นที่ 2 การคำนวณหาตำแหน่งและความเร็วที่ถูกต้อง

กับความต้องการ F1 NF1 และ NF2

มีรายละเอียดคือ การคำนวณหาตำแหน่งและความเร็วนั้นจะต้องพิจารณาความคลาดเคลื่อนที่เกิดขึ้นทั้งหมดประกอบเสมอ

5.6.12 มุมมองจากผู้เกี่ยวข้อง

5.6.12.1 บทบาทที่ 1 พนักงานควบคุม

มีความเห็นต่อความต้องการ F3 ดังนี้

การนำเข้รถไฟเพื่อทำการติดตามและควบคุมตำแหน่งและความเร็วรถไฟ ควรจะดำเนินการโดยอัตโนมัติเมื่อเดินรถ โดยต้องมีการแจ้งให้เข้าสู่การกำกับตำแหน่งและความเร็ว แล้วพนักงานที่ควบคุมจะสามารถเลือกให้มีการดำเนินการหรือไม่ดำเนินการกำกับก็ได้

มีความเห็นต่อความต้องการ F4 ดังนี้

เมื่อรถกลับเข้าสู่สถานะปกติระบบควรดำเนินการในการนำเข้รถไฟเพื่อติดตามและควบคุมตำแหน่งโดยอัตโนมัติให้เป็นปกติเหมือนกับการดำเนินการก่อนหน้านี้ นั่นคือหากมีการดำเนินการกำกับก่อนที่จะหยุดการกำกับเพราะสาเหตุใดๆ เมื่อรถไฟสามารถกลับสู่การทำงานตามเดิมได้โดยสมบูรณ์ก็ต้องถูกกำกับโดยอัตโนมัติทันที

5.6.13 สถานะผลลัพธ์

ระบบป้องกันรถไฟอัตโนมัติจะมีการควบคุมและติดตาม ตำแหน่ง ความเร็ว อัตราเร่ง และทิศทางของรถไฟได้ ระบบจะมีความสามารถในการป้องกันอันตรายจากการใช้รางร่วมกัน

5.6.14 ระดับการยอมรับความปลอดภัย

5.6.14.1 เงื่อนไขการยอมรับที่ 1 การวัดตำแหน่งและความเร็วของรถไฟคลาดเคลื่อนเกินที่กำหนด

ระดับความสำคัญ สำคัญยิ่ง

จัดการและรับผิดชอบโดย ระบบดำเนินการ

5.6.14.2 เงื่อนไขการยอมรับที่ 2 การกำหนดค่าประเมินความคลาดเคลื่อนสำหรับรถไฟมี
ความผิดพลาด

ระดับความสำคัญ สำคัญยิ่ง

จัดการและรับผิดชอบโดย การกำหนดระบบงาน

5.6.15 ความต้องการความปลอดภัย

SR1: ทำการป้องกัน การคำนวณตำแหน่งรถไฟผิดพลาด

SR2: ทำการป้องกัน การคำนวณความเร็วรถไฟผิดพลาด

5.6.16 นัยความปลอดภัย

5.6.16.1 สถานการณ์ที่ 1 การกำหนดตำแหน่งรถไฟผิดพลาด

ระดับความเป็นไปได้ มีความเป็นไปได้ (Probable)

ระดับความอันตราย ชั่วอันตราย (Critical)

5.6.16.2 สถานการณ์ที่ 2 การกำหนดความเร็วรถไฟผิดพลาด

ระดับความเป็นไปได้ มีความเป็นไปได้ (Probable)

ระดับความอันตราย ชั่วอันตราย (Critical)

5.6.17 เหตุการณ์นำไปสู่อุบัติเหตุ

5.6.17.1 เหตุการณ์ที่ 1 ตัวตรวจจับความเร็วของรถทำงานผิดพลาด

เนื่องจาก ความผิดพลาดจากการทำงานของฮาร์ดแวร์

5.6.17.2 เหตุการณ์ที่ 2 ตัวตรวจจับตำแหน่งของรถทำงานผิดพลาด

เนื่องจาก ความผิดพลาดจากการทำงานของฮาร์ดแวร์

5.6.17.3 เหตุการณ์ที่ 3 การคำนวณตำแหน่งมีความคลาดเคลื่อน

เนื่องจาก ความผิดพลาดจากการทำงานของซอฟต์แวร์

5.6.17.4 เหตุการณ์ที่ 4 การคำนวณความเร็วมีความคลาดเคลื่อน

เนื่องจาก ความผิดพลาดจากการทำงานของซอฟต์แวร์

5.6.18 อันตรายที่เกิดขึ้นได้

5.6.18.1 อันตรายที่ 1 รถไฟอยู่ในตำแหน่งที่ไม่เหมาะสม

ความเสียหายที่เกิดขึ้น ทำให้รถไฟมีโอกาที่จะชนกัน

5.6.18.2 อันตรายที่ 2 รถไฟมีความเร็วที่ไม่เหมาะสม

ความเสียหายที่เกิดขึ้น รถไฟมีความเสี่ยงที่จะเกิดการชนหรือควกราง

5.6.18.3 อันตรายที่ 3 รถไฟไม่สามารถไปยังตำแหน่งอื่นได้

ความเสียหายที่เกิดขึ้น ระบบจะชะงักไม่สามารถดำเนินการต่อไปได้ หรือถ้ามีการดำเนินการต่อก็จะก่อให้เกิดอันตรายตามมา

5.6.19 การเข้าใจการใช้งาน

ยังไม่มีให้นำไปใช้

5.6.20 ตัวอย่าง

ยังไม่มีตัวอย่าง

5.6.21 แบบรูปที่เกี่ยวข้อง

1) ATP: Route Interlocking

ตำแหน่งของรถไฟสามารถควบคุมและเปลี่ยนแปลงได้จากการกำหนดการใช้ราง
ซึ่งจะเป็นตัวกำหนดความเร็วและทิศทางโดยอัตโนมัติ

5.6.22 แบบรูปกระบวนการ

1) HA: Fault Tree Analysis

พิจารณาความผิดพลาดของการกำหนดตำแหน่งและความเร็วในเงื่อนไขต่างๆ
เพื่อหาผลร้ายที่อาจตามมา

2) Cross-check Operation and Function Requirements to Safety

ตรวจสอบการดำเนินการและหน้าที่ที่เกี่ยวข้อง หน้าที่ทางด้านการปลอดภัยของ
การกำหนดตำแหน่งและความเร็วเป็นผลโดยตรงจากการสั่งการควบคุมรถ

บทที่ 6

การออกแบบกระบวนการในการใช้แบบรูปความต้องการ

เพื่อให้การใช้งานแบบรูปความต้องการเกิดประสิทธิภาพสูงสุด ในบทนี้จึงได้พิจารณาการทำงานร่วมกับกระบวนการวิศวกรรมซอฟต์แวร์ในส่วนต่างๆ โดยให้ความสำคัญกับการพัฒนาทั้งในแบบน้ำตกและแบบทำซ้ำ และยังได้เสนอแนวทางในการทำงานร่วมกับกระบวนการและแบบรูปประเภทอื่นๆ ในกระบวนการวิศวกรรมซอฟต์แวร์ รวมทั้งได้เสนอการนำไปใช้งานร่วมกับกระบวนการพัฒนาซอฟต์แวร์สำหรับระบบปลอดภัยเชิงวิกฤต

6.1 วิศวกรรมความต้องการซอฟต์แวร์ในกระบวนการพัฒนาซอฟต์แวร์

กระบวนการวิศวกรรมความต้องการซอฟต์แวร์ถือได้ว่าอยู่ในช่วงต้นของการพัฒนาซอฟต์แวร์ในการพัฒนาแบบน้ำตก (Waterfall Model) และอยู่ช่วงต้นของเฟสในการพัฒนาแบบทำซ้ำ (Iterative Model) เพื่อให้ครอบคลุมรูปแบบของการใช้งานแบบรูป จึงพิจารณากระบวนการวิศวกรรมความต้องการซอฟต์แวร์ที่สอดคล้องกับรูปแบบการพัฒนาซอฟต์แวร์ทั้ง 2 ลักษณะ

การพัฒนาซอฟต์แวร์ทั้ง 2 แบบ มีกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ที่แตกต่างกันเล็กน้อย โดยมีรายละเอียดของกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ที่สอดคล้องในแต่ละแบบสรุปได้ ดังนี้

6.1.1 กระบวนการวิศวกรรมความต้องการซอฟต์แวร์ที่สอดคล้องกับการพัฒนาแบบน้ำตก

- 1) เป็นขั้นแรกสุดของการพัฒนาซอฟต์แวร์
- 2) เป็นการเก็บและสรุปความต้องการทั้งหมดให้มีความสมบูรณ์ก่อนจะไปสู่การวิเคราะห์ระบบ
- 3) เมื่อเกิดความผิดพลาดในขั้นตอนถัดไปจะทวนย้อนกลับมาเพื่อแก้ไข

6.1.2 กระบวนการวิศวกรรมความต้องการซอฟต์แวร์ที่สอดคล้องกับการพัฒนาแบบทำซ้ำ

- 1) กระบวนการวิศวกรรมความต้องการซอฟต์แวร์เริ่มขึ้นช่วงต้นของการพัฒนาในแต่ละเฟส
- 2) กระบวนการวิศวกรรมความต้องการซอฟต์แวร์มีความสำคัญมากกว่ากระบวนการอื่นๆ ในเฟสแรกๆ
- 3) การเก็บความต้องการอาจไม่สมบูรณ์ในครั้งเดียว

- 4) เมื่อเกิดความผิดพลาดในส่วนของการต้องการจะยกไปเป็นเงื่อนไขของกระบวนการวิศวกรรมความต้องการในเฟสถัดไป

ก่อนจะถึงกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ อาจมีกระบวนการวิเคราะห์กิจกรรม (Business Analysis) ซึ่งจะเป็นการศึกษาลักษณะของกิจกรรมการทำงาน โดยการวิเคราะห์กิจกรรมจะช่วยให้กระบวนการวิศวกรรมความต้องการซอฟต์แวร์ทำได้ง่ายขึ้น โดยผลลัพธ์ที่ได้จากกระบวนการวิเคราะห์กิจกรรมจะนำไปใช้สนับสนุนกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ และหลังจากกระบวนการวิศวกรรมความต้องการซอฟต์แวร์จะเป็นส่วนของการวิเคราะห์ระบบ ดังนั้นข้อมูลที่ได้จากการเก็บความต้องการจะถูกนำมาใช้ในส่วนนี้ เพื่อวิเคราะห์ว่าระบบจะต้องมีลักษณะอย่างไร และนำไปสู่กระบวนการออกแบบต่อไป

6.2 การใช้แบบรูปความต้องการในวิศวกรรมความต้องการซอฟต์แวร์

การใช้แบบรูปความต้องการ ในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์จะไม่ทำให้เกิดความเปลี่ยนแปลงในส่วนของการงานแบบเดิม เพราะแบบรูปความต้องการเป็นเครื่องมือช่วยในการเก็บและใช้ข้อมูลหรือองค์ความรู้ในงานวิศวกรรมความต้องการซอฟต์แวร์ ไม่ได้รวมเอากระบวนการทำงานเข้ามาด้วยจึงไม่มีความจำเป็นต้องปรับเปลี่ยนวิธีการ ในการดำเนินงาน

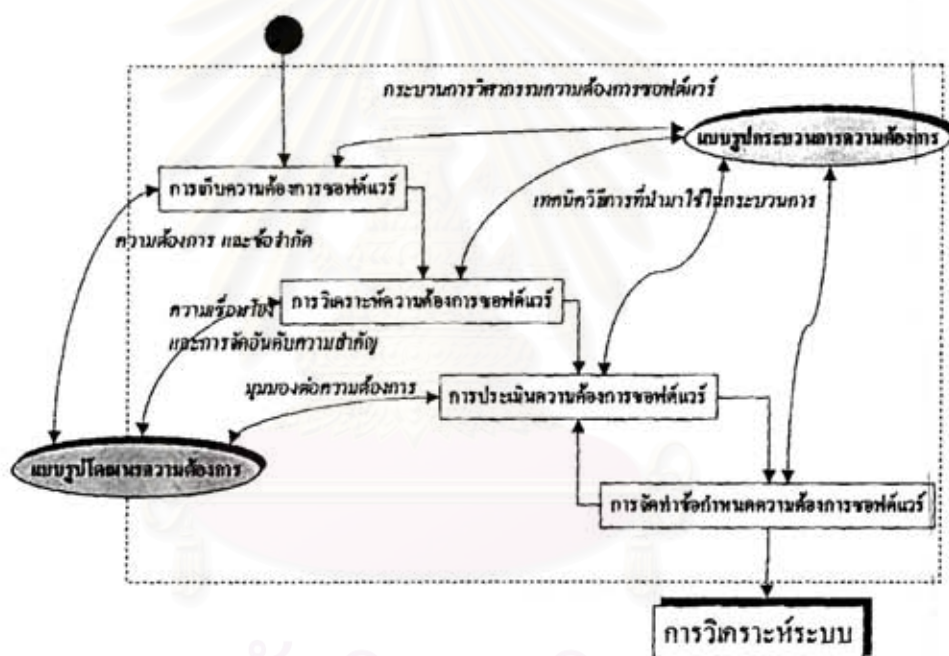
6.2.1 กระบวนการวิศวกรรมความต้องการซอฟต์แวร์ที่สอดคล้องกับการใช้แบบรูปความต้องการ

ในกระบวนการวิศวกรรมความต้องการหากพิจารณาการนำแบบรูปความต้องการมาใช้ประกอบด้วยจะมีความเกี่ยวข้องในขั้นตอนต่างๆ ดังนี้

- 1) การเก็บรวบรวมความต้องการ เมื่อมีการเก็บรวบรวมความต้องการซอฟต์แวร์ จะสามารถใช้แบบรูปโดเมนความต้องการในการอ้างอิงความต้องการ ผู้เกี่ยวข้อง ข้อจำกัด และความเชื่อมโยงของความต้องการได้
- 2) เมื่อประสบปัญหาในการทำงาน สามารถศึกษาแนวทางการแก้ปัญหาตามแบบรูปความต้องการที่เกี่ยวข้อง หากเป็นปัญหาในเรื่องของลักษณะระบบหรือตัวความต้องการจะอาศัยแบบรูปโดเมนความต้องการเป็นตัวช่วย และหากมีปัญหในเรื่องของกระบวนการหรือเทคนิควิธีที่นำมาใช้ก็สามารถศึกษาได้จากแบบรูปกระบวนการความต้องการ
- 3) เมื่อเก็บรวบรวมความต้องการเสร็จสิ้น ควรจะบันทึกความต้องการ ในแบบรูปโดเมนความต้องการ เพื่อใช้กับระบบที่คล้ายกันในครั้งต่อไป
- 4) เมื่อแก้ปัญหาที่เกิดขึ้นใหม่ ควรบันทึกลักษณะปัญหา โดยสร้างแบบรูปความต้องการขึ้นใหม่ หรือบันทึกเพิ่มเติมในแบบรูปความต้องการเดิมที่สอดคล้องกัน

- 5) เมื่อมีการนำเทคนิควิธีการใหม่ๆ มาใช้ ควรบันทึกเทคนิควิธีการนั้นลงในแบบรูปกระบวนการความต้องการ และอธิบายสาเหตุการใช้เทคนิควิธีการนั้นในแบบรูปโดเมนความต้องการ
- 6) เมื่อมีการใช้แบบรูปความต้องการให้ ทำการบันทึกผลการใช้ แกะไขความผิดพลาดในแบบรูป บันทึกผลที่ได้สำหรับแบบรูปที่ยังไม่สมบูรณ์ ควรทำการเพิ่มเติมและลงผลความสำเร็จในการใช้งานเอาไว้

เมื่อพิจารณาการทำงานจะเห็นว่าการใช้งานแบบรูปความต้องการจะแทรกอยู่ในการทำงานซึ่งสรุปได้ดังรูปที่ 6.1



รูปที่ 6.1 การใช้แบบรูปความต้องการในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์

รูปที่ 6.1 แสดงให้เห็นว่าในขั้นตอนการเก็บความต้องการ การวิเคราะห์ความต้องการ การประเมินความต้องการ และการจัดทำข้อกำหนดความต้องการ สามารถนำแบบรูปกระบวนการความต้องการมาใช้สนับสนุนในแง่ของเทคนิคและวิธีการที่จะนำมาใช้ในการทำงานในส่วนต่างๆ ได้ ในทางกลับกัน ปัญหาในการปฏิบัติที่เกิดขึ้นในกระบวนการเหล่านี้ก็สามารถนำไปสร้างเป็นแบบรูปกระบวนการความต้องการได้

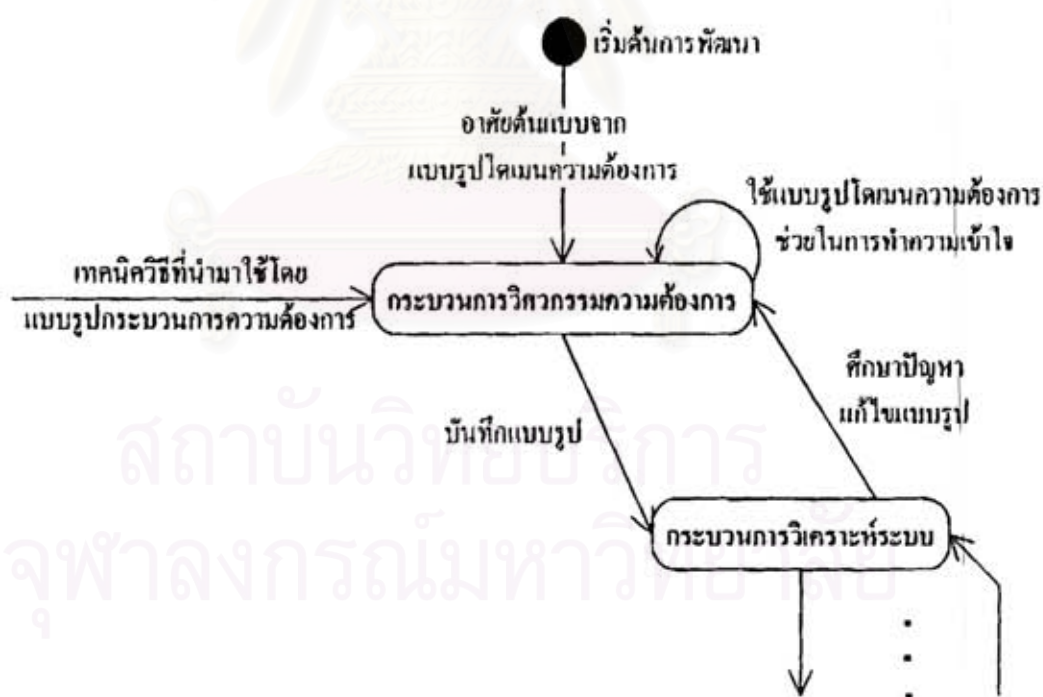
รูปที่ 6.1 แสดงให้เห็นว่าในกระบวนการการเก็บความต้องการสามารถนำความต้องการและข้อจำกัดที่แบบรูปโดเมนความต้องการเสนอไว้มาใช้ได้โดยตรง ในกระบวนการวิเคราะห์ความต้องการจะอาศัยข้อมูลองค์ประกอบความเชื่อมโยงและข้อจำกัดของความต้องการ ในแบบรูปโดเมนความต้องการมา

ใช้พิจารณาได้ และในกระบวนการประเมินความต้องการจะนำข้อมูลจากองค์ประกอบผู้เกี่ยวข้องและมุมมองต่างๆ ในแบบรูปโดเมนความต้องการมาใช้เป็นส่วนประกอบการวิเคราะห์ ซึ่งปัญหาหรือผลลัพธ์ที่เกิดขึ้นในกระบวนการเหล่านี้จะนำมาใช้เป็นข้อมูลบันทึกกลับไปในแบบรูปโดเมนความต้องการ

6.2.2 การใช้แบบรูปความต้องการในกระบวนการพัฒนาซอฟต์แวร์แบบน้ำตก

การทำงานจะเริ่มจากการหาแบบรูปโดเมนความต้องการที่สอดคล้องกับลักษณะของระบบ ในระหว่างกระบวนการวิศวกรรมความต้องการสามารถนำแบบรูปโดเมนความต้องการมาช่วยในแง่ของการพิจารณาปัญหาของโดเมน รายละเอียดของข้อจำกัด และความต้องการ และในแง่ของกลวิธีในการทำงานจะนำแบบรูปความต้องการมาช่วยเหลือได้

เมื่อกระบวนการวิศวกรรมความต้องการเสร็จสิ้นลงควรที่จะบันทึกข้อมูลในการทำงานเพื่อนำไปใช้ในครั้งต่อไป และหากมีปัญหาเกิดขึ้นกับในการทำงานก็ควรกลับมาบันทึกหรือแก้ไขความถูกต้องของแบบรูปความต้องการ โดยลักษณะการใช้งานแบบรูปความต้องการในกระบวนการพัฒนาซอฟต์แวร์แบบน้ำตกสามารถแสดงได้ดังรูปที่ 6.2



รูปที่ 6.2 การใช้งานแบบรูปความต้องการในกระบวนการวิศวกรรมซอฟต์แวร์แบบน้ำตก

นอกจากนี้หากกระบวนการพัฒนาซอฟต์แวร์มีการวิเคราะห์กิจกรรม ผลของการวิเคราะห์กิจกรรมสามารถนำมาใช้ในการเลือกแบบรูปโดเมนความต้องการได้

6.2.3 การใช้แบบรูปความต้องการในกระบวนการพัฒนาซอฟต์แวร์แบบทำซ้ำ

การทำงานจะเริ่มต้นจากการหาแบบรูปโดเมนความต้องการที่สอดคล้องกับลักษณะของระบบเพื่อใช้ในเฟสแรกของการพัฒนา ในแต่ละเฟสสามารถใช้แบบรูปโดเมนความต้องการช่วยในการทำงานเพื่อปรับความต้องการจากเฟสก่อนหน้าให้ถูกต้องสมบูรณ์ แต่เมื่อเป็นปัญหาที่ไม่มีแนวทางเสนอไว้ก่อน อาจนำแบบกระบวนการความต้องการมาใช้เป็นแนวทางเลือกเทคนิคหรือวิธีการเพื่อแก้ปัญหาในส่วนนี้ ถ้าวิธีการที่นำมาใช้ประสบผลสำเร็จจะทำการบันทึกเสนอเป็นแบบรูปกระบวนการสำหรับแบบรูปโดเมนความต้องการนั้น เมื่อมีการแก้ไขปัญหาก็จะทำการบันทึกหรือแก้ไขแบบรูปความต้องการให้มีความถูกต้องตาม ลักษณะการใช้งานแบบรูปความต้องการสามารถแสดงได้ดังรูปที่ 6.3



รูปที่ 6.3 การใช้แบบรูปความต้องการในกระบวนการวิศวกรรมซอฟต์แวร์แบบทำซ้ำ

กระบวนการวิเคราะห์กิจกรรมจะทำการพัฒนาซอฟต์แวร์แบบทำซ้ำทุกเฟสก่อนกระบวนการวิศวกรรมความต้องการ ดังนั้นการวิเคราะห์กิจกรรมนอกจากจะนำมาช่วยในการเลือกใช้แบบรูปโดเมนความต้องการแล้วยังสามารถใช้เป็นเครื่องมือตรวจสอบการใช้แบบรูปโดเมนความต้องการได้ด้วย โดยองค์ประกอบสถานะผลลัพธ์ของแบบรูปโดเมนความต้องการควรสอดคล้องกับสิ่งที่ได้จากการวิเคราะห์กิจกรรม

6.2.4 การใช้แบบรูปความต้องการสำหรับระบบปลอดภัยเชิงวิกฤต

สำหรับกระบวนการวิศวกรรมความต้องการสำหรับระบบปลอดภัยเชิงวิกฤตจะมีความแตกต่างจากระบบทั่วไปในส่วนของการให้ความสำคัญกับปัจจัยที่จะก่อให้เกิดอุบัติเหตุหรืออันตราย และการพิจารณาความต้องการ ในระบบปลอดภัยเชิงวิกฤตจะเน้นไปที่ภาพรวมในระดับระบบไม่ได้พิจารณาเฉพาะด้านซอฟต์แวร์เพียงอย่างเดียว

ในกระบวนการความต้องการซอฟต์แวร์สำหรับระบบปลอดภัยเชิงวิกฤต สามารถนำแบบรูปความต้องการส่วนปลอดภัยมาใช้ร่วมในส่วนการวิเคราะห์อันตราย เนื่องจากเป็นการวิเคราะห์เหตุการณ์

ลักษณะการทำงาน และความบกพร่องที่เกิดขึ้นขณะดำเนินการ ซึ่งในแบบรูปความต้องการส่วนความพลอดภัยจะมีข้อมูลที่เป็นประโยชน์สามารถนำมาใช้ในการวิเคราะห์อันตรายได้ เช่น องค์ประกอบอุบัติเหตุสามารถนำมาใช้ในการวิเคราะห์ต้นไม้อุบัติเหตุซึ่งเทคนิคของการวิเคราะห์อันตรายแบบหนึ่ง การวิเคราะห์อันตรายจะเกิดขึ้นตลอดทุกช่วงของการพัฒนาระบบ นั้นแสดงว่าเราสามารถนำข้อมูลส่วนนี้มาใช้ได้ตลอดเวลา

6.3 การใช้แบบรูปความต้องการร่วมกับแบบรูปประเภทอื่นๆ

การพัฒนาซอฟต์แวร์ในปัจจุบันให้ความสำคัญกับการนำกลับมาใช้ใหม่อย่างมาก ซึ่งแบบรูปในวิทยานิพนธ์นี้ก็จะเป็นลักษณะการนำกลับมาใช้ใหม่แบบหนึ่ง แต่แนวคิดของแบบรูปนั้นเป็นเรื่องของการแก้ปัญหาที่เกิดขึ้นในส่วนหนึ่งส่วนใด การใช้งานที่เป็นระบบและครอบคลุมกระบวนการในการพัฒนาซอฟต์แวร์จึงอาจไม่สมบูรณ์ หรือไม่มีประสิทธิภาพเท่าที่ควร แม้จะสามารถปรับใช้กับกระบวนการพัฒนาซอฟต์แวร์ที่มีอยู่เดิมได้ตามที่เสนอในข้อ 6.2 ก็ยังขาดความต่อเนื่องในวงจรการพัฒนาซอฟต์แวร์ ดังนั้นในหัวข้อนี้จึงมีแนวคิดในการพิจารณาใช้งานแบบรูปความต้องการควบคู่กับแบบรูปประเภทอื่นๆ ได้แก่ แบบรูปธุรกิจ และแบบรูปการวิเคราะห์ เพื่อให้เกิดความต่อเนื่องในการนำกลับมาใช้ใหม่ที่เป็นระบบมากขึ้น ซึ่งมีแนวทางสรุปได้ ดังนี้

6.3.1 การใช้งานร่วมกับแบบรูปธุรกิจ (Business Pattern) [29]

แบบรูปธุรกิจเสนอลักษณะสภาพแวดล้อมที่เป็นอยู่ในการทำงาน ดังนั้นการใช้แบบรูปธุรกิจจะช่วยให้เข้าใจระบบได้เร็วขึ้น และสามารถนำมาใช้เป็นเครื่องมือค้นหาความต้องการของระบบนั้นโดยผ่านแบบรูปความต้องการได้

องค์ประกอบลักษณะโดเมนเป็นส่วนที่อธิบายลักษณะสภาพแวดล้อมเช่นเดียวกัน สิ่งที่เสนอในแบบรูปธุรกิจสามารถนำมาใช้กับองค์ประกอบลักษณะโดเมนได้ หรืออาจสรุปได้ว่าผลลัพธ์จากการใช้แบบรูปธุรกิจจะนำมาใช้เปรียบเทียบเพื่อหาแบบรูปโดเมนความต้องการที่มีองค์ประกอบลักษณะโดเมนที่ตรงกัน

6.3.2 การใช้งานร่วมกับแบบรูปการวิเคราะห์ [15]

แบบรูปการวิเคราะห์เสนอปัญหาที่เกิดขึ้นในการวิเคราะห์ระบบ ซึ่งการจะวิเคราะห์ระบบที่สมบูรณ์นั้นต้องมีความเข้าใจในความต้องการและการทำงาน โดยการวิเคราะห์เป็นการหาแนวทางในการพัฒนาซอฟต์แวร์ให้เป็นไปตามความต้องการ โดยพิจารณาความเป็นไปได้และความเหมาะสมในการพัฒนา ซึ่งในเบื้องต้นนั้นแบบรูปความต้องการก็ได้พยายามเสนอลักษณะปัญหาที่ทำให้ความต้องการนั้นๆ ไม่สามารถนำมาปฏิบัติได้หรือไม่มีความเหมาะสม

จากแบบรูปโคเมนความต้องการจะเห็นว่าการจัดประเภทแบบรูปจะเป็นไปตามกลุ่มความต้องการที่ทำงานสัมพันธ์กันอย่างใกล้ชิด และแสดงความเชื่อมโยงหรือความสัมพันธ์กับกลุ่มความต้องการหรือข้อจำกัดอื่นๆ ซึ่งลักษณะเช่นนี้ทำให้เกิดความสะดวกในการวิเคราะห์ระบบ ทำให้ทราบความเกี่ยวข้องและความอิสระของการทำงาน ดังนั้นหากมีการใช้แบบรูปการวิเคราะห์ควบคู่กับแบบรูปความต้องการแบบรูปการวิเคราะห์จะสามารถดึงเอาเฉพาะส่วนที่เป็นข้อสรุปความต้องการและข้อจำกัดมาใช้ในการเลือกแบบรูปการวิเคราะห์ได้



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 7

การพัฒนาเครื่องมือเพื่อช่วยในการสร้างและใช้งานแบบรูปความต้องการ

ในบทนี้จะเป็นการนำเสนอการพัฒนาเครื่องมือที่สร้างขึ้นเพื่อช่วยในการสร้างและใช้งานแบบรูปความต้องการ โดยเป็นการแสดงรายละเอียดที่เกี่ยวข้องกับกระบวนการพัฒนา ข้อกำหนดความต้องการ การวิเคราะห์และออกแบบ ตลอดจนทำการทดสอบเครื่องมือ และในตอนท้ายของบทจะเป็นการแสดง ส่วนต่อประสานของเครื่องมือที่พัฒนา

7.1 สภาพแวดล้อมและกระบวนการในการพัฒนาเครื่องมือ

7.1.1 สภาพแวดล้อมทางฮาร์ดแวร์

- 1) เครื่องคอมพิวเตอร์ส่วนบุคคล (PC) หน่วยประมวลผลอินเทลเพนเทียมโฟร์ 2.60 GHz
กะเฮิรตซ์ (Intel Pentium4 2.6 GHz)
- 2) หน่วยความจำหลัก (RAM) 512 เมกะไบต์ (512 MB)
- 3) ฮาร์ดดิสก์ (Hard Disk) 40 กิกะไบต์ (40 GB)

7.1.2 สภาพแวดล้อมทางซอฟต์แวร์

- 1) ระบบปฏิบัติการ (Operating System) ไมโครซอฟท์วินโดวส์เอ็กซ์พี โพรเฟชันแนล (Microsoft Windows XP Professional)
- 2) เอ็กซ์เอ็มแอลแพด 3.0 โปรอีดิชัน (XMLPad 3.0 Pro Edition) เพื่อใช้ในการจัดการกับเอกสาร เอ็กซ์เอ็มแอล (XML) เอ็กซ์เอ็มแอลสกีมา (XML Schema) และ เอ็กซ์เอสแอลสไตล์ชีต (XSL Style Sheet)
- 3) ไมโครซอฟท์วิสวลสตูดิโอไอคอตเน็ต รุ่น 2003 (Microsoft Visual Studio .NET 2003) เพื่อใช้ในการพัฒนาภาษาซีชาร์ป (C#)
- 4) ไมโครซอฟท์คอตเน็ตเฟรมเวิร์ค รุ่น 1.1 (Microsoft .NET Framework 1.1) ซึ่งเป็นเฟรมเวิร์คที่การรันโปรแกรม
- 5) ไมโครซอฟท์อินเทอร์เน็ตเอ็กโพลอเรอร์ รุ่น 6.0 (Microsoft Internet Explorer 6.0)

7.1.3 กระบวนการในการพัฒนาเครื่องมือ

การพัฒนาเครื่องมือจะใช้วิธีการพัฒนาแบบน้ำตก โดยเริ่มจากการกำหนดขอบเขตและกำหนดหน้าที่ความต้องการของเครื่องมือ แล้วจึงนำข้อกำหนดที่ได้มาวิเคราะห์และออกแบบเพื่อ

พัฒนา และในส่วนท้ายสุดจะทำการทดสอบเครื่องมือว่าสามารถทำงานได้ถูกต้องและเป็นไปตามที่กำหนดไว้หรือไม่

7.2 ข้อกำหนดและความสามารถของเครื่องมือ

7.2.1 ความต้องการที่เป็นหน้าที่

7.2.1.1 ต้องสามารถดูแบบรูปได้

- 1) ดูประเภทของแบบรูปความต้องการ
- 2) ดูข้อมูลทั่วไป เช่น วันที่สร้าง วันที่แก้ไข ผู้เขียน
- 3) ดูข้อมูลองค์ประกอบตามชนิดของแบบรูป

7.2.1.2 ต้องสามารถสร้างแบบรูปได้

- 1) สร้างแบบรูปที่ถูกต้องตามรูปแบบที่กำหนด
- 2) มีเครื่องมือตรวจสอบความถูกต้องของแบบรูปที่สร้าง
- 3) มีเครื่องมือช่วยในการสร้างแบบรูปแบบทีละขั้นตอน

7.2.1.3 ต้องสามารถใช้ประเมินและแก้ไขแบบรูปได้

- 1) ใช้แก้ไขความถูกต้องของแบบรูป
- 2) ใช้ประเมินความถูกต้องของแบบรูป
- 3) ใช้เพิ่มเติมความสมบูรณ์ของแบบรูป

7.2.1.4 ต้องสามารถค้นหาแบบรูปได้

- 1) ค้นหาตามประเภทของแบบรูป
- 2) ค้นหาจากคำสำคัญที่เกี่ยวข้อง
- 3) ค้นหาตามปัญหาหรือลักษณะแวดล้อมของปัญหา

7.2.1.5 ต้องสามารถแสดงเป็นเอกสารสำหรับการอ่านหรือพิมพ์ได้

- 1) แสดงในรูปแบบที่ง่ายต่อการอ่าน
- 2) พิมพ์แบบรูปเพื่อใช้ในการอ่าน

7.2.2 ความต้องการที่ไม่ใช่หน้าที่

7.2.2.1 ความง่ายในการใช้งาน

ผู้ใช้งานไม่จำเป็นต้องเข้าใจโครงสร้างแบบรูปหรือรูปแบบในการเก็บข้อมูลอย่างลึกซึ้งก็สามารถใช้เครื่องมือได้ โดยเครื่องมือต้องมีความง่ายในการใช้งาน

7.2.2.2 ความสามารถด้านข้อมูล

รูปแบบในการจัดเก็บข้อมูลต้องเหมาะสมและสามารถขยายผลของการใช้ข้อมูลได้

7.2.2.3 รายละเอียดของการพัฒนา

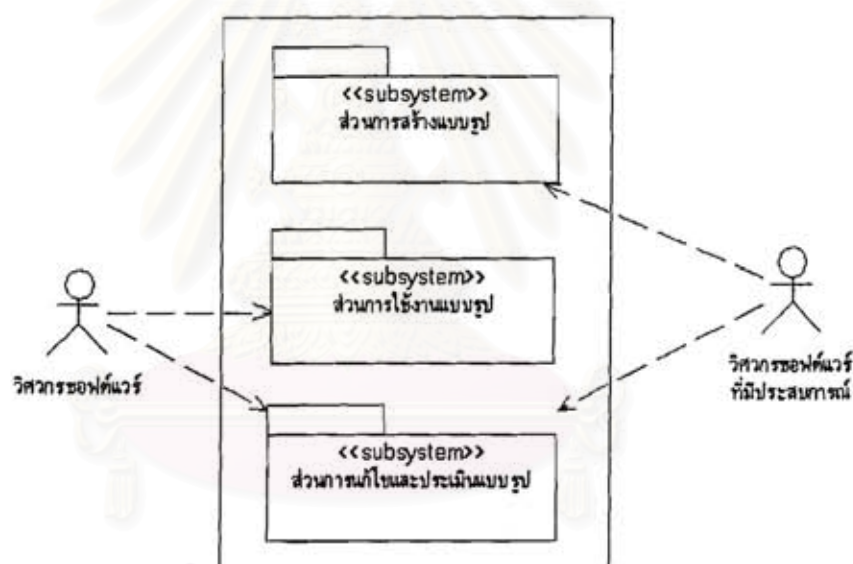
ต้องมีรายละเอียดเกี่ยวกับการทำงาน การออกแบบ และ โครงสร้างของเครื่องมือ

7.2.2.4 การติดตั้งเครื่องมือ

เครื่องมือและข้อมูลแบบรูปต้องง่ายต่อการติดตั้งและนำไปใช้งาน

7.3 การวิเคราะห์และออกแบบเครื่องมือ

7.3.1 หน้าที่ความสามารถของเครื่องมือ

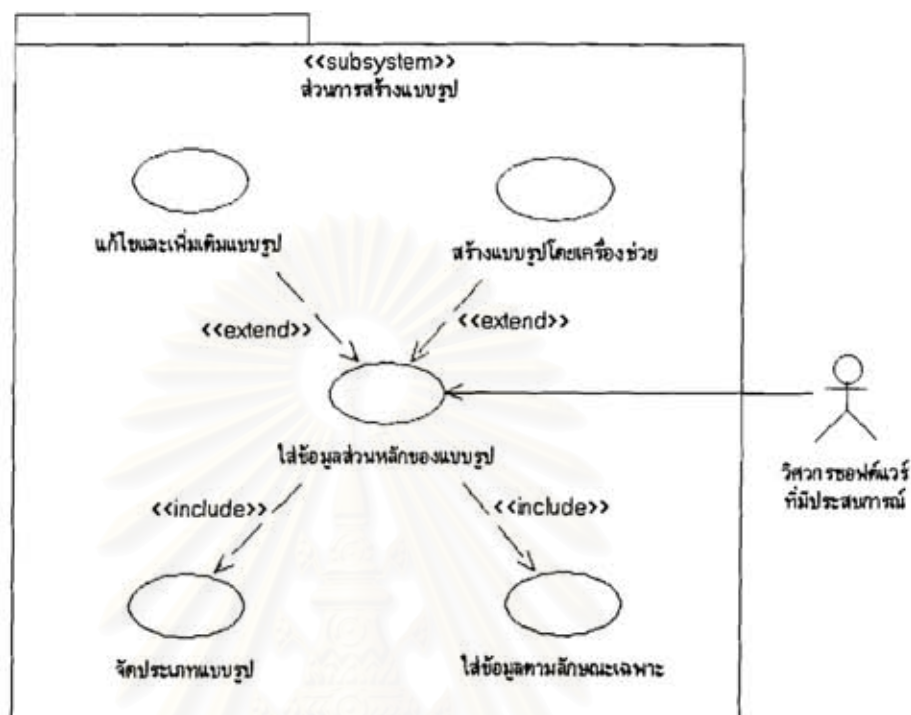


รูปที่ 7.1 ส่วนต่างๆของเครื่องมือที่ช่วยในการสร้างและใช้งานแบบรูป

หน้าที่ความสามารถของเครื่องมือสามารถแบ่งตามลักษณะการใช้งานได้ 3 ส่วน ดังรูปที่ 7.1 โดยมีรายละเอียดของหน้าที่การทำงาน ดังนี้

7.3.1.1 ส่วนการร่างแบบรูป

เป็นส่วนของเครื่องมือที่ช่วยในการสร้างแบบรูปความต้องการ เพื่อช่วยให้การสร้างแบบรูปทำได้ง่าย มีความถูกต้องและสมบูรณ์ และสามารถตรวจสอบได้ โดยสามารถแสดงการทำงานได้ดังแผนภาพยูสเคสในรูปที่ 7.2



รูปที่ 7.2 หน้าทีการทำงานของเครื่องมือในส่วนการสร้างแบบรูป

ในรูปที่ 7.2 แสดงแผนภาพยูสเคสการทำงานในส่วนการสร้างแบบรูปความต้องการ ซึ่งมีรายละเอียดของการทำงานในส่วนต่างๆ ดังนี้

1) ใส่ข้อมูลส่วนหลักของแบบรูป

ใช้ใส่ข้อมูลที่สำคัญของแบบรูปที่เป็นลักษณะทั่วไปของแบบรูปความต้องการ ซึ่งรวมถึงการกำหนดประเภทของแบบรูป โดยข้อมูลเหล่านี้จะเป็นข้อมูลในองค์ประกอบหลักของแบบรูปความต้องการ

2) ใส่ข้อมูลตามลักษณะเฉพาะ

ใช้ใส่ข้อมูลในส่วนที่ขึ้นอยู่กับประเภทของแบบรูปความต้องการซึ่งเป็นลักษณะเฉพาะที่เพิ่มเข้ามา เช่น ลักษณะเฉพาะในแบบรูปโดเมนความต้องการ

3) จัดประเภทแบบรูป

ใช้ในการจัดกลุ่มหรือกำหนดประเภทให้แบบรูปที่สร้าง รวมทั้งนำไปใช้ช่วยในการค้นหาประเภทที่ใกล้เคียงเพื่อนำมากำหนดให้แบบรูป

4) ตรวจสอบองค์ประกอบและความสมบูรณ์ของแบบรูป

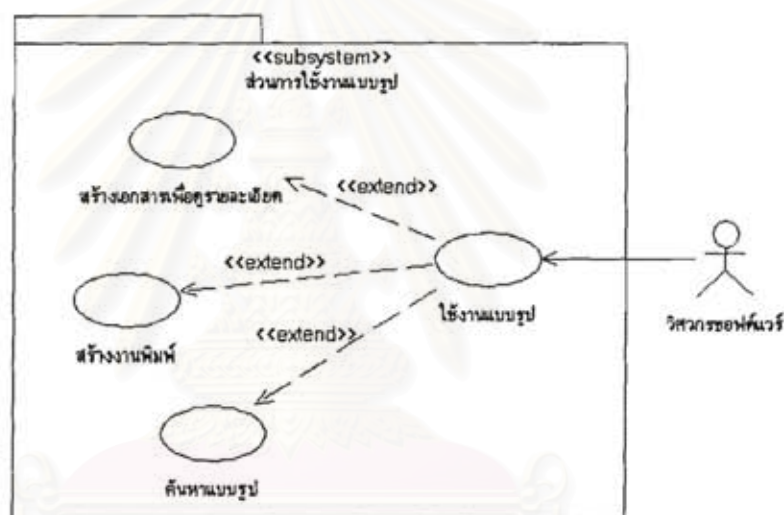
ใช้ในการตรวจสอบการกรอกข้อมูลในองค์ประกอบของแบบรูป โดยการตรวจสอบความถูกต้องและความสมบูรณ์ให้เป็นไปตามรูปแบบและโครงสร้างที่กำหนด

5) สร้างแบบรูปโดยเครื่องช่วย

ใช้ในการสร้างแบบรูปความต้องการ โดยมีเครื่องมือช่วย ซึ่งจะเป็นการสร้างแบบรูปตามลำดับขั้นตอน เมื่อดำเนินการตามจนจบจะทำให้ได้แบบรูปความต้องการที่สมบูรณ์

7.3.1.2 ส่วนการใช้งานแบบรูป

เป็นส่วนที่นำมาช่วยในการใช้งานแบบรูป เพื่อให้การนำแบบรูปไปใช้งานทำได้ง่ายขึ้น ทั้งในด้านการค้นหาแบบรูป ดูรายละเอียด หรือแสดงแบบรูปในรูปแบบที่ง่ายต่อการนำไปใช้ ดังแสดงในแผนภาพยูสเคส ในรูปที่ 7.3



รูปที่ 7.3 หน้าการทำงานของเครื่องมือในส่วนการใช้งานแบบรูป

รูปที่ 7.3 แสดงแผนภาพยูสเคสการทำงานในส่วนการใช้งานแบบรูปความต้องการ ซึ่งมีรายละเอียดของการทำงานในส่วนต่างๆ ดังนี้

1) ใช้งานแบบรูป

ใช้ในการแสดงแบบรูป ตลอดจนช่วยในการอธิบายองค์ประกอบและข้อมูล

2) ค้นหาแบบรูป

ใช้ในการค้นหาแบบรูปตามลักษณะของปัญหาหรือสภาวะแวดล้อม ตามประเภทของแบบรูปที่ได้จัดประเภทไว้ หรือค้นหาจากคำสำคัญที่มีความเกี่ยวข้องกัน

3) สร้างเอกสารเพื่อดูรายละเอียด

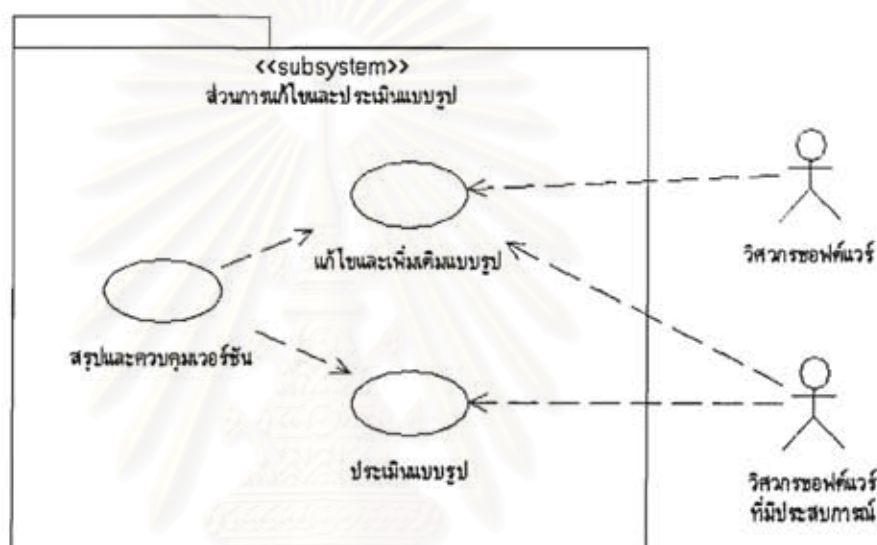
ใช้ในการแสดงแบบรูปในลักษณะของเอกสารที่ง่ายต่อการนำไปใช้

4) สร้างงานพิมพ์

ใช้ในการพิมพ์แบบรูปออกจากเครื่องพิมพ์

7.3.1.3 ส่วนการแก้ไขและประเมินแบบรูป

เป็นส่วนที่จะนำมาช่วยในการประเมินความสามารถในการนำไปใช้งานของแบบรูปนั้นๆ ซึ่งใช้ในการบันทึกผลการตรวจสอบและผลของการใช้แบบรูป สามารถแก้ไขแบบรูปเพื่อให้ชัดเจนหรือถูกต้องมากขึ้น รวมทั้งสามารถรายละเอียดของการตรวจสอบ ความคุมผลและประมาณการตรวจสอบ ดังแผนภาพยูสเคส ในรูปที่ 7.4



รูปที่ 7.4 หน้าที่การใช้งานของเครื่องมือในส่วนการประเมินแบบรูป

รูปที่ 7.4 แสดงแผนภาพยูสเคสการทำงานในส่วนการประเมินแบบรูปความต้องการ ซึ่งมีรายละเอียดของการทำงานในส่วนต่างๆ ดังนี้

1) แก้ไขเพิ่มเติมแบบรูป

ใช้แก้ไขเพิ่มเติมข้อมูลในแบบรูปเพื่อให้มีความสมบูรณ์ หรือเพื่อให้เกิดความถูกต้อง

2) ประเมินแบบรูป

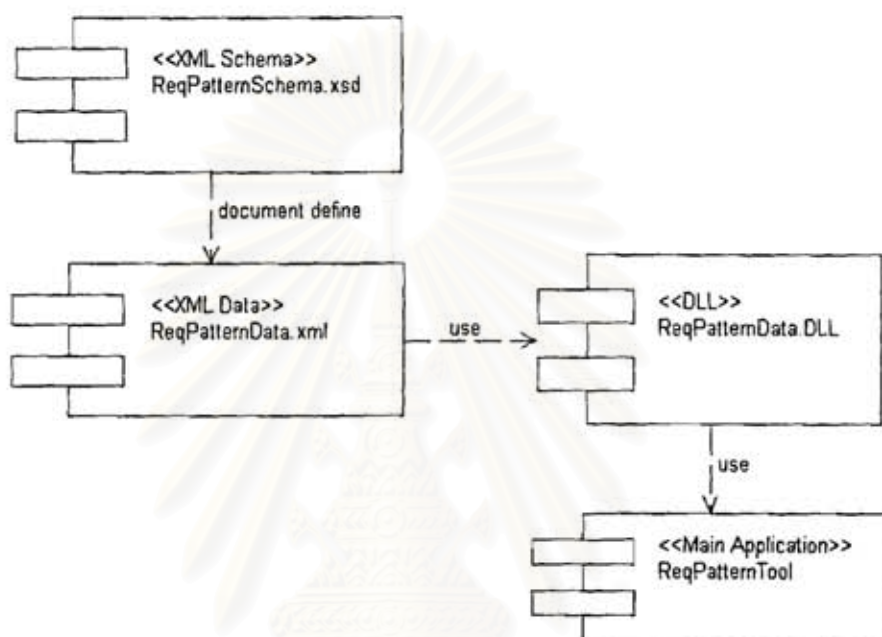
ใช้ประเมินความถูกต้องของแบบรูปว่าแบบรูปสามารถนำไปใช้ได้ถูกต้องหรือไม่ อาจจะเป็นการยอมรับ ปฏิเสธ รวมไปถึงการประเมินว่าแบบรูปไม่สามารถนำมาใช้งานได้แล้ว

3) สร้างและควบคุมเวอร์ชัน

เป็นการสร้างการแก้ไขเปลี่ยนแปลงในแบบรูปเพื่อบันทึกเป็นข้อมูลการแก้ไขและผลในการประเมิน

7.3.2 สถาปัตยกรรมของระบบ

เครื่องมือที่พัฒนาขึ้นจะทำงานแบบสแตนด์อโลน (Standalone) โดยมีส่วนประกอบตามแผนภาพคอมโพเนนต์ (Component Diagram) ดังรูปที่ 7.5



รูปที่ 7.5 แผนภาพคอมโพเนนต์ของเครื่องมือที่พัฒนา

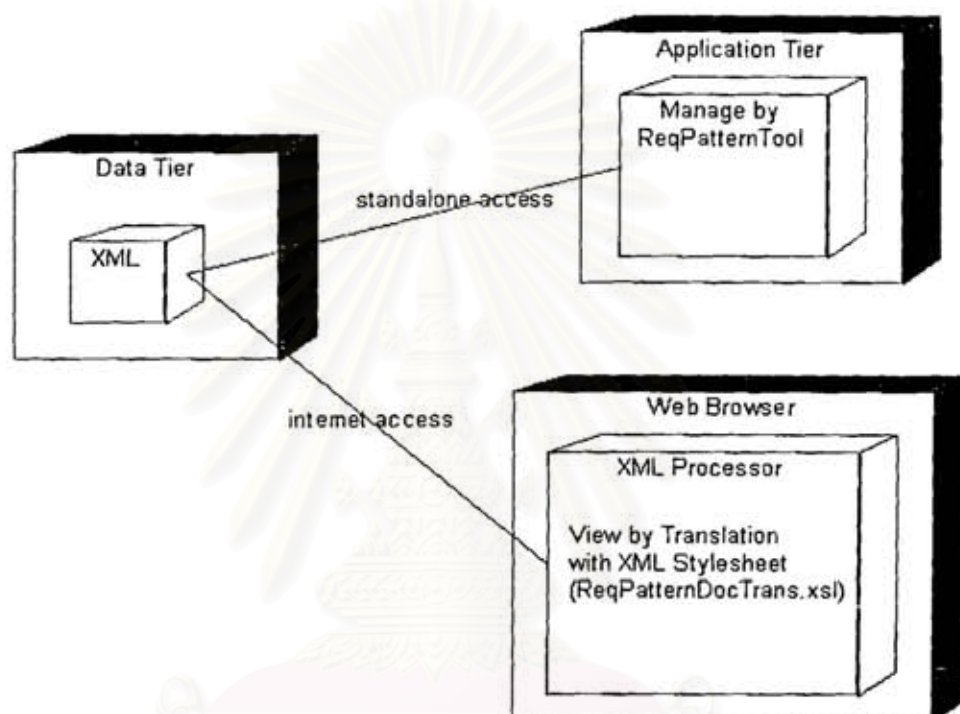
ส่วนประกอบที่อยู่ในรูปที่ 7.5 มีรายละเอียด ดังนี้

- 1) <<XML Schema>> ReqPatternSchema.xsd คือ ส่วนของเอกสารเอ็กซ์เอ็มแอลสกีมาซึ่งใช้เป็นเครื่องมือในการกำหนดและตรวจสอบความถูกต้องของข้อมูลแบบรูปความต้องการ
- 2) <<XML Data>> ReqPatternData.xml คือ ส่วนของเอกสารเอ็กซ์เอ็มแอลที่ใช้เก็บข้อมูลแบบรูปความต้องการ
- 3) <<DLL>> ReqPatternData.DLL คือ ส่วนของคลังโปรแกรมที่ใช้ในการจัดการกับข้อมูลแบบรูปความต้องการในเอกสารเอ็กซ์เอ็มแอล
- 4) <<Main Application>> ReqPatternTool คือ ส่วนหลักของโปรแกรมประยุกต์ที่ใช้งาน

เพื่อให้เกิดความยืดหยุ่นในการทำงานของเครื่องมือ จึงได้ทำการแยกส่วนของข้อมูลและส่วนของการทำงานออกจากกัน และสามารถทำการเข้าถึงข้อมูลได้หลายช่องทาง ซึ่งในวิทยานิพนธ์มีการเข้าถึงส่วนข้อมูลโดยตรงผ่านทางโปรแกรมที่พัฒนาและเข้าถึงข้อมูลในรูปแบบเอ็กซ์เอ็มแอลผ่านทางเครือข่ายอินเทอร์เน็ต โครงสร้างและข้อมูลที่เก็บในรูปแบบเอ็กซ์เอ็มแอลนั้นสามารถรายละเอียดได้ในภาคผนวก

ข. ซึ่งจะแสดงเอ็กซ์เอ็มแอลสก็มาของ โครงสร้างแบบรูปความต้องการ และตัวอย่างเอ็กซ์เอ็มแอลที่เก็บข้อมูลแบบรูปความต้องการ

โครงสร้างของเครื่องมือสามารถแสดงส่วนประกอบได้ดังรูปที่ 7.6 โดยแยกออกเป็นสามข้อมูล (Data Tier) ส่วนโปรแกรมประยุกต์ (Application Tier) และส่วนเว็บเบราว์เซอร์ (Web Browser)

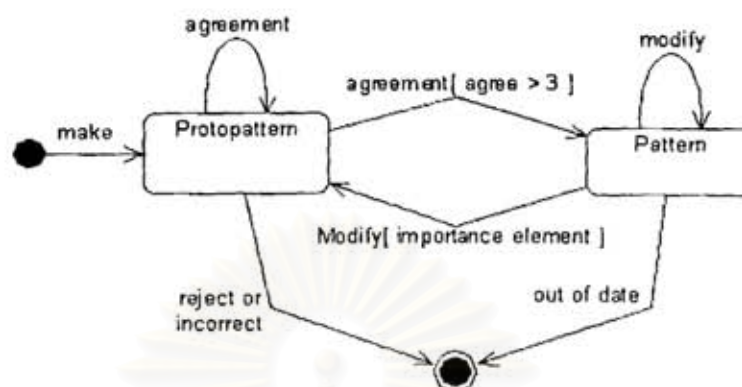


รูปที่ 7.6 ส่วนประกอบของเครื่องมือ

- 1) ส่วนข้อมูล เป็นส่วนที่เก็บเอกสารเอ็กซ์เอ็มแอลที่เป็นข้อมูลของแบบรูปในระบบ
- 2) ส่วนโปรแกรมประยุกต์ เป็นส่วนที่ทำงานแบบสแตนด์อโลนผ่านทางเครื่องมือที่สร้างขึ้น
- 3) ส่วนเว็บเบราว์เซอร์ เป็นส่วนที่เข้าถึงข้อมูลเอ็กซ์เอ็มแอลโดยตรง ซึ่งในเบื้องต้นเป็นการแสดงข้อมูลแบบรูปโดยอาศัยการแปลงข้อมูลด้วยเอ็กซ์เอ็มแอลสไคล์ชีทเพื่อให้สามารถอ่านข้อมูลแบบรูปได้เข้าใจมากขึ้น

7.3.3 โครงสร้างและการทำงานของเครื่องมือ

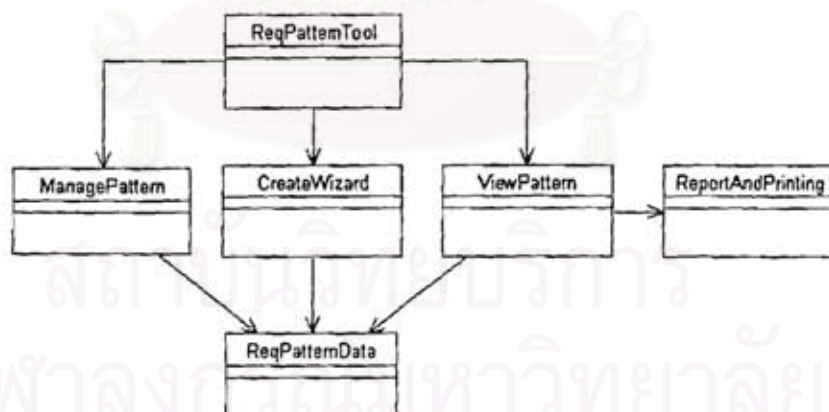
เนื่องจากเครื่องมือที่พัฒนาเป็นเครื่องมือที่ใช้ในการจัดการกับข้อมูล ดังนั้นจึงได้ทำการพิจารณาการทำงานเกี่ยวกับคุณลักษณะข้อมูลแบบรูป โดยแสดงเป็นแผนภาพสเตทชาร์ตดังรูปที่ 7.7 ซึ่งแผนภาพนี้ได้อธิบายวงจรชีวิตของข้อมูลแบบรูป ดังนั้นเครื่องมือที่พัฒนาขึ้นจะต้องครอบคลุมและรองรับทุกช่วงวงจรชีวิตของแบบรูป



รูปที่ 7.7 แผนภาพสเตตชาร์ตแสดงวงจรชีวิตข้อมูลแบบรูป

รูปที่ 7.7 อธิบายลักษณะของข้อมูลแบบรูป ซึ่งมีด้วยกัน 2 ลักษณะ คือ ในสถานะแบบรูปเบื้องต้น (Protopattern) และในสถานะแบบรูปที่สมบูรณ์แล้ว (Pattern) ซึ่งสถานะทั้ง 2 สามารถที่จะเปลี่ยนแปลงตามการแก้ไขหรือการแสดงความเห็น โดยข้อมูลแบบรูปเริ่มต้นขึ้นที่สถานะแบบรูปเบื้องต้นเมื่อทำการสร้าง และเมื่อมีความเห็นด้วยกับเนื้อหาในแบบรูปเบื้องต้นมากกว่า 3 ครั้ง ก็จะถือว่าสถานะของแบบรูปนั้นเป็นแบบรูปที่สมบูรณ์แล้ว

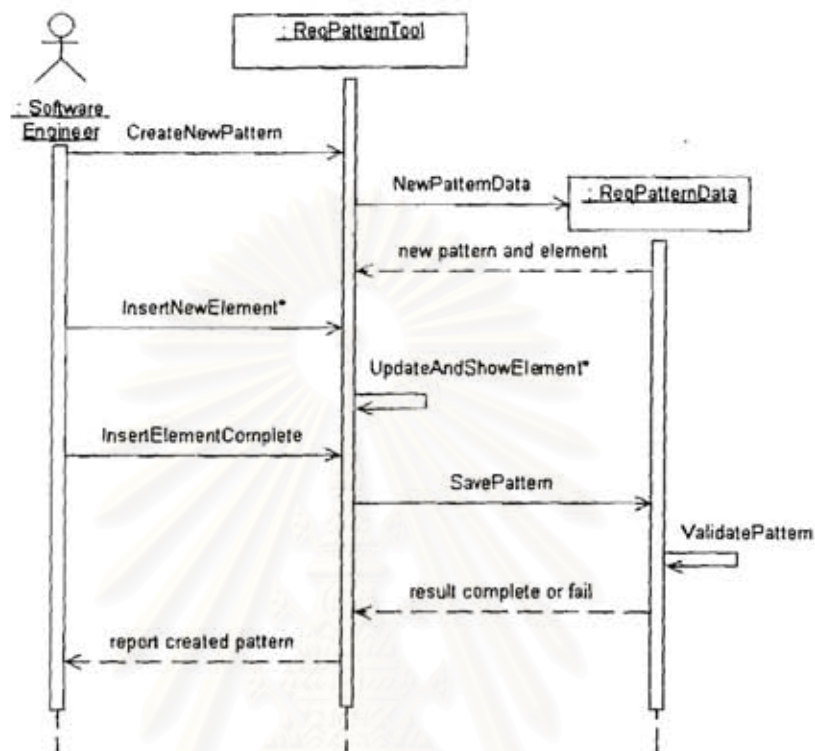
การทำงานของเครื่องมือจะเป็นไปตามการจัดการข้อมูล เมื่อทำการออกแบบเครื่องมือ คลาสหลักที่ทำงานในระบบจะประกอบไปด้วยคลาสต่างๆ ดังที่แสดงในแผนภาพคลาสในรูปที่ 7.8



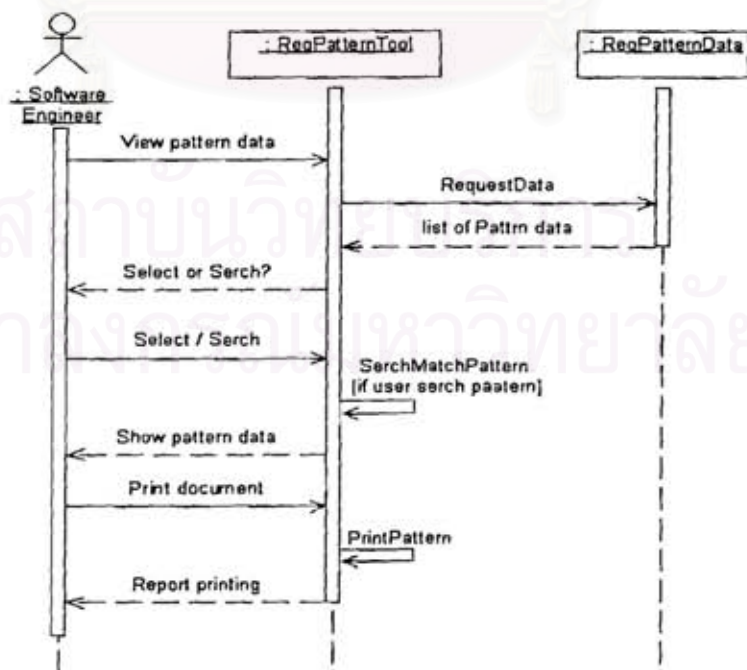
รูปที่ 7.8 แผนภาพคลาสของเครื่องมือที่พัฒนา

ลักษณะการทำงานของเครื่องมือสามารถสรุปได้เป็น 4 ส่วน ดังนี้

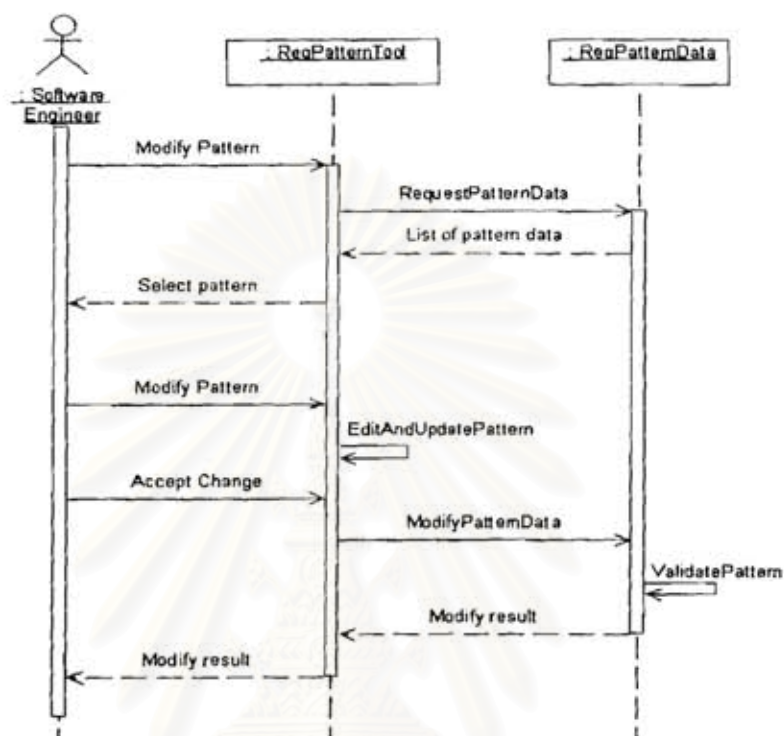
- 1) การสร้างแบบรูปใหม่ ซึ่งมีการทำงานดังรูปที่ 7.9
- 2) การดูหรือใช้งานแบบรูป ซึ่งมีการทำงานดังรูปที่ 7.10
- 3) การแก้ไขแบบรูป ซึ่งมีการทำงานดังรูปที่ 7.11
- 4) การประเมินแบบรูป ซึ่งมีการทำงานดังรูปที่ 7.12



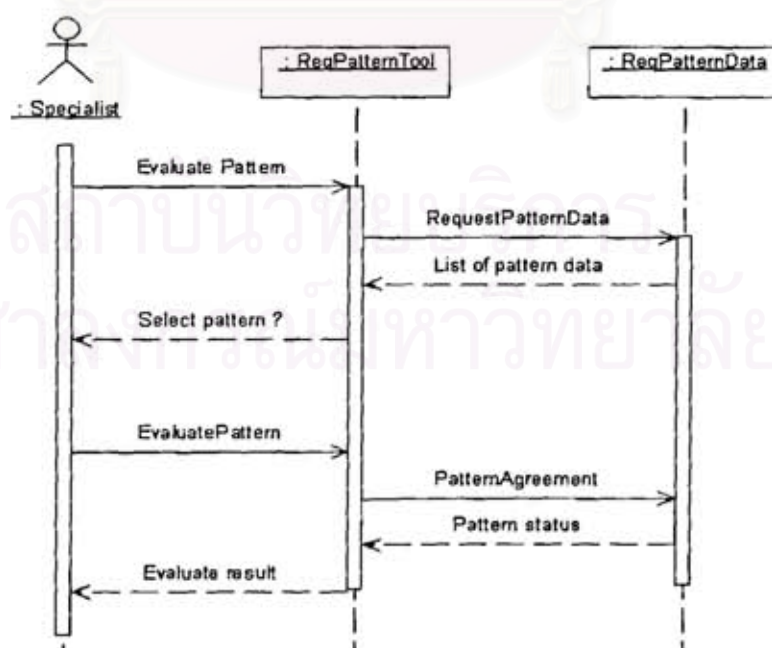
รูปที่ 7.9 แผนภาพลำดับงานในการสร้างแบบรูปขึ้นมาใหม่



รูปที่ 7.10 แผนภาพลำดับงานในการดูหรือใช้งานแบบรูป



รูปที่ 7.11 แผนภาพลำดับงานในการแก้ไขข้อมูลในแบบรูป



รูปที่ 7.12 แผนภาพลำดับงานในการประเมินแบบรูป

7.4 การทดสอบเครื่องมือ

การทดสอบเครื่องมือจะทำใน 2 ส่วนด้วยกัน คือ ทดสอบความถูกต้องของการทำงานในการเข้าถึงข้อมูลแบบรูปในการดู สร้าง แก้ไขและประเมิน และทดสอบการทำงานผ่านส่วนต่อประสานผู้ใช้

7.4.1 ทดสอบความถูกต้องในการทำงาน

การทดสอบความถูกต้องของเครื่องมือเป็นการทดสอบหน้าที่ว่าสามารถทำงานได้ถูกต้องตามกำหนดหรือไม่ ซึ่งจะทดสอบขั้นต้นตามความต้องการหน้าที่ที่ระบุ จากนั้นจึงจะทดสอบโดยรวมด้วยการเปรียบเทียบเนื้อหาของข้อมูลแบบรูป

การทดสอบโดยรวมจะทำการสร้างข้อมูลแบบรูปในรูปเอกสารอิเล็กทรอนิกส์โดยใช้เครื่องมือแก้ไขเอกสารอิเล็กทรอนิกส์ โดยสร้างเป็นเอกสารอิเล็กทรอนิกส์ที่ถูกกฎหมายของแบบรูป และนำมาเปรียบเทียบเทียบกับเอกสารอิเล็กทรอนิกส์ที่ได้จากการใช้เครื่องมือที่สร้างขึ้น

7.4.2 ทดสอบการใช้งานและส่วนต่อประสานผู้ใช้

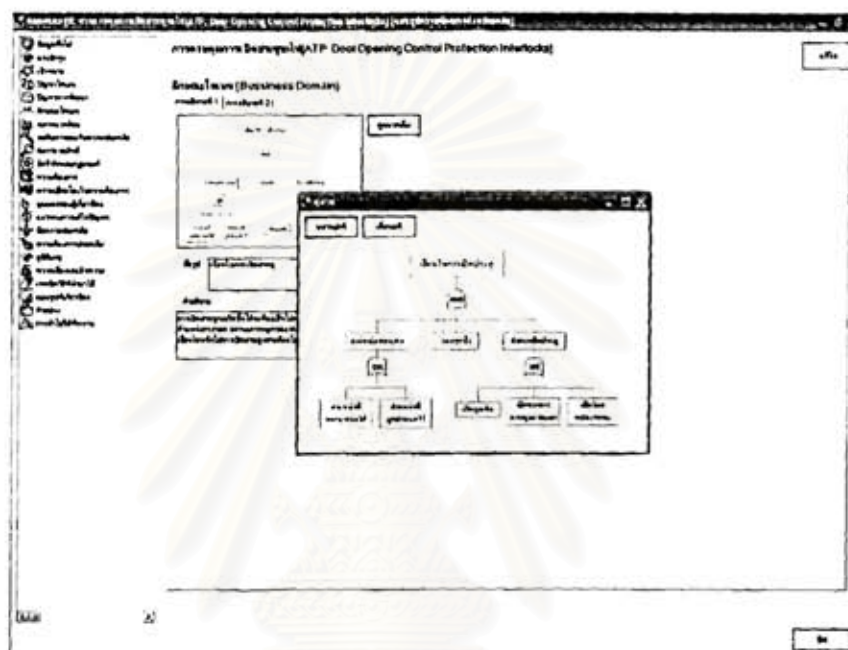
หลังจากทดสอบถูกต้องของเครื่องมือแล้ว จะทำการทดสอบส่วนต่อประสานผู้ใช้โดยการประเมินผลการใช้งาน โดยเครื่องมือจะต้องมีความง่ายในการใช้ การแสดงผลต้องมีความชัดเจน มีรูปแบบที่ดี และมีความสวยงามเหมาะสม

7.5 ตัวอย่างของเครื่องมือที่พัฒนา

ในส่วนนี้จะแสดงตัวอย่างส่วนต่อประสานของเครื่องมือที่พัฒนา เพื่อแสดงรายละเอียดของเครื่องมือ และการจัดการหรือรูปแบบวิธีในการใช้งานเครื่องมือ

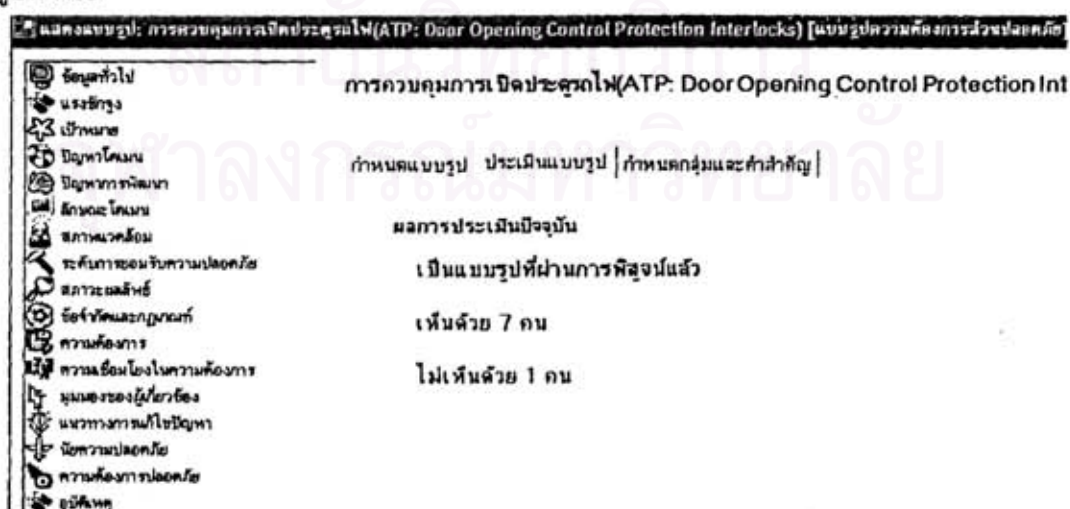
ในรูปที่ 7.13 จะแสดงหน้าจอที่ปรากฏขึ้นเมื่อเริ่มใช้งาน โดยจะแสดงรายการของแบบรูปความต้องการที่จำแนกตามประเภทในแถบด้านซ้าย และเมื่อทำการเลือกแบบรูปความต้องการจะแสดงรายละเอียดข้อมูลพื้นฐานของแบบรูปความต้องการที่เลือกทางด้านขวา และเครื่องมือจะแสดงเมนูพื้นฐานที่จำเป็นในการใช้งาน และมีปุ่มคำสั่งเพื่อดูรายละเอียดแบบรูป แก้ไขแบบรูป และแสดงแบบรูปในลักษณะเอกสารรายงาน

ในการใช้งานเครื่องมือเมื่อเลือกแบบรูปความต้องการที่สนใจแล้วจะเข้าไปดูรายละเอียดหรือทำการแก้ไขข้อมูล โดยจะแสดงหน้าต่างใหม่เพื่ออธิบายรายละเอียดหรือใช้ในการทำการแก้ไขแบบรูป ดังรูปที่ 7.15 ซึ่งเป็นส่วนที่ใช้แสดงลักษณะโดเมนของแบบรูปโดเมนความต้องการ



รูปที่ 7.15 หน้าจอแสดงลักษณะโดเมนของแบบรูปโดเมนความต้องการ

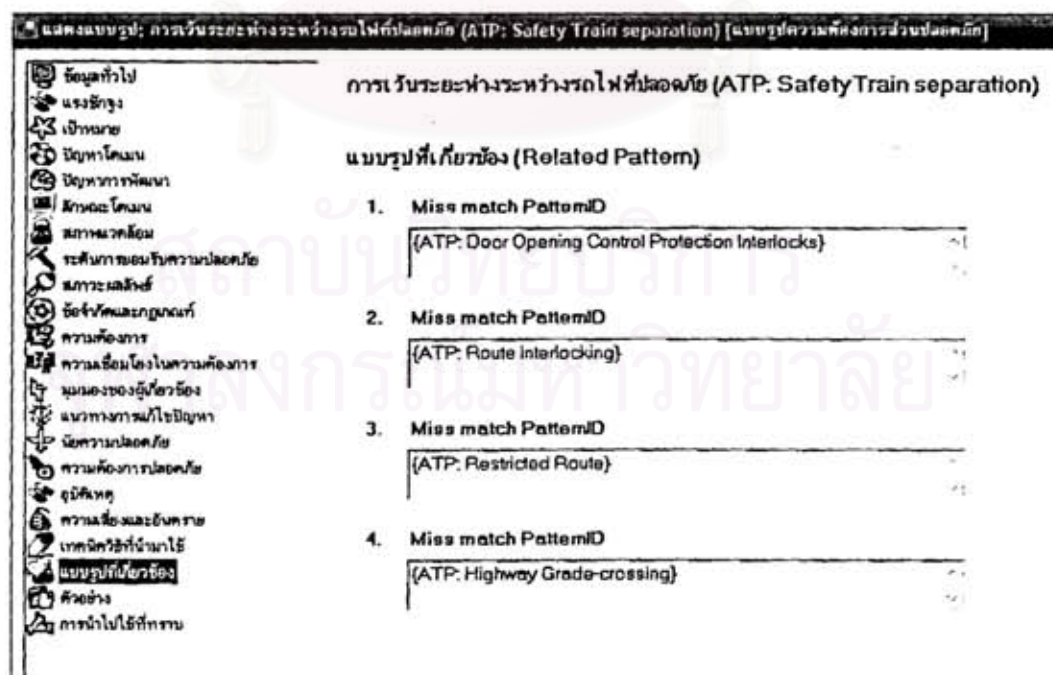
ในส่วนการแสดงผลข้อมูลต่างๆ ของแบบรูปซึ่งมีรายละเอียดและรูปแบบองค์ประกอบแต่ละส่วนแตกต่างกัน การแสดงรายละเอียดของแต่ละองค์ประกอบก็จะมีแตกต่างกันไปด้วย ดังเช่นในรูปที่ 7.16 ถึงรูปที่ 7.18



รูปที่ 7.16 ตัวอย่างการแสดงผลรายละเอียดของแบบรูปโดยเครื่องมือ



รูปที่ 7.17 ตัวอย่างการแสดงรายละเอียดของแบบรูปโดยเครื่องมือ



รูปที่ 7.18 ตัวอย่างการแสดงรายละเอียดของแบบรูปโดยเครื่องมือ

รูปที่ 7.16 ถึง 7.18 เป็นการแสดงตัวอย่างในส่วนของการรายละเอียดส่วนต่างๆของแบบรูปสำหรับการแก้ไขหรือสร้างแบบรูปใหม่จะมีการแสดงผลคล้ายกัน แต่จะสามารถแก้ไขหรือเพิ่มเติมข้อมูลในแบบรูปความต้องการได้ และยังสามารถใช้ในการประเมินและกำหนดสถานะให้แบบรูปได้ โดยมีตัวอย่างหน้าจอแสดงดังรูปที่ 7.19 ถึงรูปที่ 7.21

แสดงแบบรูป: การเว้นระยะห่างระหว่างรถไฟที่ปลอดภัย (ATP: Safety Train separation) [แบบรูปความต้องการส่วนปลอดภัย]

การ เว้นระยะห่างระหว่างรถไฟที่ปลอดภัย (ATP: Safety Train separation)

กำหนดแบบรูป ประเมินแบบรูป | กำหนดกลุ่มและค่าสำคัญ |

ผลการประเมินปัจจุบัน

เป็นแบบรูปที่ผ่านการพิสูจน์แล้ว

เห็นด้วย 3 คน

ไม่เห็นด้วย 0 คน

(ยกเลิกการประเมินทั้งหมด ทั้งที่เห็นด้วยและไม่เห็นด้วย โดยไม่บันทึกเป็นต้นฉบับ ซึ่งจะเป็นการเปลี่ยนสถานะแบบรูปกับสถานะแบบรูปต้นแบบด้วย)

ข้อมูลทั่วไป
แรงรักษา
เป้าหมาย
ปัญหาใดแทน
ปัญหาการพัฒนา
ลักษณะใดแทน
สถานะแวดล้อม
ระดับการยอมรับความปลอดภัย
สถานะผลลัพธ์
ข้อจำกัดและกฎเกณฑ์
ความต้องการ
ความเชื่อมโยงในความต้องการ
มุมมองของผู้เกี่ยวข้อง
แนวทางการแก้ไขปัญหา
นิยามความปลอดภัย
ความต้องการปลอดภัย
อุปสรรค
ความเสี่ยงและอันตราย
เทคนิควิธีที่นำมาใช้

รูปที่ 7.19 ตัวอย่างการแก้ไขหรือสร้างแบบรูปความต้องการ โดยเครื่องมือ

แสดงแบบรูป: การเว้นระยะห่างระหว่างรถไฟที่ปลอดภัย (ATP: Safety Train separation) [แบบรูปความต้องการส่วนปลอดภัย]

การ เว้นระยะห่างระหว่างรถไฟที่ปลอดภัย (ATP: Safety Train separation)

ความต้องการ (Requirements)

ความต้องการที่ 1 | ความต้องการที่ 2 | ความต้องการที่ 3 | ความต้องการที่ 4 | ความต้องการที่ 5 | ความต้องการที่ 6 | ความต้องการที่ 7

หมายเลขความต้องการ F6

ประเภทของความต้องการ

☐ ความต้องการเชิงหน้าที่

☒ ความต้องการที่ไม่ใช่หน้าที่

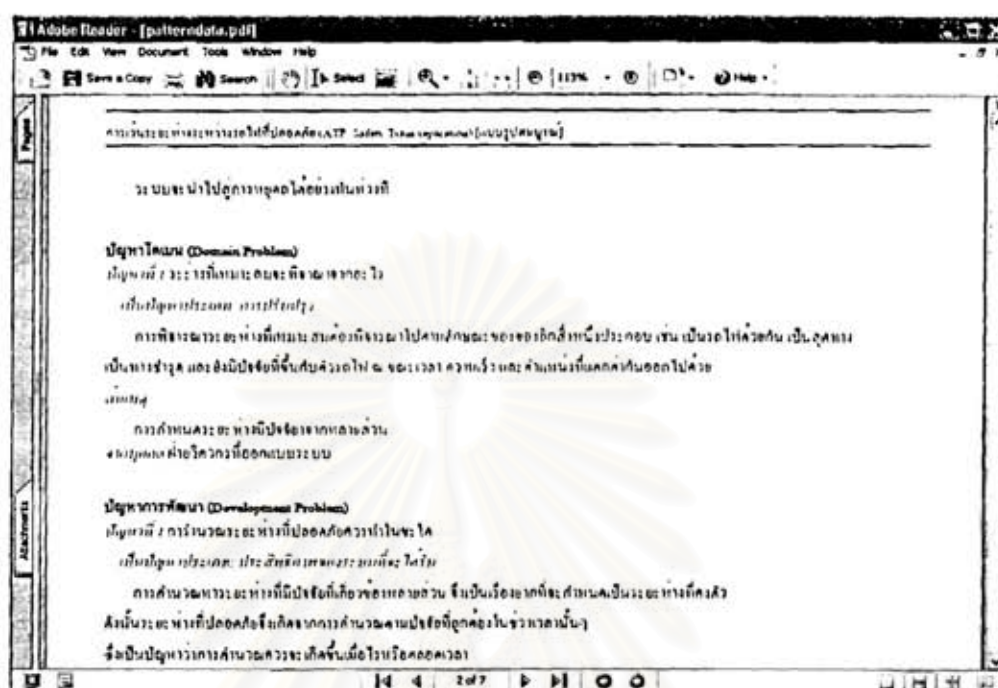
ประเภทของความต้องการที่ไม่ใช่หน้าที่

ความต้องการเชิงคุณสมบัติ

ระบุความต้องการ

จาก

รูปที่ 7.20 ตัวอย่างการแก้ไขหรือสร้างแบบรูปความต้องการ โดยเครื่องมือ



รูปที่ 7.23 ตัวอย่างเอกสารแบบรูปความต้องการที่เครื่องมือสร้างขึ้น

สถาบันวิทยบริการ
 จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 8

การทดสอบโครงสร้างแบบรูปความต้องการ

ในบทนี้จะเป็นการทดสอบโครงสร้างแบบรูปที่ได้ทำการออกแบบ โดยการสร้างชุดแบบรูปมาทดลองใช้งานเปรียบเทียบโดยอาศัยการตอบคำถามเพื่อวัดระดับความสามารถของการทำงานในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ และทำการประเมินผลจากผู้ที่ได้ลองใช้แบบรูปความต้องการ สำหรับชุดแบบรูปและคำถามที่ใช้ในการทดสอบสามารถรายละเอียดได้ในภาคผนวก ก

8.1 จุดประสงค์ในการทดสอบ

เนื่องจากวิทยานิพนธ์ชิ้นนี้เป็นการพิจารณาโครงสร้างที่นำมาใช้กับแบบรูปความต้องการ การทดสอบจึงให้ความสำคัญกับการประเมินโครงสร้างแบบรูปความต้องการเป็นหลัก และให้ความสำคัญกับตัวอย่างแบบรูปความต้องการที่สร้างขึ้นเพื่อใช้ทดสอบรองลงไป โดยมีจุดประสงค์ดังนี้

- 1) เพื่อศึกษาผลของการใช้แบบรูปความต้องการต่องานวิศวกรรมความต้องการซอฟต์แวร์
- 2) เพื่อศึกษาผลของการใช้แบบรูปความต้องการต่อความต้องการของระบบปลอดภัยเชิงวิกฤติ
- 3) เพื่อประเมินความเหมาะสมของโครงสร้างแบบรูปความต้องการ
- 4) เพื่อพิจารณาผลของการทำงานที่มีการใช้แบบรูปความต้องการร่วมด้วย

8.2 องค์ประกอบในการทดลอง

8.2.1 หน่วยทดลอง

ผู้ทดลองจะต้องมีความรู้พื้นฐานสำคัญ 2 ด้าน คือ ด้านวิศวกรรมความต้องการ และด้านระบบปลอดภัยเชิงวิกฤติ จึงทำการคัดเลือกผู้ทดลองจากนักศึกษาระดับบัณฑิตศึกษาที่ผ่านการเรียนวิชาวิศวกรรมความต้องการซอฟต์แวร์ ของภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่มีผลการเรียนอยู่ในช่วง B ถึง A เท่านั้น และผู้ทดลองทุกคนจะได้รับการอบรมเกี่ยวกับการพัฒนาซอฟต์แวร์สำหรับระบบปลอดภัยเชิงวิกฤติ และได้รับรายละเอียดของระบบเดินรถอัตโนมัติซึ่งเป็นระบบที่นำมาใช้ประกอบการทดลอง

ในการทดลองจะใช้หน่วยทดลอง 2 กลุ่ม แต่ละกลุ่มประกอบด้วย 9 หน่วยทดลอง รวมเป็น 18 หน่วยทดลอง

8.2.2 ตัวอย่างประกอบการทดลอง

ตัวอย่างที่ใช้ในการทดลองจะได้มาจากการศึกษากระบวนการเดินรถอัตโนมัติ ซึ่งข้อมูลที่น่ามาใช้เป็นตัวอย่างได้มาจากการศึกษามาตรฐาน [30] และการเข้าเยี่ยมชมการทำงานจริงของระบบโดยจะเลือกใช้ส่วนของระบบมาสร้างเป็นแบบทดสอบ 3 ชุด โดยแต่ละชุดจะมีชุดคำถามและแบบรูปที่สอดคล้องกัน แบบรูปที่น่ามาใช้ในการทดลองคือแบบรูปที่สร้างขึ้นในบทที่ 5

8.2.3 การดำเนินการและการควบคุมการทดลอง

ก่อนการทดลองเริ่มต้นจะมีการบรรยายอบรมผู้ทดลองเพื่อให้ผู้ทดลองมีความเข้าใจพื้นฐานในเรื่องที่จำเป็น การทดลองจะเริ่มขึ้นโดยการให้ผู้ทดลองตอบคำถามภายในช่วงระยะเวลาที่กำหนด ทั้งนี้ผู้ทดลองสามารถนำกลับไปพิจารณาได้

ผู้ทดลองจะทำการตอบคำถามในชุดคำถามเกี่ยวกับความต้องการของระบบเดินรถอัตโนมัติ โดยกลุ่มที่ 1 ได้รับแบบทดสอบแต่ไม่ได้รับแบบรูปความต้องการ และกลุ่มที่ 2 ได้รับแบบทดสอบที่มีแบบรูปความต้องการประกอบ โดยผู้ทดลองมีเวลาในการตอบคำถาม 2 วัน โดยกลุ่มที่ใช้แบบรูปความต้องการจะประเมินความถูกต้องของแบบรูป และประเมิน โครงสร้าง การใช้งาน และความพึงพอใจในการใช้งานแบบรูปความต้องการด้วย

8.2.4 การประเมินผลการทดลอง

การประเมินผลที่ได้จากการทดลองจะเป็นการประเมินจากการตอบคำถาม ซึ่งในแต่ละส่วนของคำถามมีจุดประสงค์และเป้าหมายที่จะใช้ประเมินผลแตกต่างกัน โดยการประเมินผลแบ่งออกเป็น 2 ส่วน คือ

- 1) ประเมินเปรียบเทียบผลในการนำแบบรูปไปใช้
- 2) ประเมินโครงสร้าง การใช้งาน และความพึงพอใจในการใช้งานแบบรูปความต้องการ

8.3 ขั้นตอนวิธีในการทดสอบ

8.3.1 ศึกษาและทำความเข้าใจระบบเดินรถอัตโนมัติและแบบรูปความต้องการ

หน่วยทดลองจะได้ทำความเข้าใจเกี่ยวกับระบบเดินรถอัตโนมัติในภาพรวมผ่านการบรรยายและการแนะนำเบื้องต้นจากผู้วิจัย สำหรับหน่วยทดลองที่ใช้แบบรูปความต้องการประกอบการตอบคำถามจะทำการศึกษาแบบรูปความต้องการผ่านตัวอย่างและการอธิบายอย่างสั้นๆ

8.3.2 ตอบคำถามโดยไม่ใช้แบบรูป

หน่วยทดลองจะเริ่มทำการทดลองโดยการตอบคำถามเกี่ยวกับความต้องการซอฟต์แวร์ในระบบเดินรถอัตโนมัติ โดยที่ไม่ได้รับแบบรูปความต้องการ โดยจะแบ่งผู้ทดลองเป็น 3 กลุ่ม กลุ่มละ 3 คน โดยแต่ละกลุ่มจะได้รับคำถามต่างกัน

8.3.3 ตอบคำถามโดยใช้แบบรูป

หน่วยทดลองจะได้รับชุดทดลองที่เป็นคำถามเกี่ยวกับความต้องการซอฟต์แวร์ในระบบเดินรถอัตโนมัติ และแบบรูปที่จะนำมาใช้ประกอบ ซึ่งจะแบ่งเป็น 3 กลุ่ม กลุ่มละ 3 คน เช่นเดียวกันกับการทดลองโดยไม่มีแบบรูป

ในการทดลองส่วนนี้จะต้องประเมินความถูกต้องของแบบรูปด้วย

8.3.4 ตอบคำถามเพื่อประเมินแบบรูปโดยรวม

หน่วยทดลองจะตอบคำถามเพื่อประเมินแบบรูปความต้องการ ในภาพรวม โดยหน่วยทดลองที่ทดลองในส่วนนี้จะใช้หน่วยทดลองที่ผ่านการตอบคำถามโดยใช้แบบรูปใน 8.3.4

8.4 ผลที่ได้จากการทดลอง

ผลการทดลองที่ได้จากการใช้และไม่ใช้แบบรูป สามารถสรุปได้ดังตารางที่ 8.1 ซึ่งเป็นผลคะแนนที่ได้จากการเปรียบเทียบผลการทดลองที่มีต่อชุดคำถามเดียวกัน โดยแสดงผลคะแนนแต่ละตอนและคะแนนรวมในการตอบคำถาม แสดงความเห็นที่มีต่อแบบรูปที่ใช้เป็นคะแนนความถูกต้องของตัวแบบรูปที่นำมาใช้ และแสดงผลต่างเมื่อเปรียบเทียบคะแนนของการตอบคำถามที่มีและไม่มีแบบรูปประกอบ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 8.1 ผลการทดลองที่ได้จากการใช้และไม่ใช้แบบรูป

ชุดคำถามที่	ไม่มีการใช้แบบรูป (1)			มีการใช้แบบรูป (2)				คะแนนที่ต่างกัน (2) – (1)
	ตอน 1 (10)	ตอน 2 (10)	รวม (20)	ตอน 1 (10)	ตอน 2 (10)	รวม (20)	ความถูกต้อง (25)	
1	3	6	9	7	6	13	18	4
1	6	4	10	7	9	16	18	6
1	3	8	11	6	7	13	21	2
เฉลี่ยชุดที่ 1	4.00	6.00	10.00	6.67	7.33	14.00	19.00	4.00
2	3	3	6	6	8	14	17	8
2	5	4	9	9	7	16	20	7
2	3	4	7	3	10	13	22	6
เฉลี่ยชุดที่ 2	3.67	3.67	7.33	6.00	8.33	14.33	19.67	7.00
3	2	7	9	7	6	13	18	4
3	3	6	9	4	7	11	18	2
3	2	6	8	6	6	12	15	4
เฉลี่ยชุดที่ 3	2.33	6.33	8.67	5.67	6.33	12.00	17.00	3.33
ค่าเฉลี่ยรวม	3.33	5.33	8.67	6.11	7.33	13.44	18.56	4.78
ส่วนเบี่ยงเบนมาตรฐาน								2.1082

ตาราง 8.1 แสดงผลลัพธ์ของการทดลองแบ่งตามชุดคำถาม โดยแสดงผลคะแนนจากการตอบคำถามในลักษณะเปรียบเทียบระหว่างการทดลองที่ใช้และไม่ใช้แบบรูปความต้องการ โดยเปรียบเทียบกับผู้ทดลองในระดับเดียวกันที่มีการตอบคำถามในชุดคำถามเดียวกัน การตอบคำถามที่มีการใช้แบบรูปจะแสดงผลการประเมินความถูกต้องของแบบรูปความต้องการประกอบด้วย

การประเมินความถูกต้องของแบบรูปความต้องการทำเพื่อยืนยันผลของการใช้แบบรูปความต้องการ ถ้าแบบรูปความต้องการที่นำมาใช้ไม่มีความถูกต้องเพียงพอ ผลลัพธ์ที่ได้จากการใช้และไม่ใช้แบบรูปความต้องการย่อมไม่แตกต่างกัน ดังนั้นผลการทดลองที่จะนำมาใช้วิเคราะห์จึงจะต้องผ่านการประเมินความถูกต้องโดยได้คะแนนในระดับดี คือ 15 คะแนนจากคะแนนเต็ม 25 เป็นอย่างน้อย

ผลการทดลองในตาราง 8.1 จะได้ค่าเฉลี่ยของผลต่างคะแนนทดลองที่ใช้และไม่ใช้แบบรูปเป็น 4.78 และได้ค่าส่วนเบี่ยงเบนมาตรฐานเป็น 2.1082

ผลที่ได้จากการประเมินโครงสร้างแบบรูปความต้องการ ซึ่งสามารถแยกได้เป็น 3 ด้าน คือ ด้านผล
ต่อการทำงาน ด้านโครงสร้างและองค์ประกอบ และด้านความรู้สึกของผู้ใช้งาน สามารถสรุปได้ดังตารางที่
8.2 โดยแสดงแยกตามคะแนนของแต่ละข้อและคะแนนสรุปแต่ละด้าน

ตารางที่ 8.2 ผลการประเมินแบบรูปความต้องการ

หน่วย ทดลอง ที่	ผลการประเมินแบบรูปความต้องการ											
	ผลต่อการทำงาน (คะแนนเต็มข้อละ 5)				โครงสร้างและ องค์ประกอบ (คะแนนเต็มข้อละ 5)			ความรู้สึกละเมือใช้งาน (คะแนนเต็มข้อละ 5)			รวม	
	1	2	3	4	5	6	7	8	9	10	เต็ม 50	เปอร์เซ็นต์
1	4	4	5	4	4	3	3	4	4	4	39	78%
2	3	4	4	4	4	4	3	4	4	3	37	74%
3	4	2	4	4	4	4	4	4	4	4	38	76%
4	5	5	4	5	5	4	4	5	4	5	46	92%
5	4	4	2	4	5	4	4	3	4	4	38	76%
6	4	5	4	4	4	3	4	3	4	4	39	78%
7	4	4	3	3	4	4	4	4	4	3	37	74%
8	4	5	4	4	5	4	3	4	3	5	41	82%
9	4	5	4	4	4	5	4	4	4	3	41	82%
ค่าเฉลี่ย	4.00	4.22	3.78	4.00	4.33	3.89	3.67	3.89	3.89	3.89	39.56	79%
	4.00				3.96			3.89				

8.5 วิเคราะห์และสรุปผลการทดสอบ

8.5.1 วิเคราะห์เปรียบเทียบผลลัพธ์ของการใช้และไม่ใช้แบบรูป

เนื่องจากการทดลองเป็นการใช้หน่วยตัวอย่างเดียวกันในการทดลองใช้และไม่ใช้แบบรูปความ
ต้องการสำหรับกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ จะเห็นว่าเป็นการทดสอบผลต่างระหว่าง
ค่าเฉลี่ยแบบจับคู่

1) สมมุติฐาน

μ_1 และ μ_2 เป็นคะแนนในการทดสอบที่ใช้และไม่ใช้แบบรูปตามลำดับ
ตั้งสมมุติฐาน

H_0 : การทดสอบโดยใช้และไม่ใช้แบบรูปให้ผลลัพธ์เท่ากัน

H_1 : การทดสอบโดยใช้แบบรูปให้ผลลัพธ์ดีกว่าการไม่ใช้แบบรูป

หรือ

$$H_0: \mu_1 - \mu_2 = 0 \text{ และ } H_1: \mu_1 - \mu_2 > 0$$

$$\mu_d(d_0) = \mu_1 - \mu_2$$

2) สถิติทดสอบ

ขนาดตัวอย่าง ขนาดเล็ก $n=10$ ($n < 30$) และไม่ทราบค่าแปรปรวนประชากร จึงใช้สถิติ

$$\text{ทดสอบ } t = \frac{\bar{d} - d_0}{S_d / \sqrt{n}}$$

3) การคำนวณค่าทางสถิติ

จากผลการทดลองที่ได้ $\bar{d} = 4.78$, $d_0 = 0$, $S_d = 2.1082$, $\sqrt{n} = \sqrt{10} = 3.16$

$$\therefore t = \frac{4.78 - 0}{(2.1082/3.16)} = 6.8020$$

กำหนดระดับนัยสำคัญที่ $\alpha = 0.05$

ดังนั้น จะปฏิเสธสมมุติฐาน H_0 ถ้า $t > t_{1-\alpha; n-1}$ หรือ $t > t_{0.95; 9}$ หรือ $t > 1.86$

4) สรุปผล

เนื่องจาก $t > t_{1-\alpha; n-1}$ ($6.8020 > 1.86$) จึงอยู่ในเขตปฏิเสธ H_0 ทำให้สรุปได้ว่า ผลการทดสอบโดยใช้แบบรูปให้ผลลัพธ์ดีกว่าการไม่ใช้แบบรูป ที่ระดับนัยสำคัญ .05 หรือที่ระดับความเชื่อมั่น 95 เปอร์เซ็นต์ หรือสรุปได้ว่าแบบรูปความต้องการสามารถนำไปใช้ช่วยในงานวิศวกรรมความต้องการซอฟต์แวร์ได้อย่างมีนัยสำคัญ

8.5.2 วิเคราะห์ผลการประเมินแบบรูป

ผลจากการประเมินการใช้แบบรูปสามารถสรุปออกเป็น 3 กลุ่ม ตามลักษณะหัวข้อของการประเมิน ดังนี้

1) ผลต่อการทำงาน

ผลการประเมินโดยเฉลี่ยในการประเมินกลุ่มผลต่อการทำงาน ข้อ 1 ถึง 4 คือ 4.00, 4.22, 3.78 และ 4.00 ตามลำดับ และคะแนนเฉลี่ยของทั้งกลุ่มคือ 4.00 ซึ่งแสดงว่าผู้ทดลองส่วนใหญ่เห็นด้วยว่าแบบรูปความต้องการส่งผลดีต่อการทำงาน หรือวิเคราะห์ได้ว่าผู้ใช้งานแบบรูปความต้องการคิดว่า แบบรูปความต้องการช่วยให้การทำงานง่ายขึ้น มีประสิทธิภาพ ลดเวลาในการทำงานลง และยังมีความเหมาะสมที่จะนำไปใช้กับระบบปลอดภัยเชิงวิกฤตเป็นอย่างดี

8.5.2.1 โครงสร้างและองค์ประกอบ

ผลการประเมินโดยเฉลี่ยในการประเมินกลุ่มโครงสร้างและองค์ประกอบ ข้อ 5 ถึง 7 คือ 4.33, 3.89 และ 3.67 ตามลำดับ และคะแนนเฉลี่ยของทั้งกลุ่มคือ 3.96 ซึ่งแสดงว่าผู้ทดลองส่วนใหญ่เห็นด้วยว่าแบบรูปความต้องการมีโครงสร้างและองค์ประกอบที่เหมาะสมอย่างมาก หรือวิเคราะห์ได้ว่าโครงสร้าง และองค์ประกอบของแบบรูปความต้องการมีความเหมาะสมก่อนข้างมาก

8.5.2.2 ความรู้สึกของผู้ใช้งาน

ผลการประเมินโดยเฉลี่ยในการประเมินกลุ่มความรู้สึกของผู้ใช้งาน ข้อ 8 ถึง 10 คือ 3.89, 3.89 และ 3.89 ตามลำดับ และคะแนนเฉลี่ยของทั้งกลุ่มคือ 3.89 ซึ่งแสดงว่าผู้ทดลองส่วนใหญ่มีความรู้สึกก่อนข้างดีต่อแบบรูปความต้องการ หรือวิเคราะห์ได้ว่าแบบรูปความต้องการทำให้ผู้ใช้งานพึงพอใจ เล็งเห็นการนำไปใช้ และคิดว่ามีความคุ้มค่าที่จะใช้งานก่อนข้างมาก

8.5.3 สรุปผลการทดลอง

จากที่ได้ทำการทดลองสามารถสรุปผลได้ดังต่อไปนี้

- 1) แบบรูปความต้องการสามารถนำมาช่วยในงานวิศวกรรมความต้องการซอฟต์แวร์ได้อย่างชัดเจน
- 2) แบบรูปความต้องการส่วนความปลอดภัยสามารถนำมาใช้กับการทำความเข้าใจความต้องการสำหรับระบบปลอดภัยเชิงวิกฤตได้
- 3) ความเหมาะสมในการนำไปใช้งานของแบบรูปความต้องการอยู่ในเกณฑ์ดี
- 4) ความเหมาะสมของโครงสร้างและองค์ประกอบแบบรูปความต้องการอยู่ในเกณฑ์ดี
- 5) ความพึงพอใจในการใช้งานแบบรูปความต้องการอยู่ในเกณฑ์ปานกลางก่อนไปทางดี

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 9

บทสรุป

ในบทนี้จะเป็นการสรุปผลลัพธ์ของโครงสร้างแบบรูปความต้องการที่ได้นำเสนอ โดยจะอภิปรายผลงานวิจัยในแง่มุมต่างๆ ก่อนจะทำการสรุปและให้ข้อเสนอแนะเกี่ยวกับโครงสร้างที่ออกแบบ

9.1 อภิปรายผลการวิจัย

โครงสร้างแบบรูปความต้องการที่ได้เสนอในวิทยานิพนธ์ ถือได้ว่าเป็นแนวคิดใหม่ที่นำมาใช้ในงานวิศวกรรมความต้องการซอฟต์แวร์ ในแง่ของแนวทางในการออกแบบได้ยึดถือหลักเกณฑ์ในการสร้างแบบรูป และนำแนวคิดมาผสมกับหลักการ ในการทำวิศวกรรมความต้องการซอฟต์แวร์ ซึ่งทำให้ผลลัพธ์ที่ได้มีลักษณะของแบบรูปโดยสมบูรณ์ และการนำมาใช้นั้นก็เป็นไปตามแนวทางปฏิบัติและการแก้ปัญหาที่เกิดขึ้น

ลักษณะพิเศษของโครงสร้างที่ได้นำเสนอ คือ การแยกประเภทแบบรูปออกเป็น 2 กลุ่มหลักๆ แม้ว่าการแบ่งประเภทของแบบรูปจะมีให้เห็นอยู่บ้าง เช่น แบบรูปการออกแบบ [4] แบบรูปธุรกิจ [29] เป็นต้น แต่ก็ไม่ได้มีลักษณะแตกต่างกันมากนัก โดยแบบรูปที่นำเสนอในงานมีความแตกต่างทั้งในลักษณะการนำไปใช้งานไปถึงองค์ประกอบของแบบรูป ซึ่งมีความแตกต่างกันมากและอาจมองว่าทำให้การใช้งานซับซ้อนหรือยุ่งยากขึ้น แต่การแบ่งประเภทของแบบรูปออกเป็น 2 กลุ่ม กลับทำให้เกิดผลดี ทั้งในแง่ของการใช้งานทำให้เกิดความสมบูรณ์และชัดเจนในแบบรูปแต่ละประเภท เนื่องจากแบบรูปทั้ง 2 ประเภทสามารถใช้งานแยกกัน โดยสามารถเลือกใช้งานแบบรูปเพียงแค่ประเภทใดประเภทหนึ่ง เพราะระบบแบบรูปเป็นเพียงแต่การเพิ่มประสิทธิภาพของการใช้งานเมื่อนำมาใช้ร่วมกันเท่านั้น

องค์ประกอบของแบบรูปโดเมนความต้องการพิจารณาจากผลลัพธ์จากกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ในบริบทของโดเมนนั้น ซึ่งเป็นการบันทึกส่วนสำคัญในด้านต่างๆ ที่มีความเหมาะสมและจำเป็น โดยเฉพาะในส่วนผลเฉลยของแบบรูปโดเมนความต้องการสามารถนำมาใช้สร้างเอกสารข้อกำหนดความต้องการซอฟต์แวร์ได้

องค์ประกอบของแบบรูปกระบวนการความต้องการเป็นส่วนที่พิจารณาจากการใช้เทคนิคหรือวิธีการต่างๆ ในวิศวกรรมความต้องการซอฟต์แวร์ ซึ่งทำให้สามารถกำหนดส่วนจำเป็นที่ใช้เป็นองค์ความรู้สำหรับใช้ในเทคนิคหรือวิธีการนั้นๆ ได้ ซึ่งรูปแบบและองค์ประกอบที่กำหนดไว้จะช่วยให้การใช้แบบรูปกระบวนการความต้องการเป็นแนวทางเสริมในการทำงานที่มีความชัดเจน ทำให้เข้าใจในเทคนิคหรือวิธีการมากขึ้น

องค์ประกอบของแบบรูปความต้องการส่วนความปลอดภัยเป็นการเพิ่มเติมจากองค์ประกอบแบบรูปความต้องการ โดเมน ซึ่งถือเป็นข้อดีที่ทำให้แบบรูปความต้องการส่วนความปลอดภัยไม่แตกต่างไปจาก

แบบรูปโดเมนความต้องการมากนัก โดยเฉพาะการใช้งานมีความเป็นไปได้ในแนวทางเดียวกันไม่เกิดความสับสน และส่วนที่เพิ่มเข้ามาก็มีจุดประสงค์ที่ชัดเจนที่จะช่วยในการทำงานที่มีประเด็นเกี่ยวกับความปลอดภัย

การพิจารณาการใช้งานแบบรูปความต้องการในวิทยานิพนธ์นี้ ได้เสนอแนวทางที่สอดคล้องกับการทำงานในกระบวนการวิศวกรรมซอฟต์แวร์ และได้มองไปยังการใช้งานที่สัมพันธ์กับแบบรูปประเภทอื่นๆ อีกด้วย ซึ่งจะเห็นได้ว่าการนำแบบรูปความต้องการมาใช้งานจะไม่ทำให้การทำงานเปลี่ยนแปลงไปจากปกติและน่าจะทำให้กระบวนการพัฒนาซอฟต์แวร์ทำได้ดียิ่งขึ้น

สำหรับเครื่องมือที่สร้างขึ้นเพื่อสนับสนุนการสร้างและใช้งานแบบรูปความต้องการน่าจะสามารถช่วยให้การใช้งานแบบรูปมีประสิทธิภาพมากขึ้น จากการออกแบบการทำงานของเครื่องมือจะเห็นว่า การพัฒนาเครื่องมือได้ให้ความสำคัญในเรื่องของง่ายในการใช้งาน ความสอดคล้องกับลักษณะการใช้งานแบบรูปความต้องการ และความยืดหยุ่นในการนำข้อมูลแบบรูปไปใช้ในแง่ต่างๆ

ในวิทยานิพนธ์นี้นอกจะมีการเสนอ โครงสร้างและองค์ประกอบของแบบรูปความต้องการ ยังมี การสร้างแบบรูปความต้องการที่สามารถใช้ได้จริงขึ้นมา โดยมีจุดประสงค์หลักเพื่อทำการทดสอบการใช้งาน ซึ่งจากการทดลองก็สามารถสรุปได้ว่าแบบรูปความต้องการสามารถนำไปใช้งานได้จริง และความเห็นของคนส่วนใหญ่ที่ใช้ยังเห็นว่าแบบรูปความต้องการมีโครงสร้างและองค์ประกอบที่เหมาะสม มีความพึงพอใจในการใช้งาน และเห็นว่าเหมาะสมกับการนำไปใช้งานจริง

9.2 สรุปผลการวิจัย

- 1) การออกแบบแบบรูปความต้องการอาศัยแนวทางในการสร้างแบบรูปซึ่งพิจารณาจากปัญหาที่เกิดขึ้นซ้ำๆ ในสถานะแวดล้อมที่สนใจ จึงทำให้เกิดการพิจารณาแบบรูปใน 2 แนวทางจากลักษณะปัญหาในวิศวกรรมความต้องการซอฟต์แวร์ ที่ขึ้นกับโดเมน และที่ขึ้นกับกระบวนการ
- 2) โครงสร้างแบบรูปความต้องการที่เสนอประกอบด้วยแบบรูป 3 ประเภท คือ แบบรูปโดเมนความต้องการ แบบรูปกระบวนการความต้องการ และแบบรูปความต้องการส่วนความปลอดภัย
- 3) แบบรูปโดเมนความต้องการคือแบบรูปที่ได้จากการพิจารณาในแนวทางของปัญหาที่จำกัดจากโดเมน
- 4) แบบรูปกระบวนการความต้องการคือแบบรูปที่ได้จากการพิจารณาในแนวทางปัญหาที่เกิดจากกระบวนการการทำงาน
- 5) แบบรูปความต้องการส่วนความปลอดภัยเป็นการขยายลักษณะของแบบรูปโดเมนความต้องการเพื่อเจาะจงไปยังการใช้งานกับระบบปลอดภัยเชิงวิกฤต

- 6) แบบรูปทั้ง 3 ชนิดมีความเชื่อมโยงและสัมพันธ์กัน การใช้แบบรูปเพียงชนิดใดชนิดหนึ่งอย่างเดียวยังสามารถทำได้ แต่การนำมาใช้งานร่วมกันจะเกิดประสิทธิภาพสูงกว่า
- 7) ได้มีการเสนอรูปแบบขององค์ประกอบในแบบรูปความต้องการแต่ละชนิดเพื่อให้เกิดความชัดเจนและมีความเข้าใจที่ถูกต้องในการใช้งาน
- 8) มีการพัฒนาเครื่องมือและเสนอแนวทางที่เหมาะสมในการใช้งานแบบรูปความต้องการประกอบในวิทยานิพนธ์
- 9) จากการทดลองสามารถสรุปได้ว่าโครงสร้างแบบรูปความต้องการที่เสนอมีความเหมาะสมที่จะนำไปใช้งานจริง
- 10) การนำโครงสร้างแบบรูปความต้องการที่เสนอไปใช้งานน่าจะเกิดประโยชน์อย่างยิ่งเมื่อพิจารณาในแง่ของการนำกลับมาใช้ใหม่ และการแก้ปัญหาเรื่ององค์ความรู้ในงานวิศวกรรมความต้องการซอฟต์แวร์ ซึ่งจะเห็นได้ว่าบรรทัดอุปสงค์ในการออกแบบโครงสร้างแบบรูปความต้องการ

9.3 ข้อเสนอแนะ

แบบรูปความต้องการที่เสนอในวิทยานิพนธ์นี้เป็นการวิเคราะห์ในแง่ของวิศวกรรมความต้องการสำหรับลักษณะโดเมนทั่วไป และเจาะจงเฉพาะส่วนของระบบปลอดภัยเชิงวิกฤต ดังนั้นการใช้งานจึงอาจจะไม่ครอบคลุมระบบในทุกประเภทโดเมน เช่น ระบบในโดเมนของเกมส์ ระบบในโดเมนที่เกี่ยวข้องกับความมั่นคง (Security) เป็นต้น ดังนั้นหากมีความเป็นไปได้จึงควรจะพิจารณาให้ครอบคลุมในระบบกลุ่มอื่นๆ ด้วย

การใช้งานแบบรูปความต้องการมีรายละเอียดค่อนข้างมาก การอธิบายแต่ละส่วนแม้จะมีการกำหนดรูปแบบที่ชัดเจนแล้วแต่ก็ยังขาดความชัดเจนทางภาษา ดังนั้นหากมีการประยุกต์นำเอาภาษารูปนัย (Formal Language) มาใช้ร่วมด้วย น่าจะทำให้การใช้งานแบบรูปมีประสิทธิภาพและชัดเจน หรือสามารถพิสูจน์และตรวจสอบความถูกต้องได้ดียิ่งขึ้น

จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

1. Brad Appleton. Patters and software: Essential concepts and terminology. Object Magazine online V.3, 5 (May 1997): 20-25.
2. S. Fraser, et al. Patterns: Cult to culture?. Proceedings of the 10th annual conference on Object-oriented programming systems languages and applications (OOPSLA) Vol. 30 Issue10, pp. 231-234. October, 1995.
3. Christopher Alexander. A Pattern Language: Towns, Buildings, Construction. Oxford University Press, 1977.
4. E. Gamma, R. Helm, R. Johnson and J. Wlissides. Design Patterns Elements of Reusable Object-Oriented Software. 7th Indian Reprint Edition. Pearson Education, 2002.
5. Patricia L. Ferdinandi. A Requirements Pattern: Succesing in the Internet Economy. Addison-Wesley, June 2001.
6. S.Konrad and Betty H.C. Cheng. Requirements patterns for embedded systems. Proceedings of IEEE Joint International Conference on Requirements Engineering (RE'02), pp. 127-136. 2002.
7. Suzanne Robertson. Requirements patterns via events/use cases. The Atlantic Systems Guild, 1996.
8. Toshihiko Tsumaki. Requirements engineering pattern structure. Proceeding of IEEE The 11th Asia-Pacific software Engineering Conference (APSEC'2004), pp. 502-509. 2004.
9. L. Hagge, K. Lappe. Patterns for the RE process. Proceedings of IEEE The 12th International Requirements Engineering Conference (RE'04), pp. 90-99. 2004.
10. Nancy G. Leveson. Software: system safety and computers. Addison-Wesley, 1995.
11. David Alberico, et al. Software System Safety Handbook: A Technical & Managerial Team Approach. Joint Sevices Computer Resources Management Group, U.S. Navy, U.S Army and the U.S. Air Force, December 1999.
12. Robyn Lutz. Software engineering for safety: A roadmap. Proceedings of the conference on The Future of Software Engineering. International Conference on Software Engineering, pp. 213-226. 2000.
13. Gerard Meszaros and Jim Doble. A pattern language for pattern writing. Pattern languages of program design 3, pp. 529-574. 1997.

14. K. Beck and W. Cunningham. Using pattern languages for object-oriented programs. Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'87), December, 1987.
15. Martin Fowler. Analysis Patterns: Reusable Object Models. Addison-Wesley, 1996.
16. James O. Coplien. Advanced C++ Programming Styles and Idioms. Addison-Wesley, 1991.
17. Frank Buschmann, et al. Pattern-Oriented Software Architecture, Volume 1: A System of Patterns. John Wiley & Sons, 1996.
18. Mark Grand. Patterns in Java, Volume 1: A Catalog of Reusable Design Patterns Illustrated with UML. John Wiley & Sons, 1998.
19. G. Kotonya and I. Sommerville. Requirements Engineering: processes and techniques. John Wiley & Sons, 1997.
20. D. Leffingwell and D. Widrig. Managing Software Requirements: A Use Case Approach. Second Edition. Pearson Education, 2003.
21. IEC 16508, Functional safety of electrical/electronic/programmable electronic safety-related systems. International Electrotechnical Commission (IEC), December 1997.
22. Edmund M. Clarke and Jeannette M. Wing. Formal methods: State of the art and future directions. ACM Computing Surveys, V.28, 4 (December 1996): 624-643.
23. F. Modugno, Nancy G. Leveson, et al. Integrated safety analysis of requirements specifications. Proceedings of the Third IEEE International Symposium on Requirements Engineering, pp. 148-159. January, 1997.
24. Donald Firesmith. A taxonomy of safety-related requirements. 3rd International Workshop on Requirements for High Assurance System (RHAS 2004), 2004.
25. I. Sommerville. Software engineering. 7 Edition. Addison Wesley, May 2004.
26. MIL-STD-882D, Standard practice for system safety. Department of Defense (DoD), February 10, 2000.
27. M. Mahemoff, A. Hussey and L. Johnston. Pattern-based reuse of successful designs: Usability of safety-critical systems. Proceedings of Software Engineering Conference (ASWEC'01), pp.31-39. August 2001.
28. I. Sommerville and P. Sawyer. Requirements Engineering: A good practice guide. John Wiley & Sons, 1997.

29. Magnus Penker, Penker and Hans-Erik Eriksson, Business Modeling With UML: Business Patterns at Work. John Wiley & Sons, 2000.
30. IEEE Std 1474.1-1999, IEEE Standard for Communications-Based Train Control(CBTC) Performance and Functional Requirements, September 16, 1999.



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



ภาคผนวก

สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

บทความวิชาการ

บทความวิชาการของผู้วิจัยที่ได้รับการคัดเลือกและตีพิมพ์ในการประชุมวิชาการระดับชาติ คือ

บทความวิชาการเรื่อง “การนำความต้องการซอฟต์แวร์กลับมาใช้ใหม่ด้วยแบบรูปความต้องการ (Software Requirements Reuse by Requirements Patterns)” ซึ่งได้นำเสนอและตีพิมพ์ในงาน “การประชุมทางวิชาการทางวิทยาศาสตร์และวิศวกรรมคอมพิวเตอร์ระดับชาติ ครั้งที่ 9 (The 9th National Computer Science and Engineering Conference: NCSEC 2005)” ระหว่างวันที่ 27-28 ตุลาคม 2548 ณ มหาวิทยาลัยหอการค้าไทย กรุงเทพฯ



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

การนำความต้องการซอฟต์แวร์กลับมาใช้ใหม่ด้วยแบบรูปความต้องการ Software Requirements Reuse by Requirements Patterns

เอกชัย แซ่ตั้ง และ นครทิพย์ พร้อมพูล

ห้องปฏิบัติการวิศวกรรมซอฟต์แวร์ ศูนย์เชี่ยวชาญเฉพาะทางด้านวิศวกรรมซอฟต์แวร์

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

เขตปทุมวัน กรุงเทพฯ

อีเมล: Eakachai.C@gmail.com, Nakornthip.S@chula.ac.th

Abstract

Pattern is one form of software reuse which significantly beneficial to developer develops quality software. It is always used in system design and implement phase. If pattern is used in the early phase such as requirement engineering, it will effectively improve the quality of software development. This research presents two types of patterns used for requirements engineering: requirements domain patterns and requirements process patterns. These patterns a proposed based on the problem domain and software requirements engineering process.

Key Words: Software Engineering, Software Requirements Engineering, Pattern, Requirements Pattern, Requirements Reuse

บทคัดย่อ

แบบรูป (Pattern) เป็นลักษณะหนึ่งของการนำกลับมาใช้ใหม่ ซึ่งนำไปใช้เพิ่มคุณภาพในการพัฒนาซอฟต์แวร์ แบบรูปมักถูกนำมาใช้ในขั้นตอน การออกแบบ และการพัฒนาโปรแกรม ถ้าสามารถนำแบบรูปมาใช้ในการพัฒนาในเฟสต้นๆของวิศวกรรมความต้องการซอฟต์แวร์ได้ ก็จะสามารถเพิ่มประสิทธิภาพ ของการพัฒนาซอฟต์แวร์ได้อีกมาก งานวิจัยนี้จึงได้นำเสนอแบบรูป 2 ประเภทสำหรับ วิศวกรรมความต้องการซอฟต์แวร์ ซึ่งคือ แบบรูปโดเมน ความต้องการ และแบบรูปกระบวนการความต้องการ ซึ่ง

นำเสนอมาจากปัญหาในโดเมนและกระบวนการใน วิศวกรรมความต้องการซอฟต์แวร์

คำสำคัญ: วิศวกรรมซอฟต์แวร์ วิศวกรรมความต้องการ ซอฟต์แวร์ แบบรูป แบบรูปความต้องการ การนำความต้องการกลับมาใช้ใหม่

1. บทนำ

การพัฒนาซอฟต์แวร์ในยุคปัจจุบัน วิศวกรรมความต้องการซอฟต์แวร์มีความสำคัญมากขึ้นเรื่อยๆ เนื่องจากพบว่าบ่อยครั้ง ที่การพัฒนาซอฟต์แวร์เกิดความล้มเหลวอันเนื่องมาจาก การไม่ให้ความสำคัญกับกระบวนการ วิศวกรรมความต้องการซอฟต์แวร์ [1] ทำให้ได้ความต้องการที่ไม่สมบูรณ์ ไม่ได้รับความต้องการที่แท้จริงของผู้ใช้ ไม่สามารถรองรับเปลี่ยนแปลงความต้องการได้

แบบรูปเป็นที่รู้จักกันอย่างแพร่หลาย จากการนำไปใช้กับการออกแบบ [2] แต่โดยค้นกำเนิดแล้วแบบรูปมีขึ้นเพื่อนำไปใช้กับงานด้านสถาปัตยกรรม [3] แล้วจึงนำมาใช้ในการออกแบบส่วนต่อประสานผู้ใช้ [4] เป็นครั้งแรก ปัจจุบันมีผู้นำแบบรูปมาใช้ในขั้นตอนต่างๆของการพัฒนาซอฟต์แวร์เป็นจำนวนมาก [5, 6]

การนำแบบรูปมาใช้ในการวิศวกรรมความต้องการ ซึ่งเรียกว่าแบบรูปความต้องการ (Requirements Pattern) มี

เป้าหมายในการแก้ปัญหาที่เกิดขึ้นในลักษณะเดียวกันบ่อยๆ มุ่งหวังผลลัพธ์ที่เป็นข้อกำหนดความต้องการของระบบ เพื่อนำไปใช้ในการพัฒนาซอฟต์แวร์

อนึ่งการใช้แบบรูปในปัจจุบัน มีความครอบคลุมในทุกส่วนของกระบวนการพัฒนาซอฟต์แวร์ ดังนั้นการทำแบบรูปให้มีความสมบูรณ์ และสามารถเชื่อมต่อไปถึงแบบรูปส่วนอื่นๆ ย่อมเพิ่มความสามารถในการนำกลับมาใช้ใหม่ และเป็นผลดีโดยรวม ในวงจรการพัฒนาซอฟต์แวร์ (Software Development Life Cycle: SDLC) ที่

งานวิจัยนี้ทำการวิเคราะห์ลักษณะปัญหา ที่พบอยู่เสมอในงานวิศวกรรมความต้องการซอฟต์แวร์ โดยวิเคราะห์ตามมุมมองที่ขึ้นกับโดเมนของระบบ ประกอบกับแนวทางของกระบวนการทางวิศวกรรมความต้องการซอฟต์แวร์ เพื่อให้ได้ลักษณะแบบรูปความต้องการที่เหมาะสม สามารถนำมาใช้กับงานวิศวกรรมความต้องการซอฟต์แวร์

โดยงานวิจัยนี้ได้แบ่งหัวข้อออกเป็น 6 หัวข้อด้วยกัน โดยหัวข้อที่ 2 ที่ เป็นหัวข้อถัดไปจะกล่าวถึงงานวิจัยที่เกี่ยวข้อง ในหัวข้อที่ 3 จะกล่าวถึงหลักการและการวิเคราะห์การนำกลับมาใช้ใหม่สำหรับวิศวกรรมความต้องการซอฟต์แวร์ ก่อนจะสรุปออกมาเป็นโครงสร้างแบบรูปทั้ง 2 ประเภทในหัวข้อที่ 4 ในหัวข้อที่ 5 จะยกตัวอย่างกรณีศึกษาของการใช้แบบรูปกับระบบห้องสมุด และสรุปงานวิจัยในหัวข้อที่ 6

2. งานวิจัยที่เกี่ยวข้อง

แบบรูปเริ่มนำมาใช้กับ วิศวกรรมความต้องการซอฟต์แวร์ครั้งแรก โดย S. Robertson [7] ซึ่งเป็นแบบรูปที่ได้จากการวิเคราะห์ความต้องการจากเหตุการณ์ และแผนภาพยูสเคส (Use Case)

Sascha Konrad และ Betty H.C. Cheng ได้ใช้แบบรูปความต้องการกับระบบฝังตัว (Embedded Systems) [8] ทางด้าน Romi Satria Wohomo และ Jingde Cheng ได้นำแบบรูปความต้องการไปใช้กับ การพัฒนาโปรแกรมประยุกต์บนเว็บ [9] แบบรูปทั้งสองมีลักษณะการใช้ที่ขึ้นอยู่กับโดเมนของระบบเหมือนกัน แต่โครงสร้างของแบบรูป

กลับมีลักษณะที่ต่างกัน ซึ่งทั้งสองแบบประยุกต์โครงสร้างแบบรูปที่ใช้จากแบบรูปการออกแบบ (Design Pattern) คนละตัวกัน ทั้งนี้โครงสร้างส่วนที่แต่ละงานวิจัยเพิ่มเติมเข้ามาขึ้นอยู่กับลักษณะโดเมนการนำไปใช้

งานวิจัยสองชิ้นข้างต้นมีมุมมอง การใช้แบบรูปความต้องการที่ขึ้นอยู่กับโดเมนของระบบ แต่งานของ Marta Pantoquihio และคณะ [10] ให้มุมมองที่ต่างออกไป โดยพิจารณาแบบรูปการวิเคราะห์ (Analysis Pattern) เทียบกับแบบรูปความต้องการ ซึ่งให้ข้อสังเกตว่างานของ Konrad ที่กล่าวถึงในข้างต้นก็ไม่แตกต่างไปจากแบบรูปการวิเคราะห์ และเสนอลักษณะแบบรูปที่รวมลักษณะแบบรูปความต้องการ และแบบรูปการวิเคราะห์เข้าด้วยกัน

จะเห็นว่าแบบรูปความต้องการส่วนใหญ่จะเน้นไปที่ความต้องการ แต่ Tsumaki [11] ได้นำเสนอแบบรูปวิศวกรรมความต้องการ (Requirements Engineering Pattern) ซึ่งเป็นการนำแบบรูปไปใช้กับกระบวนการทางวิศวกรรมความต้องการซอฟต์แวร์ โดยพิจารณาไปยังสถานการณ์ของปัญหา และแสดงผลลัพธ์ของแบบรูปเป็นวิธีการที่จะนำมาใช้แก้ปัญหาแทน

ในการนำแบบรูปไปใช้สำหรับวิศวกรรมความต้องการซอฟต์แวร์ จะเห็นได้ว่าด้านหนึ่งพิจารณาตามลักษณะของโดเมนของระบบ อีกด้านหนึ่งพิจารณาตามลักษณะกระบวนการ ซึ่งทั้ง 2 ลักษณะมีข้อดีแตกต่างกัน งานวิจัยนี้จึงได้นำเสนอแบบรูปความต้องการ ที่มีข้อดีของแบบรูปทั้ง 2 ลักษณะ กำหนดเป็นโครงสร้างที่เหมาะสมเพื่อให้เกิดประสิทธิภาพในการนำไปใช้ และครอบคลุมงานวิศวกรรมความต้องการซอฟต์แวร์

3. การวิเคราะห์วิศวกรรมความต้องการซอฟต์แวร์เพื่อการนำกลับมาใช้ใหม่

แบบรูป คือ ปัญหา (Problem) และผลเฉลย (Solution) ซึ่งเกิดขึ้นซ้ำๆกัน [12] การวิเคราะห์หาโครงสร้างแบบรูปจึงต้องทำการพิจารณาไปยังปัญหา และวิธีการที่จะได้ผลเฉลย และอาศัยมุมมองของการนำกลับมาใช้ใหม่ตามที่วิเคราะห์ ดังนี้

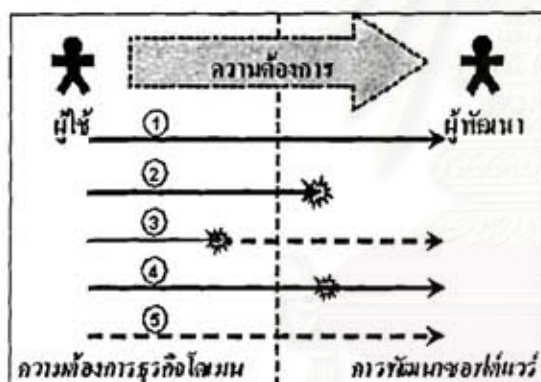
3.1 ปัญหาในวิศวกรรมความต้องการซอฟต์แวร์

ลักษณะปัญหาที่พิจารณา ในวิศวกรรมความต้องการซอฟต์แวร์ แบ่งออกเป็น 2 กลุ่ม คือ กลุ่มปัญหาที่สัมพันธ์กับโดเมนของระบบ และ กลุ่มปัญหาที่เกี่ยวข้องกับเทคนิคและกระบวนการ

1) ปัญหาด้านโดเมน

มีสาเหตุที่สำคัญมาจากองค์ประกอบของความ ต้องการมีความคลาดเคลื่อนซึ่งเกิดขึ้น 3 ลักษณะ ดังนี้

- ผู้ใช้ (User) เช่น ผู้ใช้ให้ความต้องการ ไม่ชัดเจน
- ผู้พัฒนา (Developer) เช่น ผู้พัฒนาทราบความต้องการ ไม่ครบ
- การสื่อสาร (Communicate) เช่น ผู้ใช้และผู้พัฒนา มีความเข้าใจไม่ตรงกัน



รูปที่ 1 แบบจำลองปัญหาในวิศวกรรมความต้องการซอฟต์แวร์

รูปที่ 1 แสดงแบบจำลองลักษณะของปัญหาใน วิศวกรรมความต้องการซอฟต์แวร์ ซึ่งเห็นได้ว่าปัญหา สามารถเกิดขึ้นได้ทั้งในมุมมองความต้องการธุรกิจโดเมน (Business Domain) และมุมมองการพัฒนาซอฟต์แวร์ โดย สามารถจำแนกลักษณะความต้องการได้เป็น 5 ประเภท ตามรูป ซึ่งมีความหมาย ดังนี้

ประเภทที่ 1

ความต้องการที่ส่งไปยังผู้พัฒนาโดยไม่เกิดปัญหา

ประเภทที่ 2

ความต้องการที่ไม่สามารถเกิดขึ้นได้ เนื่องจากไม่สามารถหลีกเลี่ยงปัญหาที่พบ เช่น ความต้องการซับซ้อนมาก ความต้องการไม่สามารถพัฒนาให้ใช้ได้จริง เป็นต้น

ประเภทที่ 3

ความต้องการที่เบี่ยงเบนหรือผิดพลาดเนื่องจาก ปัญหาระหว่างกระบวนการวิศวกรรมความต้องการ ซอฟต์แวร์

ประเภทที่ 4

ความต้องการที่นำไปใช้พัฒนาได้ตามปกติ แม้มี ปัญหาเกิดขึ้น

ประเภทที่ 5

ความต้องการที่ผิดพลาดมาแต่ต้น เป็นผลให้การ นำไปใช้เกิดความผิดพลาดตามมา โดยไม่รู้ตัว

2) ปัญหาด้านกระบวนการ

ขึ้นอยู่กับกลุ่มกระบวนการหลักทางวิศวกรรม ความต้องการซอฟต์แวร์ที่นำมาใช้ เกิดปัญหาและความ ผิดพลาดขึ้น ซึ่งกลุ่มกระบวนการทั้งหมด [13, 14] ประกอบด้วย

- การเก็บรวบรวมความต้องการ
- การวิเคราะห์ความต้องการ
- การจัดทำเอกสารความต้องการ
- การประเมินความต้องการ
- การบริหารความต้องการ

ความสัมพันธ์ของกระบวนการวิศวกรรม ความ ต้องการซอฟต์แวร์ สามารถพิจารณาได้ตามรูปที่ 2 โดยเป็น กระบวนการที่สัมพันธ์กับความต้องการ จากมุมมองด้าน ธุรกิจโดเมน และมุมมองด้านการพัฒนาซอฟต์แวร์ ซึ่ง กระบวนการทางวิศวกรรมความต้องการซอฟต์แวร์ อาจ อาศัยวิธีการที่เป็นรูปนัย (Formal) กึ่งรูปนัย (Semi-formal) หรือไม่เป็นรูปนัย (Informal) แต่โดยมากแล้ว กระบวนการที่เกิดขึ้นมักจะอาศัยวิธีการที่เป็นรูปนัยหรือกึ่ง รูปนัยทั้ง กระบวนการประเมิน กระบวนการตรวจสอบ และกระบวนการจัดทำเอกสาร

เมื่อเปรียบเทียบกับโมเดลของปัญหาสามารถสรุปได้ว่า การเก็บความต้องการจะเป็นการเริ่มต้นส่งต่อความต้องการ

ในธุรกิจโดเมนไปสู่ผู้พัฒนาซอฟต์แวร์ การจัดทำเอกสารจะเป็นการจัดระเบียบความต้องการ ให้เกิดความชัดเจนมากขึ้น และสามารถนำไปประเมินและวิเคราะห์ความต้องการต่อไป โดยการประเมินจะทำได้กับความต้องการโดยอิสระเพื่อพิจารณากลับไปว่า ความต้องการเป็นไปตามความต้องการเริ่มต้นของผู้ใช้หรือไม่ ส่วนการพิจารณาระหว่างความต้องการว่าตรงหรือขัดแย้งหรือก่อให้เกิดปัญหามา จะเป็นหน้าที่ของกระบวนการวิเคราะห์ความต้องการ

สำหรับกระบวนการบริหารความต้องการจะเป็นกระบวนการที่พิจารณาในระดับบนควบคู่กับการบริหารโครงการ ซึ่งจะมาจัดการกับความเปลี่ยนแปลงความต้องการ การกำหนดขั้นตอนและกระบวนการ



รูปที่ 2 แบบจำลองปัญหา มุมมอง และกระบวนการ ในวิศวกรรมความต้องการซอฟต์แวร์

การใช้เทคนิคและวิธีการที่เหมาะสมที่ใช้กันเสมอ เรียกว่าเป็นวิธีการปฏิบัติ (Practice) [1] หากพิจารณาตามลักษณะกระบวนการ วิธีการปฏิบัติเป็นขั้นตอนที่ช่วยลดปัญหาที่เกิดขึ้นโดยไม่ขึ้นกับลักษณะโดเมน หากพิจารณาในรูปที่ 2 วิธีการปฏิบัติก็คือความต้องการที่ 1 และ 2 ที่ส่งไปพัฒนาซอฟต์แวร์ได้อย่างสมบูรณ์ ซึ่งวิธีการปฏิบัติก็จะมีส่วนช่วยลด และตรวจพบปัญหาได้ ดังเช่น ความต้องการที่ 2 ในรูป อย่างไรก็ตามวิธีการปฏิบัตินี้จะไปอยู่ที่กระบวนการ ดังนั้นวิธีการปฏิบัติจึงเป็นแนวทางที่เป็นกลางไม่ขึ้นกับโดเมนโดเมนหนึ่ง ซึ่งถือเป็นข้อดีหากต้องการที่จะมีการเจาะจงการทำงานที่เป็นไปในแนวทางของโดเมนใดเป็นพิเศษ

3.2 การนำกลับมาใช้ใหม่สำหรับวิศวกรรมความต้องการซอฟต์แวร์

จากลักษณะปัญหาที่พิจารณาจะเห็นได้ว่าปัญหาทั้ง 2 กลุ่ม มีที่มาและพฤติกรรมที่ต่างกัน ปัญหาจากโดเมนของระบบ เป็นปัญหาที่พบอีกครั้งเมื่อมีการพัฒนาระบบใหม่ที่มีความคล้ายคลึงกับระบบเดิม ซึ่งเป็นปัญหามักจะขึ้นอยู่กับลักษณะของโดเมน แต่ปัญหาของกระบวนการเป็นปัญหาที่เกิดจากการทำงาน ซึ่งในหลายๆครั้งเป็นปัญหาในการเลือกเทคนิคหรือวิธีการ ให้เหมาะสมกับโดเมนที่กำลังพัฒนาอยู่ และมักจะไม่ใช่ติดกับประเภทของโดเมน

ดังที่แสดงในรูปที่ 2 การที่ทราบปัญหาที่เกิดขึ้นไม่ว่าจากมุมมองผู้ใช้ในธุรกิจโดเมน และมุมมองผู้พัฒนาในส่วนการพัฒนาซอฟต์แวร์ เราจะสามารถหลีกเลี่ยงปัญหาหรือหาวิธีการในการจัดการกับปัญหาได้ดีกว่า ซึ่งถือได้ว่าเป็นการนำองค์ความรู้กลับมาใช้ใหม่ และการทราบกระบวนการที่เหมาะสมตลอดจนวิธีการปฏิบัติที่ทราบแล้วว่าเป็นเทคนิค และวิธีการที่ประสบความสำเร็จก็จะช่วยให้ทราบและเข้าใจปัญหาในการใช้เทคนิควิธีการนั้น ช่วยให้สามารถตัดสินใจใช้เทคนิคและวิธีการได้อย่างถูกต้อง ซึ่งเป็นการนำเทคนิควิธีการกลับมาใช้ใหม่

ดังนั้นในการนำความต้องการกลับมาใช้ใหม่ จะทำการหาองค์ประกอบที่จะนำมาใช้แก้ปัญหา หรือองค์ประกอบที่อธิบายหรือช่วยในการทำงาน โดยองค์ประกอบความรู้ที่จะนำมาใช้ในการนำกลับมาใช้ใหม่ เกิดจากการแยกพิจารณา มุมมองของของปัญหา โดยขึ้นกับโดเมนและกระบวนการตามที่กล่าวถึงข้างต้น ซึ่งทำให้้องค์ประกอบความรู้ที่จะนำมาใช้ในการนำกลับมาใช้ใหม่ แบ่งออกเป็น 2 กลุ่มด้วยกัน คือ กลุ่มที่แสดงด้านโดเมนซึ่งทำให้เห็นปัญหาและเข้าใจในลักษณะของโดเมน และกลุ่มที่แสดงด้านกระบวนการ ซึ่งช่วยเพิ่มความเข้าใจและลดปัญหาในการใช้เทคนิคหรือวิธีการ โดยสามารถสรุปองค์ประกอบความรู้ได้ดังนี้

• องค์ประกอบความรู้ด้านโดเมน ประกอบด้วย

- 1) เป้าหมาย
- 2) ความหมายและนิยาม

- 3) ลักษณะของโดเมน
- 4) ความต้องการ
- 5) ปัญหาของโดเมน
- 6) ข้อกำหนดและกฎเกณฑ์
- 7) ปัญหาของการพัฒนา
- 8) สภาพแวดล้อมของระบบ

• องค์ประกอบความรู้ด้านกระบวนการ ประกอบด้วย

- 1) สถานการณ์
- 2) จุดประสงค์
- 3) กลุ่มประเภทของระบบ
- 4) ประสิทธิภาพ
- 5) ขั้นตอนวิธี
- 6) เครื่องมือ
- 7) ค่าใช้จ่าย

องค์ประกอบความรู้ด้านโดเมน จะช่วยให้เข้าใจและทราบปัญหาในมุมมองต่างๆ ทำให้สามารถแก้ไขหรือหลีกเลี่ยงปัญหาคามลักษณะโดเมน ส่วนองค์ประกอบ

ใหม่ในลักษณะการนำกลับมาใช้ใหม่ทางองค์ความรู้ และการนำกลับมาใช้ใหม่ทางวิธีการ

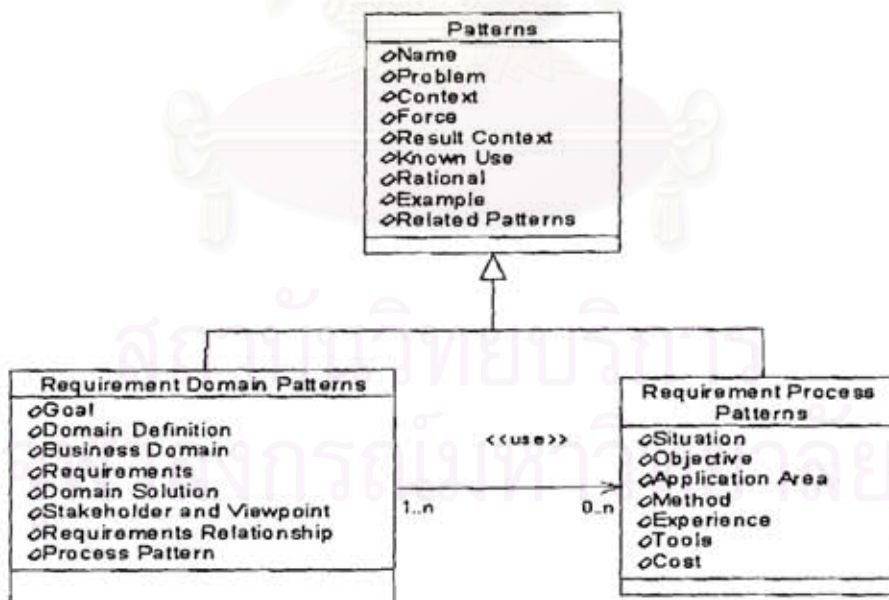
4. การออกแบบโครงสร้างแบบรูปความต้องการซอฟต์แวร์

โครงสร้างของแบบรูปความต้องการ เกิดจากการนำเอาองค์ประกอบความรู้ ที่ได้จากการวิเคราะห์ปัญหาและการนำกลับมาใช้ใหม่ ทางวิศวกรรมความต้องการซอฟต์แวร์มาสรุปเป็นลักษณะของแบบรูปที่เหมาะสม

การกำหนดโครงสร้างแบบรูปให้เหมาะสม ไม่มีลักษณะตายตัว แต่เมื่อพิจารณาตามลักษณะขององค์ความรู้ในการนำความต้องการซอฟต์แวร์กลับมาใช้ใหม่ ร่วมกับการใช้แบบรูปในการสร้างแบบรูป [15] ทำให้สามารถวิเคราะห์และสรุปแบบรูปได้ ดังนี้

4.1 ลักษณะและประเภทของแบบรูป

แบบรูปสำหรับวิศวกรรมความต้องการซอฟต์แวร์ เมื่อพิจารณาจากปัญหาซ้ำเพื่อหาผลเฉลยที่เหมาะสม จะพบว่ามี



รูปที่ 3 องค์ประกอบและความสัมพันธ์ของแบบรูปความต้องการซอฟต์แวร์

ความรู้ด้านกระบวนการจะเป็นเครื่องมือในการตัดสินใจใช้เทคนิคหรือวิธีการที่มีประสิทธิภาพวิธีปฏิบัติ ซึ่งทั้งหมดเป็นองค์ความรู้ที่เพียงพอในการนำความต้องการกลับมาใช้

ปัญหาใน 2 กลุ่ม ตามที่เสนอในเบื้องต้น คือ ปัญหาของโดเมน และปัญหาของกระบวนการ และตามงานที่มีผู้

นำเสนอไปก่อนหน้า มีทั้งการนำเสนอแบบรูปที่แก้ปัญหาเพียงแค่ด้านโดเมนหรือกระบวนการเพียงด้านเดียว

ดังนั้นแบบรูปที่วิเคราะห์ตามงานวิจัยนี้จึงควรประกอบไปด้วยทั้ง 2 ส่วน แต่ก็จะมีลักษณะองค์ประกอบในเบื้องต้นของแบบรูปที่เหมือนกันอยู่ ได้แก่ ชื่อ (Name) ปัญหา (Problem) สภาพแวดล้อม (Context) พอร์ซ (Force) แบบรูปที่เกี่ยวข้อง (Related Patterns) และตัวอย่าง (Example) โดยแบบรูปทั้ง 2 ประเภท กำหนดชื่อเรียก ความหมาย และรายละเอียดไว้ ดังนี้

- 1) แบบรูปโดเมนความต้องการ
คือ แบบรูปที่นำมาใช้แก้ปัญหาที่เกิดขึ้นในการเก็บความต้องการสำหรับโดเมนที่กำลังพูดถึง
 - 2) แบบรูปกระบวนการความต้องการ
คือ แบบรูปที่นำมาใช้แก้ปัญหาที่เกิดขึ้นกับเทคนิคหรือกระบวนการ ที่นำมาใช้กับความต้องการที่กำลังพูดถึง
- ความรูปที่ 3 แสดงโครงสร้างและองค์ประกอบของแบบรูปความต้องการ โดยอาศัยโครงสร้างแบบรูปที่เป็นพื้นฐานและใช้กันทั่วไป [2] มาทำการเพิ่มเติมองค์ประกอบตามลักษณะของ แบบรูปทั้ง 2 ประเภท

สำหรับองค์ประกอบของแบบรูปทั้ง 2 ที่ใช้ร่วมกันอธิบายได้ ดังนี้

1) Pattern Name

ชื่อแบบรูปเพื่อใช้เรียก และบอกประเภทของแบบรูป

2) Problem

ปัญหาที่เกิดขึ้น โดยจะนำไปใช้ตัดสินใจเลือกใช้แบบรูป ปัญหาที่เกิดขึ้นจะแสดงลักษณะปัญหาที่สังเกตได้และเกิดขึ้นซ้ำๆกัน

3) Context

สภาวะแวดล้อมที่เคยเกิดปัญหานั้น ซึ่งใช้ประกอบกับปัญหาที่เกิดขึ้น เป็นอีกส่วนที่ช่วยในการพิจารณาใช้แบบรูป แม้ว่าปัญหาที่เกิดขึ้นเหมือนกัน แต่ลักษณะสภาวะแวดล้อมต่างกัน อาจใช้ผลเฉลยแบบเดียวกันไม่ได้

4) Force

ปัจจัยบังคับลักษณะที่เจาะจงลงไป เพื่อช่วยตัดสินใจเลือกแบบรูปมาใช้แก้ปัญหา

5) Result Context

แสดงผลลัพธ์ที่ได้ หรือผลลัพธ์ที่ตามมาหลังจากใช้แบบรูปนี้ แสดงสิ่งที่น่าจะเกิดขึ้นจากการใช้แบบรูป ทั้งทางตรงและทางอ้อม

6) Known Use

แสดงให้เห็นว่านำไปใช้แก้ปัญหาได้จริง และลักษณะการนำเอาไปใช้มีลักษณะอย่างไร การใช้จะนำไปใช้ในกรณีไหนบ้าง

7) Rational

ให้เหตุผลในการสนับสนุนว่า ผลลัพธ์ของแบบรูปนั้นใช้ได้จริง แสดงความสำเร็จหรือผลที่จะได้ซึ่งเกิดขึ้นจริง

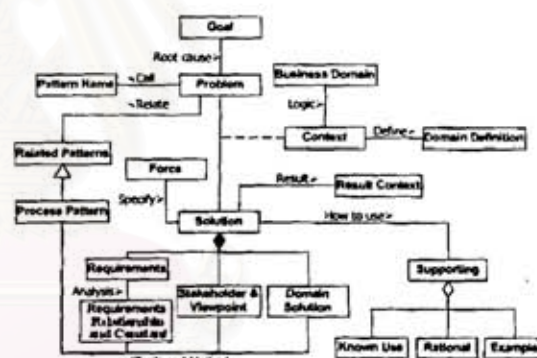
8) Example

แสดงตัวอย่างของการนำไปใช้

9) Related Patterns

แสดงความเกี่ยวข้องกับแบบรูปอื่นๆ

4.2 แบบรูปโดเมนความต้องการ



รูปที่ 4 โครงสร้างแบบรูปโดเมนความต้องการ

แบบรูปโดเมนความต้องการ มีองค์ประกอบที่สัมพันธ์กันซึ่งพิจารณาได้จากรูปที่ 4 โดยมีองค์ประกอบที่เพิ่มเติม เข้ามาจากลักษณะแบบรูปโครงสร้างปกติ ดังนี้

1) Goal

แสดงเป้าหมาย ซึ่งเป็นตัวบ่งบอกความต้องการเพื่อให้เข้าใจมูลเหตุของปัญหาในความต้องการนั้น

2) Domain Definition

เป็นตัวช่วยในการให้ความหมายของระบบ ซึ่งมีความจำเป็นต้องเข้าใจในทางเดียวกัน

3) Business Domain

แสดงลักษณะและการทำงานตามธุรกิจโดเมนที่เป็นอยู่ เพื่อจะได้เห็นภาพและเข้าใจเหตุผลทางธุรกิจ

4) Requirements

ความต้องการซึ่งเป็นส่วนสำคัญและกำหนดระบบในการพัฒนา

5) Domain Solution

คือแนวทางและผลลัพธ์ สำหรับวิศวกรรมความต้องการตามโดเมนที่เป็นปัญหาอยู่ คำถามหรือประเด็นที่ค้นหาคำตอบที่นำไปใช้กับความต้องการ

6) Stakeholders and Viewpoint

วิธีการและมุมมองที่ช่วยให้เข้าใจผลเฉลยมากขึ้น ซึ่งเป็นลักษณะผลเฉลยสำหรับมุมมองหรือบุคคล ที่มีน้ำหนักกัน

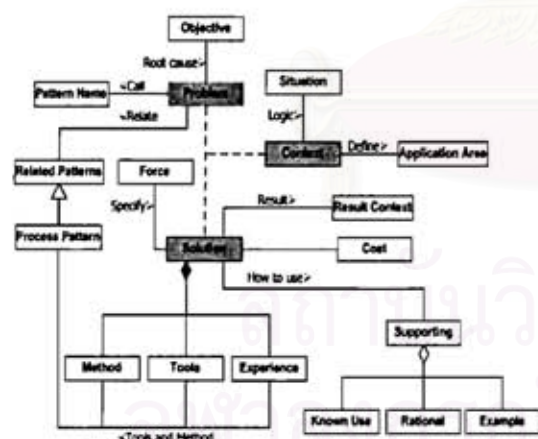
7) Requirements Relationship and Constant

อธิบายแสดงความเกี่ยวข้อง การจัดแย้งหรือซ้ำซ้อนในข้อความความต้องการ การให้ระดับความสำคัญ ความต้องการที่ขึ้นแก่กัน

8) Process Patterns

ส่วนทางอิงไปยังแบบรูปกระบวนการความต้องการ เพื่อที่จะนำวิธีการหรือเครื่องมือที่เหมาะสมมาใช้ได้อย่างมีประสิทธิภาพ

4.3 แบบรูปกระบวนการความต้องการ



รูปที่ 5 โครงสร้างแบบรูปกระบวนการความต้องการ

แบบรูปกระบวนการความต้องการ มีโครงสร้างและความสัมพันธ์ระหว่างองค์ประกอบดังรูปที่ 5 โดยมีองค์ประกอบที่เป็นลักษณะของแบบรูปกระบวนการความต้องการ ดังนี้

1) Situation

เป็นสถานการณ์ ที่ควรนำแบบรูปกระบวนการความต้องการไปใช้

2) Objective

คือจุดประสงค์หลักของกระบวนการ ที่จะสามารถนำไปใช้แก้ปัญหาในลักษณะใดอย่างไร

3) Application Area

อธิบายว่าแบบรูปมีความเหมาะสม ที่จะนำไปใช้กับโดเมนในกลุ่มหรือลักษณะใด

4) Method

วิธีการของกระบวนการ วิธีปฏิบัติที่ทำให้ประสบผลตามจุดประสงค์ที่มี

5) Experience

ลักษณะประสบการณ์ ที่สามารถนำมาใช้ช่วยให้เกิดประสิทธิภาพในกระบวนการที่นำมาปฏิบัติ

6) Tools

เครื่องมือที่มีความเหมาะสม และสามารถนำมาช่วยเพิ่มประสิทธิภาพในการใช้กระบวนการได้

7) Cost

คือค่าใช้จ่ายที่ต้องเสียไป ไม่ว่าจะเป็นเวลาหรือทรัพยากร ซึ่งสามารถนำมาประกอบการตัดสินใจในการเลือกกระบวนการได้ หากเห็นว่าค่าใช้จ่ายที่ไม่เหมาะสม

4.4 ความสัมพันธ์และการนำแบบรูปไปใช้

แบบรูปสำหรับวิศวกรรมความต้องการซอฟต์แวร์ทั้ง 2 ประเภท มีลักษณะแตกต่างกัน แม้ว่าทั้ง 2 จะเป็นอิสระ แต่ยังคงมีความสมบูรณ์ในการนำไปใช้ในตัวเอง และเนื่องจากแบบรูปทั้ง 2 มีความสัมพันธ์กันทั้งในแง่ของการใช้ ซึ่งแบบรูปกระบวนการเป็นเครื่องมือและวิธีการ ที่สนับสนุนแบบรูปโดเมนให้เกิดประสิทธิภาพ และขยายลักษณะองค์ประกอบของผลเฉลยออกไปได้ เมื่อนำแบบรูปทั้ง 2 ไปใช้งานร่วมกันจึงทำให้เกิดประสิทธิภาพมากกว่าการใช้ลักษณะแบบรูปเพียงอย่างเดียวใดอย่างหนึ่ง

5. แบบรูปความต้องการ

ในส่วนนี้จะนำเสนอตัวอย่างแบบรูปความต้องการ ทั้ง 2 ประเภท ซึ่งเป็นความต้องการซอฟต์แวร์ของระบบห้องสมุด

5.1 แบบรูปโดเมนความต้องการ: ยืม-คืน หนังสือ

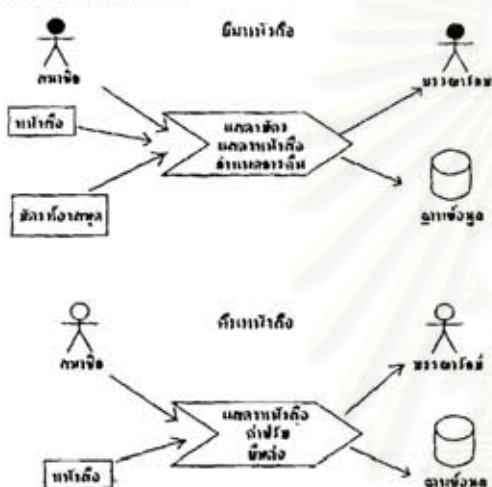
Pattern Name: ยืม-คืน หนังสือ

Problem: ระบบบันทึกและบริหารการยืมคืนหนังสือ ซึ่งการยืมคืนหนังสือจะต้องมีการเก็บข้อมูลผู้ยืม และแก้ไขสถานะของหนังสือในระบบ

Context: ระบบหอสมุดแบบเดี่ยว (ไม่มีการยืมหรือคืนหนังสือข้ามห้องสมุด)

Goal: สามารถให้บริการยืมหรือคืนหนังสือได้ พร้อมทั้งบันทึกข้อมูลและเปลี่ยนสถานะการยืม

Business Domain:



Domain Definition:

บรรณารักษ์ – เจ้าหน้าที่ดูแลห้องสมุดซึ่งเป็นผู้ให้บริการยืม-คืนหนังสือด้วย

สมาชิกห้องสมุด – ผู้มีสิทธิยืมหนังสือ

รหัสหนังสือ – รหัสที่ใช้อ้างอิงหมวดหมู่หรือการจัดประเภทของหนังสือ

หมายเลขหนังสือ – หมายเลขที่ใช้อ้างอิงในฐานข้อมูลหนังสือ หมายเลขจะใช้กับหนังสือหนึ่งเล่มเท่านั้น ซึ่งอาจใช้ร่วมกันกับรหัสหนังสือ

บัตรห้องสมุด – บัตรที่ใช้แสดงความเป็นสมาชิกห้องสมุด และใช้ในการยืม-คืนหนังสือ

Requirements:

- 1) สมาชิกห้องสมุดทำการยืมหนังสือได้
- 2) บรรณารักษ์ทำการตรวจสอบสถานะของผู้ยืมหนังสือ
- 3) ระบบทำการแก้ไขข้อมูลในฐานข้อมูลเมื่อมีการยืมหนังสือ

4) สมาชิกห้องสมุดทำการคืนหนังสือได้

5) ระบบตรวจสอบและคำนวณค่าปรับการยืมหนังสือเกินเวลาเมื่อมีการคืนหนังสือ

6) ระบบทำการแก้ไขฐานข้อมูลหนังสือเมื่อทำการคืนหนังสือ

7) สมาชิกห้องสมุดทำการต่อเวลาการยืมหนังสือได้

Requirements Relationship and Constant:

- 1) จำนวนหนังสือที่จะยืมได้จะต้องไม่เกินจำนวนและประเภทที่ถูกกำหนด
- 2) หนังสือที่ให้อืมต้องไม่ถูกจองหรือเป็นหนังสือที่ห้ามยืม
- 3) การยืมจะต้องทำได้ถ้าหนังสือไม่ถูกจอง
- 4) การยืมจะต้องทำได้ถ้าหนังสือยังไม่เกินเวลายืมปกติ
- 5) เมื่อหนังสือถูกยืม การสืบค้นต้องแสดงว่ามีแต่ถูกยืมอยู่

Stakeholder and Viewpoint:

บรรณารักษ์ – ต้องการเห็นสถานะของหนังสือ และผู้เป็นสมาชิกทั้งหมดบนหน้าจอแสดงผล การที่สมาชิกเข้ามายืมหนังสือ จะต้องขึ้นหนังสือและบัตรสมาชิกห้องสมุดเพื่อใส่ค่าลงระบบ การคืนหนังสือก็มีลักษณะคล้ายกัน บรรณารักษ์ไม่จำเป็นต้องดูแลอะไร และมีข้อความเตือนหรือแจ้งทุกครั้งเกี่ยวกับ การยืมลักษณะที่ยืมไม่ได้ การคืนที่จะต้องเสียค่าปรับ หรือสมาชิกที่มายืมมีปัญหายังไม่คืนหนังสือก็ควรแจ้งด้วย

สมาชิกห้องสมุด – อยากให้งานยืมและคืนทำได้เร็วที่สุด รายละเอียดและกฎในการยืมเป็นสิ่งที่รู้อยู่แล้ว ไม่จำเป็นต้องมีระบบส่วนนี้สนับสนุนก็ได้ แต่อยากได้รายละเอียดที่แสดงว่ายืมเมื่อใดและจะต้องคืนเมื่อไร และไม่ต้องกรอกแบบฟอร์ม หรือรายการใดๆเพื่อทำการยืม-คืนหนังสือ

Domain Solution:

มีอุปกรณ์รับค่าเข้าสู่ระบบเช่นคีย์บอร์ดเมาส์ แท็บเล็ต และทำการเก็บข้อมูลทั้งหนังสือและสมาชิกลงฐานข้อมูลกลางที่ใช้ร่วมกันกับระบบส่วนอื่น มีการออกใบเสร็จค่าปรับและกำหนดเวลาคืนหนังสือซึ่งเป็นไปได้ที่

จะเป็นระบบอัตโนมัติ จุดยึดคินหนังสือเป็นจุดที่จะได้ติดต่อกับผู้ถือบัตร ดังนั้นจุดนี้อาจใช้แจ้งข้อมูลกับผู้ถือบัตรรายบุคคล เช่น มีหนังสือที่ยังไม่คืน

การทำรายการความต้องการจะช่วยตรวจสอบว่าความต้องการมีความครบถ้วนแค่ไหน การวิเคราะห์กลุ่มความต้องการจะทำให้เห็นส่วนที่เกี่ยวข้องกับความต้องการส่วนอื่น การทำต้นแบบก็จะช่วยทดสอบลักษณะหน้างานของเจ้าหน้าที่บรรณารักษ์เพื่อเก็บความต้องการด้านการใช้งานได้

Result Context:

การยืม-คืนหนังสือแบบอัตโนมัติหรือกึ่งอัตโนมัติ ที่มีการนำเครื่องมืออ่านค่าจากบัตรและหนังสือมาใช้ ซึ่งทำให้ระยะเวลาในการยืม-คืนหนังสือสั้นลง และทราบข้อมูลหนังสือและสมาชิกห้องสมุดในขณะยืม-คืนทันที

Known Use:

นำไปใช้กับห้องสมุดที่มีขนาดใหญ่ จะได้ผลดีมากสำหรับห้องสมุดขนาดเล็กอาจไม่มีความจำเป็น หรือได้ประโยชน์ไม่มากนัก

Related Patterns:

การสืบค้นหนังสือ

การจองหนังสือ

การทำหนังสือหาย

Process Patterns:

รายการตรวจสอบความต้องการ

ต้นแบบ (Prototype)

การวิเคราะห์กลุ่มความต้องการ

5.2 แบบรูปกระบวนการความต้องการ: รายการตรวจสอบ

ความต้องการ (Requirements Checklist)

Pattern Name: รายการตรวจสอบความต้องการ

Problem: จะทำอย่างไรเพื่อให้มั่นใจได้ว่าจะสามารถแสดงความต้องการที่มีทั้งหมดได้อย่างครบถ้วน

Objective: ต้องการรายการที่สามารถนำมาใช้ตรวจสอบความต้องการ ที่มีทั้งหมดได้โดยครบถ้วน

Context: มีความต้องการที่ครอบคลุมในหลายๆส่วน ต้องการนำความต้องการ มาทำเป็นรายการ เพื่อตรวจสอบ

Situation:

- 1) ต้องการแสดงความต้องการทั้งหมดที่มี
- 2) ต้องการนำความต้องการ ไปตรวจสอบ
- 3) จะจัดลำดับและให้ความสำคัญกับความต้องการ
- 4) การมีความต้องการในลักษณะกว้าง มีลักษณะแยกย่อยหลายด้าน

Application Area:

- 1) ในระบบที่ต้องการตรวจสอบให้แน่ใจว่าความต้องการครอบคลุมหรือครบหรือไม่
- 2) ในระบบที่ต้องการกลับมาตรวจสอบว่า สิ่งที่พัฒนาเป็นไปตามความต้องการที่มีทั้งหมด
- 3) ระบบที่เน้นกับความต้องการที่เป็นหน้าที่การทำงาน

Method:

- 1) ทำการจดบันทึกความต้องการทั้งหมดที่มีอยู่
- 2) จัดกลุ่มหรือหมวดหมู่ของความต้องการ
- 3) บันทึกรายละเอียดในการลำดับรายการ
- 4) เพิ่มเติม แก้ไข หรือตรวจสอบรายการทุกครั้ง ที่มีการเปลี่ยนแปลงความต้องการ

Experience:

การจัดหมวดหมู่ของความต้องการ จะช่วยให้ลำดับรายการได้ไม่สับสน การลำดับอาจเขียนหรือจดขึ้นมาเป็นส่วนๆ และอาจได้มาจากขั้นตอนการเก็บความต้องการที่มีการบันทึกไว้ การเขียนความต้องการควรเขียนให้ชัดเจน คล้ายกับการเขียนในข้อกำหนดความต้องการ

ควรนำรายการตรวจสอบความต้องการ ไปจัดลำดับความสำคัญ อาจมีการกำหนดรายการตรวจสอบในลักษณะแบบฟอร์มที่มีช่องตรวจสอบ ช่องใส่คำอธิบายเพิ่มเติม และช่องใส่ตัวอย่างความต้องการนั้น

Cost:

ค่าใช้จ่ายและเวลา ในการทำรายการตรวจสอบความต้องการอยู่ในขั้นต่ำ การทำรายการตรวจสอบความต้องการทำได้ง่าย และคุ้มค่า

Result Context:

มีรายการตรวจสอบความต้องการที่สามารถนำมาใช้เป็นตัวตรวจสอบการพัฒนา ระบบ เห็นภาพความต้องการโดยรวมของระบบมากขึ้น และง่ายต่อการจัดลำดับความสำคัญความต้องการ

Related Patterns:

การจัดลำดับความสำคัญของความต้องการ
การตรวจสอบความต้องการ

6. บทสรุป

เราสามารถนำความต้องการกลับมาใช้ใหม่ ในลักษณะของแบบรูปได้ ซึ่งเป็นวิธีนำกลับมาใช้ใหม่ทั้งองค์ความรู้และองค์ประกอบ ซึ่งการนำแบบรูปมาใช้ ช่วยทำให้การนำความต้องการกลับมาใช้ใหม่เป็นแบบแผนและชัดเจนมากขึ้น และยังมีประสิทธิภาพ สามารถนำไปช่วยในกระบวนการพัฒนาซอฟต์แวร์ได้

แบบรูปความต้องการทั้ง 2 ประเภท คือ แบบรูปโดเมนความต้องการ และแบบรูปกระบวนการความต้องการ สามารถนำมาใช้คู่กัน เอื้อประโยชน์ในการปรับใช้กับความต้องการซอฟต์แวร์ของระบบต่างๆ และเสริมการใช้งานของกันและกันได้เป็นอย่างดี

อนึ่งปัจจุบัน มีแบบรูปที่นำไปใช้ในขั้นตอนต่างๆของกระบวนการพัฒนาซอฟต์แวร์ ดังนั้นการใช้แบบรูปการออกแบบร่วมกับแบบรูปอื่นจะทำให้ได้แนวทางในการพัฒนาซอฟต์แวร์ที่สามารถอาศัยความรู้เดิมที่มีอยู่ ทำให้เกิดประสิทธิภาพแก่พัฒนาซอฟต์แวร์ได้เร็วยิ่งขึ้น

7. รายการอ้างอิง

- [1] Ralph R. Young, *Effective Requirements Practices*: Addison-Wesley, 2001.
- [2] E. Gamma, R. Helm, R. Johnson and J. Vlissides, *Design Patterns Elements of Reusable Object-Oriented Software*, 7th ed: Pearson Education, 2002.
- [3] C. Alexander, *A Pattern Language: Towns, Buildings, Construction*: Oxford University Press, 1977.
- [4] K. Beck and W. Cunningham, "Using Pattern Languages for Object-Oriented Programs," presented at Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'87), 1987.
- [5] M. Fowler, *Analysis Pattern: Reuseable Object Models*: Addison-Wesley, 1997.
- [6] James O. Coplien, *Advanced C++ Programming Styles and Idioms*: Addison-Wesley, 1991.
- [7] S. Robertson, "Requirements Patterns Via Events/Use Case," 1996.
- [8] S. Konrad and Betty H.C. Cheng, "Requirements Patterns for Embedded Systems," presented at International Conference on Requirements Engineering, 2002.
- [9] R.S. Wahomo and J. Cheng, "Extensible Requirements Patterns of Web Application for Efficient Web Application Development," presented at First International Symposium on Cyber Worlds, 2002.
- [10] M. Pantoquilha, R. Raminhos and J. Araujo, "Analysis Patterns Specification: Filling the Gaps," presented at Viking Pattern Language of Program (Viking Plop), 2003.
- [11] T. Tsumaki, "Requirements Engineering Patterns Structure," presented at Asia-Pacific Software Engineering Conference, 2004.
- [12] S. Fraser, G. Booch, F. Buschmann, J. Coplien, N. Kerth, I. Jacobson and M.B. Rosson, "Patterns: cult to culture?," presented at the 10th annual conference on Object-oriented programming systems languages and applications (OOPSLA), 1995.
- [13] IEEE, "Guide to the Software Engineering Body of Knowledge (SWEBOK)," 2004.
- [14] G. Kotonya and I. Sommerville, *Requirements Engineering: Processes and Techniques*: John Wiley & Sons, 1997.
- [15] G. Meszaros and J. Doble, "A Pattern Language for Pattern Writing," in *Pattern languages of program design* 3, 1997.

ภาคผนวก ข

เอกสารแม่แบบและเอกสารตัวอย่างของแบบรูปความต้องการ

ข.1 เอกสารอี็กซ์เอ็มแอลสกีมาซึ่งเป็นเอกสารแม่แบบแบบรูปความต้องการ

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://se.cp.eng.chula.ac.th/repattern"
  targetNamespace="http://se.cp.eng.chula.ac.th/repattern"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="2.0">
  <xs:annotation>
    <xs:documentation>
      <![CDATA[
        This XML Schema used for "Requirements Pattern" data
        in my thesis - Design of Requirement Pattern Structure
        for
        Safety-Critical Systems.

        Name: Requirements Pattern Schema
        version: 2.0
        Create Date: 21 March 2007

      ]]>
    </xs:documentation>
  </xs:annotation>
  <xs:complexType name="tCategory">
    <xs:sequence>
      <xs:element name="cat" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="tKeywords">
    <xs:sequence>
      <xs:element name="key" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="tAgreements">
    <xs:sequence>
      <xs:element name="agree" type="xs:integer"/>
      <xs:element name="disagree" type="xs:integer"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="tPatternState">
    <xs:restriction base="xs:NCName">
      <xs:enumeration value="pattern"/>
      <xs:enumeration value="protopattern"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="tForce">
```

```

<xs:sequence>
  <xs:element name="F" minOccurs="0" maxOccurs="unbounded">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="seq" type="xs:integer" use="required"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="tRelatedPatterns">
  <xs:sequence>
    <xs:element name="problem" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="pattern" type="xs:IDREF"/>
          <xs:element name="detail" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tGoal">
  <xs:sequence>
    <xs:element name="G" minOccurs="1" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="detail" type="xs:string"/>
          <xs:element name="eval" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tDomainProblem">
  <xs:sequence>
    <xs:element name="DMP" minOccurs="1" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="problem" type="xs:string"/>
          <xs:element name="type">
            <xs:simpleType>
              <xs:restriction base="xs:NCName">
                <xs:enumeration value="solve"/>
                <xs:enumeration value="improve"/>
                <xs:enumeration value="change"/>
                <xs:enumeration value="initiate"/>
                <xs:enumeration value="define"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="detail" type="xs:string"/>
          <xs:element name="from" minOccurs="1" maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>

```

```

        <xs:element name="who" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="when" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="where" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="event" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="rootcause" type="xs:string"/>
</xs:sequence>
<xs:attribute name="seq" type="xs:integer" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="tDevelopmentProblem">
    <xs:sequence>
        <xs:element name="DEVP" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="problem" type="xs:string"/>
                    <xs:element name="type">
                        <xs:simpleType>
                            <xs:restriction base="xs:NCName">
                                <xs:enumeration value="technology"/>
                                <xs:enumeration value="performance"/>
                                <xs:enumeration value="capacity"/>
                                <xs:enumeration value="cost"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:element>
                    <xs:element name="detail" type="xs:string"/>
                </xs:sequence>
                <xs:attribute name="seq" type="xs:integer" use="required"/>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="tBusinessModel">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="reference">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="ref" type="xs:string"/>
                    <xs:element name="relate" type="xs:string"/>
                    <xs:element name="detail" type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="word_meaning">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="word" type="xs:string"/>
                    <xs:element name="meaning" type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:choice>
</xs:complexType>

```



```

</xs:element>
<xs:element name="diagram">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="description" type="xs:string"/>
      <xs:element name="source" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="seq" type="xs:integer" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="listDetail">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="topic" type="xs:string"/>
      <xs:element name="description" type="xs:string"/>
      <xs:element name="List">
        <xs:complexType>
          <xs:sequence minOccurs="1" maxOccurs="unbounded">
            <xs:element name="lst">
              <xs:complexType>
                <xs:simpleContent>
                  <xs:extension base="xs:string">
                    <xs:attribute name="seq" type="xs:integer"
use="required"/>
                  </xs:extension>
                </xs:simpleContent>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="seq" type="xs:integer" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="topicDetail">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="topic" type="xs:string"/>
      <xs:element name="description" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="seq" type="xs:integer" use="required"/>
  </xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
<xs:complexType name="tEnvironments">
  <xs:sequence>
    <xs:element name="ent" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="type">
            <xs:simpleType>
              <xs:restriction base="xs:NCName">
                <xs:enumeration value="system"/>
                <xs:enumeration value="organize"/>
                <xs:enumeration value="peple"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>

```

```

        </xs:simpleType>
      </xs:element>
      <xs:element name="detail" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="seq" type="xs:integer" use="required"/>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="tSystemConstrain">
  <xs:sequence>
    <xs:element name="const" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="seq" type="xs:integer" use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tRequirements">
  <xs:choice minOccurs="1" maxOccurs="unbounded">
    <xs:element name="functionReq">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="from" type="xs:string"/>
          <xs:element name="detail" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="Req_ID" type="xs:ID" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="non-functionReq">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="from" type="xs:string"/>
          <xs:element name="detail" type="xs:string"/>
          <xs:element name="type">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="product_req"/>
                <xs:enumeration value="process_req"/>
                <xs:enumeration value="performance_req"/>
                <xs:enumeration value="interface_req"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
        </xs:sequence>
        <xs:attribute name="Req_ID" type="xs:ID" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
<xs:complexType name="tDomainSolution">
  <xs:sequence>
    <xs:element name="sol" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>

```

```

        <xs:element name="topic" type="xs:string"/>
        <xs:element name="detail" type="xs:string"/>
        <xs:element name="deal" type="xs:string"/>
        <xs:element name="conclude" type="xs:string"/>
        <xs:element name="reqs" type="xs:IDREFS"/>
    </xs:sequence>
    <xs:attribute name="seq" type="xs:integer" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="tReqRelationshipConstrain">
    <xs:sequence>
        <xs:element name="relationship" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="topic" type="xs:string"/>
                    <xs:element name="req">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="req_rel" maxOccurs="unbounded">
                                    <xs:complexType>
                                        <xs:sequence>
                                            <xs:element name="re_ref" type="xs:IDREF"/>
                                            <xs:element name="priority" type="xs:integer"/>
                                        </xs:sequence>
                                        <xs:attribute name="seq" type="xs:integer"/>
                                    </xs:complexType>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                    <xs:element name="detail" type="xs:string"/>
                </xs:sequence>
                <xs:attribute name="seq" type="xs:integer" use="required"/>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="tStakeholderViewpoint">
    <xs:sequence>
        <xs:element name="stakeholder" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="role" type="xs:string"/>
                    <xs:element name="views">
                        <xs:complexType>
                            <xs:sequence>
                                <xs:element name="req" minOccurs="1"
maxOccurs="unbounded">
                                    <xs:complexType>
                                        <xs:sequence>
                                            <xs:element name="re_ref" type="xs:IDREF"/>
                                            <xs:element name="view" type="xs:string"/>
                                        </xs:sequence>
                                        <xs:attribute name="seq" type="xs:integer"/>
                                    </xs:complexType>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>

```



```

        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="seq" type="xs:integer" use="required"/>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="tDomainExample">
  <xs:choice>
    <xs:element name="diagram" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="name" type="xs:string"/>
          <xs:element name="description" type="xs:string"/>
          <xs:element name="source" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="description" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="topic" type="xs:string"/>
          <xs:element name="description" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="reference" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ref" type="xs:string"/>
          <xs:element name="relate" type="xs:string"/>
          <xs:element name="detail" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="document" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="doc" type="xs:string"/>
          <xs:element name="source" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
<xs:complexType name="tSafetyAcceptance">
  <xs:sequence>
    <xs:element name="acceptance" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="detail" type="xs:string"/>
          <xs:element name="cataloglevel">
            <xs:simpleType>
              <xs:restriction base="xs:NCName">
                <xs:enumeration value="Low"/>
                <xs:enumeration value="Medium"/>

```

```

        <xs:enumeration value="Hight"/>
        <xs:enumeration value="Serious"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="operatelevel" type="xs:string"/>
</xs:sequence>
<xs:attribute name="seq" type="xs:integer" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="tSafetyRequirements">
  <xs:sequence>
    <xs:element name="sre" minOccurs="1" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="type">
            <xs:simpleType>
              <xs:restriction base="xs:NCName">
                <xs:enumeration value="protect"/>
                <xs:enumeration value="detect"/>
                <xs:enumeration value="react"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="detail" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="SR_ID" type="xs:ID" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tSafetySignificant">
  <xs:sequence>
    <xs:element name="ssig" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="sreq" type="xs:string"/>
          <xs:element name="probability">
            <xs:simpleType>
              <xs:restriction base="xs:NCName">
                <xs:enumeration value="frequent"/>
                <xs:enumeration value="probable"/>
                <xs:enumeration value="occasional"/>
                <xs:enumeration value="remote"/>
                <xs:enumeration value="improbable"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="severity">
            <xs:simpleType>
              <xs:restriction base="xs:NCName">
                <xs:enumeration value="catastrophic"/>
                <xs:enumeration value="critical"/>
                <xs:enumeration value="marginal"/>
                <xs:enumeration value="negligible"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

```

        </xs:element>
      </xs:sequence>
      <xs:attribute name="seq" type="xs:integer" use="required"/>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="tAccidentIncident">
  <xs:sequence>
    <xs:element name="ac" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="type">
            <xs:simpleType>
              <xs:restriction base="xs:NCName">
                <xs:enumeration value="hardware"/>
                <xs:enumeration value="software"/>
                <xs:enumeration value="human"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="detail" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tRiskHazard">
  <xs:sequence>
    <xs:element name="rh" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="detail" type="xs:string"/>
          <xs:element name="effect" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tSafetyRelatedExample">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="diagram">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="name" type="xs:string"/>
          <xs:element name="description" type="xs:string"/>
          <xs:element name="source" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="description">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="topic" type="xs:string"/>
          <xs:element name="description" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:choice>

```



```

        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="reference">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ref" type="xs:string"/>
          <xs:element name="relate" type="xs:string"/>
          <xs:element name="detail" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="document">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="doc" type="xs:string"/>
          <xs:element name="source" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
<xs:complexType name="tObjective">
  <xs:sequence>
    <xs:element name="obj" minOccurs="1" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="seq" type="xs:integer" use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tProcessProblem">
  <xs:sequence>
    <xs:element name="PCP" minOccurs="1" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="problem" type="xs:string"/>
          <xs:element name="detail" type="xs:string"/>
          <xs:element name="rootcause" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tSituation">
  <xs:sequence>
    <xs:element name="sit" minOccurs="1" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="seq" type="xs:integer" use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

```

    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tMethod">
  <xs:sequence>
    <xs:element name="process" minOccurs="1" maxOccurs="unbounded">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="seq" type="xs:integer" use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tTools">
  <xs:sequence>
    <xs:element name="tool" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="name" type="xs:string"/>
          <xs:element name="detail" type="xs:string"/>
          <xs:element name="tool_ref" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tCost">
  <xs:sequence minOccurs="1">
    <xs:element name="money" type="xs:string" minOccurs="0"/>
    <xs:element name="peple" type="xs:string" minOccurs="0"/>
    <xs:element name="skill" type="xs:string" minOccurs="0"/>
    <xs:element name="time" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="tProcessExample">
  <xs:choice>
    <xs:element name="diagram" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="name" type="xs:string"/>
          <xs:element name="description" type="xs:string"/>
          <xs:element name="source" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="description" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="topic" type="xs:string"/>
          <xs:element name="description" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="seq" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>

```

```

<xs:element name="reference" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ref" type="xs:string"/>
      <xs:element name="relate" type="xs:string"/>
      <xs:element name="detail" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="document" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="doc" type="xs:string"/>
      <xs:element name="source" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
<xs:complexType name="tBasicPt">
  <xs:sequence>
    <xs:element name="CreateDate" type="xs:date"/>
    <xs:element name="ModifyDate" type="xs:date"/>
    <xs:element name="Author" type="xs:string"/>
    <xs:element name="Category" type="tCategory"/>
    <xs:element name="Keywords" type="tKeywords"/>
    <xs:element name="Comment" type="xs:string"/>
    <xs:element name="Agreements" type="tAgreements"/>
    <xs:element name="PatternState" type="tPatternState"/>
    <xs:element name="PatternName" type="xs:string" nillable="false"/>
    <xs:element name="PatternIntroduction" type="xs:string"
nillable="false"/>
    <xs:element name="Force" type="tForce"/>
    <xs:element name="ResultContext" type="xs:string"/>
    <xs:element name="KnownUse" type="xs:anyType"/>
    <xs:element name="RelatedPatterns" type="tRelatedPatterns"/>
  </xs:sequence>
  <xs:attribute name="Pattern_ID" type="xs:ID" use="required"/>
</xs:complexType>
<xs:complexType name="tDomainPt">
  <xs:complexContent>
    <xs:extension base="tBasicPt">
      <xs:sequence>
        <xs:element name="Goal" type="tGoal"/>
        <xs:element name="DomainProblem" type="tDomainProblem"/>
        <xs:element name="DevelopmentProblem"
type="tDevelopmentProblem"/>
        <xs:element name="BusinessModel" type="tBusinessModel"/>
        <xs:element name="Environments" type="tEnvironments"/>
        <xs:element name="SystemConstrain" type="tSystemConstrain"/>
        <xs:element name="Requirements" type="tRequirements"/>
        <xs:element name="DomainSolution" type="tDomainSolution"/>
        <xs:element name="ReqRelationshipConstrain"
type="tReqRelationshipConstrain"/>
        <xs:element name="StakeholderViewpoint"
type="tStakeholderViewpoint"/>
        <xs:element name="ProcessPatterns" type="tRelatedPatterns"/>
        <xs:element name="DomainExample" type="tDomainExample"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```



```

    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tProcessPt">
  <xs:complexContent>
    <xs:extension base="tBasicPt">
      <xs:sequence>
        <xs:element name="Objective" type="tObjective"/>
        <xs:element name="ProcessProblem" type="tProcessProblem"/>
        <xs:element name="Situation" type="tSituation"/>
        <xs:element name="ApplicationArea" type="xs:string"
nillable="false"/>
        <xs:element name="Method" type="tMethod"/>
        <xs:element name="Tools" type="tTools"/>
        <xs:element name="Cost" type="tCost"/>
        <xs:element name="Experiiece" type="xs:string"/>
        <xs:element name="ProcessExample" type="tProcessExample"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="tSafetyPt">
  <xs:complexContent>
    <xs:extension base="tDomainPt">
      <xs:sequence>
        <xs:element name="SafetyAcceptance" type="tSafetyAcceptance"/>
        <xs:element name="SafetyRequirements"
type="tSafetyRequirements"/>
        <xs:element name="SafetySignificant" type="tSafetySignificant"/>
        <xs:element name="AccidentIncident" type="tAccidentIncident"/>
        <xs:element name="RiskHazard" type="tRiskHazard"/>
        <xs:element name="SafetyRelatedExample"
type="tSafetyRelatedExample"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="RequirementsPatternSystem">
  <xs:complexType>
    <xs:choice minOccurs="1" maxOccurs="unbounded">
      <xs:element name="DomainPattern" type="tDomainPt"/>
      <xs:element name="ProcessPattern" type="tProcessPt"/>
      <xs:element name="SafetyRelatedPattern" type="tSafetyPt"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
</xs:schema>

```

ข.2 ตัวอย่างเอกสารเอ็กซ์เอ็มแอลที่เป็นข้อมูลแบบรูปความต้องการ

```

<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="PatternDocTranslate.xsl"?>
<RequirementsPatternSystem xmlns="http://se.cp.eng.chula.ac.th/repattern"
xsi:schemaLocation="http://se.cp.eng.chula.ac.th/repattern
file:///C:/Documents%20and%20Settings/Media%20Center%20Edition/Desktop/The
sis/pattern%20data/PatternSchema.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DomainPattern Pattern_ID="DPt001">

```

```

<CreateDate>2006-03-21</CreateDate>
<ModifyDate>2006-03-21</ModifyDate>
<Author>เอกชัย แซ่ตั้ง</Author>
<Category>
  <cat>ระบบจำหน่ายคั่ว</cat>
</Category>
<Keywords>
  <key>จำหน่ายคั่ว</key>
  <key>สถานีโดยสาร</key>
</Keywords>
<Comment>สร้างขึ้นเพื่อใช้ในการทดลอง</Comment>
<Agreements>
  <agree>3</agree>
  <disagree>0</disagree>
</Agreements>
<PatternState>pattern</PatternState>
<PatternName>งานจำหน่ายคั่วโดยสารที่สถานีโดยสาร</PatternName>
<PatternIntroduction>เป็นระบบจำหน่ายคั่วโดยสาร โดยเน้นไปที่การจำหน่ายคั่วแก่ผู้ที่จะเดินทางใน
ขณะนั้น และให้ความสำคัญกับการจำหน่ายคั่วแบบล่วงหน้าหรือขายแบบชุดรองลงไป การจำหน่ายคั่วโดยสารจะให้
ความสำคัญในแง่ของประสิทธิภาพเพื่อให้ทันต่อความต้องการในการเดินทางในช่วงเวลาหนึ่งๆ ทั้งนี้ข้อมูลการจำหน่าย
คั่วโดยสารยังต้องทำการบันทึกไว้ในระบบเนื่องจากมีความสำคัญเพื่อใช้ในการตรวจสอบยอดจำหน่ายคั่วโดยสาร และ
ยังสามารถนำไปใช้เพื่อการวิเคราะห์ข้อมูลในด้านต่างๆได้อีกด้วย</PatternIntroduction>
<Force>
  <F seq="1">คั่วโดยสารทั่วไปมีราคาตามระยะทางการเดินทาง</F>
  <F seq="2">รูปแบบของคั่วมีหลายลักษณะ</F>
</Force>
<ResultContext>ระบบจำหน่ายคั่วที่รองรับรูปแบบของคั่วได้หลายรูปแบบ โดยจำกัดเป็นกลุ่มประเภทของ
คั่วที่จะมีไว้ในระบบ ระบบจำหน่ายคั่วที่หน้าสถานีโดยสารโดยใช้เครื่องจำหน่ายคั่วแบบอัตโนมัติร่วมด้วย ฐานข้อมูล
จากการจำหน่ายคั่วที่นำไปใช้สรุปรายงาน</ResultContext>
<KnownUse>เอกชัย</KnownUse>
<RelatedPatterns>
  <problem seq="1">
    <pattern>p1</pattern>
    <detail>p1-detail</detail>
  </problem>
</RelatedPatterns>
<Goal>
  <G seq="1">
    <detail>การจำหน่ายคั่วในแต่ละวันจำเป็นต้องมีการบันทึกข้อมูลเพื่อสามารถใช้อย่างอิงได้ว่ามียอดใน
การจำหน่ายคั่วโดยสารเท่าไร มีการจำหน่ายไปในแบบใดบ้าง ตลอดจนต้องบันทึกเจ้าหน้าที่ที่รับผิดชอบในการ
จำหน่ายคั่วแต่ละส่วนเอาไว้ โดยจะต้องสามารถนำข้อมูลมาสรุปเป็นรายงานยอดจำหน่ายคั่วได้</detail>
    <eval>มีการบันทึกการขายเก็บไว้ และสรุปเป็นรายงานรายวัน รายเดือนได้</eval>
  </G>
  <G seq="2">

```

<detail>การจำหน่ายตัวโดยเฉพาะในช่วงเวลาเร่งด่วนจะต้องสามารถจำหน่ายได้อย่างรวดเร็ว และไม่
ควรจะต้องใช้เวลานาน การจำหน่ายตัวไม่ว่าจะเป็นประเภทใดจะต้องไม่ยุ่งยากในการใช้ การใช้งานระบบก็ควรจะช่วย
ให้ทำงานงานจำหน่ายตัวโดยสารมีความคิดพลาดคลงได้</detail>

<eval>จำหน่ายตัวได้อย่างมีประสิทธิภาพ ถูกต้อง และรวดเร็ว</eval>

</G>

</Goal>

<DomainProblem>

<DMP seq="1">

<problem>จำหน่ายตัวในลักษณะที่แตกต่างกันหลายรูปแบบ</problem>

<type>improve</type>

<detail>ต้องการให้ตัวที่จำหน่ายมีหลากหลายรูปแบบตามพฤติกรรมการใช้ของผู้โดยสาร การขายต้อง
สนับสนุนรูปแบบทั้งหมดและรูปแบบที่อาจมีเพิ่มเติมในอนาคต</detail>

<from>

<who>ฝ่ายวางแผนกลยุทธ์</who>

<event>ต้องพัฒนาระบบให้สนับสนุนรูปแบบการจำหน่ายตัวโดยสารแบบใหม่ๆ</event>

</from>

<rootcause>รูปแบบของตัวโดยสารมีไม่จำกัด</rootcause>

</DMP>

<DMP seq="2">

<problem>การจำหน่ายตัวโดยสารให้รวดเร็วทันความต้องการ</problem>

<type>improve</type>

<detail>ในช่วงเวลาเร่งด่วนจำนวนผู้โดยสารมีจำนวนมากส่วนที่จำหน่ายตัวนั้นมืออยู่อย่างจำกัด ทำให้
การจำหน่ายตัวทำได้ไม่ทันกับความต้องการ</detail>

<from>

<who>ผู้โดยสาร</who>

<where>สถานีโดยสารที่เป็นจุดเชื่อมต่อกับระบบขนส่งขนาดใหญ่อื่นๆ จะทำให้จำนวนผู้โดยสาร
ที่มาต่อรถไฟมีมากเป็นช่วงๆ</where>

</from>

<rootcause>จำนวนผู้โดยสารมีมากและไม่สม่ำเสมอ</rootcause>

</DMP>

</DomainProblem>

<DevelopmentProblem>

<DEVP seq="1">

<problem>จะทำการใช้เครื่องจำหน่ายตัวอัตโนมัติ</problem>

<type>capacity</type>

<detail>การนำเครื่องจำหน่ายตัวแบบอัตโนมัติมาใช้จะช่วยให้ผู้โดยสารสามารถซื้อตัวได้ด้วยตัวเอง
แต่เครื่องจำหน่ายตัวอัตโนมัติที่นำมาใช้อาจไม่เหมาะสมกับผู้โดยสารทุกกลุ่ม ตัวบางประเภทยุ่งยากและไม่เหมาะสมใน
การขายด้วยเครื่องอัตโนมัติ การจำหน่ายตัวด้วยเครื่องอัตโนมัติจึงจะต้องพัฒนาซอฟต์แวร์ขึ้นมาเป็นพิเศษให้รองรับ
ลักษณะการทำงานที่เฉพาะเจาะจงได้ ซึ่งจัดได้ว่าเป็นระบบที่คุ้นเคยเฉพาะผู้ที่เคยพัฒนาซอฟต์แวร์สำหรับระบบฝังตัว
ลักษณะนี้มาก่อนเท่านั้น ดังนั้นทีมพัฒนาจำเป็นต้องศึกษาลักษณะการทำงานของเครื่องจำหน่ายตัวอัตโนมัติ ตลอดจน
ต้องศึกษาคำสั่งต่างๆที่ใช้ในการพัฒนาซอฟต์แวร์สำหรับควบคุมเครื่องซึ่งรวมไปถึงลักษณะการทำงานที่เป็นส่วนต่อ
ประสานผู้ใช้ด้วย</detail>


```

</DEVP>
<DEVP seq="2">
  <problem>ไม่สามารถกำหนดประเภทของตัวโดยสารที่จะมีในอนาคต</problem>
  <type>cost</type>
  <detail>รูปแบบของตัวโดยสารเป็นปัจจัยที่ทำให้ระบบในการจำหน่ายตัวมีความซับซ้อนและเกิดการ
ทำงานที่ยุ่งยาก ซึ่งจากลักษณะของระบบทำให้ไม่สามารถกำหนดประเภทของตัวและจำกัดลักษณะได้อย่างชัดเจน การ
จะพัฒนาระบบให้สามารถสนับสนุนการจำหน่ายตัวได้ทุกประเภทจึงทำได้ยาก อาจมีค่าใช้จ่ายในการพัฒนาสูงกว่าปกติ
</detail>
</DEVP>
</DevelopmentProblem>
<BusinessModel>
  <listDetail seq="1">
    <topic>รูปแบบของตัวโดยสารที่มีการจำหน่าย</topic>
    <description>รูปแบบตัวโดยสารสำหรับการใช้ในการโดยสารในระบบรถไฟมีรูปแบบที่สามารถจำแนก
ดังที่แสดง แต่สำหรับการนำมาใช้จริงอาจไม่มีความจำเป็นต้องจำหน่ายตัวโดยสารในทุกรูปแบบ</description>
    <List>
      <lst seq="1">ตัวเที่ยวเดียว (ไปอย่างเดียว)</lst>
      <lst seq="2">ตัวไปกลับ (ไปกลับตามสถานีโดยสารที่กำหนด)</lst>
      <lst seq="3">ตัวรายวันไม่จำกัด (ใช้ไม่จำกัดใน 1 วัน)</lst>
      <lst seq="4">ตัวรายวันจำกัด (ใช้ได้หลายครั้งใน 1 วัน แต่ใช้ได้ไม่เกินเที่ยวที่
กำหนด)</lst>
      <lst seq="5">ตัวรายสัปดาห์ รายเดือน ฯลฯ (ตามแต่ละกำหนดช่วงอายุของตัว)</lst>
      <lst seq="6">ตัวตามจำนวนเที่ยว</lst>
      <lst seq="7">ตัวแบบเติมเงินหรือเติมเที่ยวได้</lst>
    </List>
  </listDetail>
  <topicDetail>
    <topic>ลักษณะการซื้อตัวจากเครื่องจำหน่ายอัตโนมัติ</topic>
    <description>การซื้อจากเครื่องจำหน่ายตัวอัตโนมัติจะเริ่มจากการเลือกสถานีโดยสาร แล้วจำนวน
ค่าโดยสารจะแจ้งให้ทราบ ในบางกรณีจะมีให้ระบุจำนวนตัวที่ซื้อด้วยว่ากี่ใบ รูปแบบการชำระค่าโดยสารของเครื่อง
จำหน่ายตัวจะขึ้นอยู่กับคุณสมบัติของเครื่อง ซึ่งอาจรวมไปถึงการชำระค่าโดยสารผ่านบัตรเครดิตด้วย
</description>
  </topicDetail>
  <topicDetail>
    <topic>ลักษณะการซื้อตัวจากพนักงาน</topic>
    <description>การซื้อตัวโดยสารจากพนักงานจำหน่ายตัว จะขึ้นอยู่กับผู้โดยสารว่าจะแจ้งซื้อตัวใน
ลักษณะใด หรือเป็นหน้าที่ของพนักงานจำหน่ายตัวที่จะช่วยให้คำแนะนำในการซื้อตัว อย่างไรก็ตามก็ตัวโดยสารบาง
ประเภทจะไม่มีเครื่องขายตัวโดยสารไว้ก่อน เช่นตัวแบบเติมเงิน ตัวแบบเที่ยวเดียว ซึ่งจะเป็นการใช้ระบบออกตั๋วแบบ
ทันทีในขณะที่ทำการซื้อตัวผ่านระบบคอมพิวเตอร์</description>
  </topicDetail>
</BusinessModel>
<Environments>
  <ent seq="1">

```

```

    <type>organize</type>
    <detail>หน่วยงานที่มีหน้าที่ดูแลระบบขนส่งมวลชนจะเป็นผู้กำหนดกรอบของราคา และส่งผลถึง
ประเภทของบัตรที่จะมีการนำมาใช้ ซึ่งอาจกำหนดเป็นบัตรที่ใช้ร่วมกับระบบขนส่งอื่นๆด้วย</detail>
  </ent>
  <ent seq="2">
    <type>peple</type>
    <detail>บุคคลบางกลุ่มจะมีสิทธิพิเศษในการซื้อตั๋วโดยสาร เช่น กลุ่มนักเรียนนักศึกษา กลุ่มผู้สูงอายุ
เป็นต้น</detail>
  </ent>
</Environments>
<SystemConstrain />
<Requirements>
  <functionReq Req_ID="F1">
    <from>ผู้จัดการฝ่ายขาย</from>
    <detail>จะต้องสามารถใช้ในการจำหน่ายตั๋วได้</detail>
  </functionReq>
  <functionReq Req_ID="F2">
    <from>ผู้จัดการฝ่ายขาย</from>
    <detail>จะต้องสนับสนุนการจำหน่ายตั๋วหลายประเภท</detail>
  </functionReq>
  <functionReq Req_ID="F3">
    <from>ผู้จัดการฝ่ายการตลาด</from>
    <detail>จะต้องมีการบันทึกข้อมูลการขายเก็บไว้เสมอ</detail>
  </functionReq>
  <functionReq Req_ID="F4">
    <from>ผู้จัดการฝ่ายขาย</from>
    <detail>จะต้องใช้ในการสรุปยอดจำหน่ายตั๋วได้</detail>
  </functionReq>
  <non-functionReq Req_ID="NF1">
    <from>ผู้จัดการฝ่ายขาย</from>
    <detail>ข้อมูลจากการจำหน่ายตั๋วต้องมีความถูกต้อง</detail>
    <type>product_req</type>
  </non-functionReq>
  <non-functionReq Req_ID="NF2">
    <from>ผู้จัดการฝ่ายการตลาด</from>
    <detail>ข้อมูลการจำหน่ายตั๋วต้องสามารถเชื่อมโยงกับฐานข้อมูลเพื่อใช้ในการวิเคราะห์ได้
</detail>
    <type>interface_req</type>
  </non-functionReq>
  <non-functionReq Req_ID="NF3">
    <from>ผู้จัดการฝ่ายขาย</from>
    <detail>การจำหน่ายตั๋วต้องทำได้ทันกับความต้องการซื้อตั๋ว</detail>
    <type>performance_req</type>
  </non-functionReq>
  <non-functionReq Req_ID="NF4">
    <from>ผู้จัดการฝ่ายขาย</from>

```

```

<detail>จะต้องรองรับจำนวนผู้โดยสารที่มาใช้งานได้อย่างเพียงพอ</detail>
<type>product_req</type>
</non-functionReq>
</Requirements>
<DomainSolution>
<sol seq="1">
<topic>จำนวนตัวมีหลายรูปแบบ</topic>
<detail>จากปัญหาที่ตัวโดยสารมีด้วยกันหลายประเภท และอาจมีการปรับเปลี่ยนหรือเพิ่มรูปแบบตัว
โดยสารสำหรับอนาคตได้ ทำให้เกิดปัญหาคงมาในหลายลักษณะ</detail>
<deal>จำกัดรูปแบบที่เป็นไปได้ โดยจัดกลุ่มตามลักษณะคุณสมบัติของตัวโดยสาร ซึ่งหากบัตรประเภท
ใดมีลักษณะการใช้งานคล้ายกันก็จะจัดให้อยู่ในกลุ่มเดียวกัน โดยชนิดของตัวโดยสารจะยอมให้มีได้ไม่จำกัดแต่จะต้อง
อยู่ภายใต้กลุ่มของรูปแบบตัวโดยสารที่กำหนดไว้แต่แรกเท่านั้น</deal>
<conclude>จำกัดกลุ่มของตัวโดยสารซึ่งแยกประเภทตามคุณสมบัติของตัวโดยสารที่คล้ายกัน
</conclude>
<reqs>F2</reqs>
</sol>
<sol seq="2">
<topic>ผู้โดยสารมีจำนวนมาก งานจำหน่ายตั๋วอาจไม่เพียงพอและทันต่อความต้องการ</topic>
<detail>กรณีที่บางช่วงเวลาและบางเหตุการณ์ทำให้ผู้ที่มาซื้อตั๋วมีจำนวนมาก และเกรงว่างานจำหน่าย
ตั๋วโดยสารอาจไม่เพียงพอต่อความต้องการ คงไม่สามารถหาวิธีการที่มีประสิทธิภาพร้อยเปอร์เซ็นต์มาแก้ปัญหาใน
ลักษณะนี้ แต่ก็น่าที่จะประมาณการ หรือมีแผนหรือระบบรองรับลักษณะที่จะเกิดขึ้น</detail>
<deal>เป็นไปได้ที่จะใช้ simulation มาช่วยในการประเมิน รวมทั้งควรเอาสถิติผู้เข้ามาซื้อตั๋วแต่
แต่ละลักษณะมาพิจารณา ซึ่งการการจะพัฒนาระบบให้สามารถรองรับการทำงานเต็มร้อยเปอร์เซ็นต์คงไม่สามารถ
เป็นไปได้ แต่ระดับที่น่าจะยอมรับได้ซึ่งควรใช้กำหนดเป็นคุณสมบัติของงานจำหน่ายตั๋วคือ 60-70% ของผู้มาซื้อตั๋ว
โดยสารจะต้องใช้เวลาไม่เกิน 5 นาที และอย่างช้าที่สุดที่ผู้โดยสารจะใช้เวลาในการซื้อตั๋วคือ 20 นาที โดยเพิ่ม
ปริมาณเครื่องจำหน่ายตั๋วและอัตโนมัติและพนักงานจำหน่ายตั๋วเป็นการแก้ปัญหา และอาจอาศัยลักษณะตัวที่มีด้วยกัน
หลายประเภทมาช่วยลดเวลาที่จะต้องซื้อตั๋วได้ เช่น ตัวแบบรายวัน รายเดือน</deal>
<conclude>นำการ simulation มาช่วย เพื่อหาวิธีให้สามารถจัดจำหน่ายตั๋วโดยสารได้อย่าง
เพียงพอ ด้วยการประเมินผลที่ได้</conclude>
<reqs>NF4</reqs>
</sol>
<sol seq="3">
<topic>การบันทึกข้อมูลการจำหน่ายตั๋ว</topic>
<detail>ข้อมูลที่ได้จากการจำหน่ายตั๋วจำเป็นต้องมีการบันทึกเก็บเอาไว้ แต่เนื่องจากรูปแบบของการ
จำหน่ายตั๋วมีหลายลักษณะไม่ว่าจะเป็นการขายโดยปกติ การขายด้วยเครื่องจำหน่ายตั๋วอัตโนมัติ รวมทั้งตัวที่มีจัด
จำหน่ายนอกเหนือจากในสถานีโดยสาร ข้อมูลที่จะต้องบันทึกนอกจากจำนวนเงินในการจำหน่ายตั๋วจำนวนตั๋วที่ขาย
อาจจะต้องทำการบันทึกข้อมูลอื่นๆเพิ่มเติมอีก เช่น พนักงานจำหน่ายตั๋ว ช่วงเวลาที่มีการจำหน่ายตั๋ว</detail>
<deal>การบันทึกข้อมูลควรบันทึกเท่าที่จำเป็นเท่านั้น ไม่เช่นนั้นข้อมูลที่จะจัดเก็บจะมีจำนวนมากเกินไป
เช่นการเก็บบันทึกเวลาที่ทำการจำหน่ายตั๋วซึ่งไม่มีความจำเป็นเพียงพอ ซึ่งหนทางที่ดีและน่าจะยอมรับได้ คือการเก็บ

```


ข้อมูลโดยแจกแจงตามประเภทของรูปแบบการจำหน่าย โดยบันทึกผลลัพธ์ที่ถึงผลสรุป เช่น จำนวนตัวโดยสารที่จำหน่ายผ่านเครื่องจำหน่ายตั๋วอัตโนมัติในช่วงเวลา 8.00 - 8.30น. เป็นต้น</deal>

<conclude>ทำการบันทึกข้อมูลเพียงบางส่วน จะไม่บันทึกข้อมูลทั้งหมดโดยละเอียด แต่ก็ยังคงแจกแจงข้อมูลที่มีที่มาต่างกันถือเป็นข้อมูลต่างชุดกัน</conclude>

```
<reqs>F3</reqs>
</sol>
</DomainSolution>
<ReqRelationshipConstrain>
  <relationship seq="1">
    <topic>การรองรับผู้โดยสารที่มีความหลากหลายเป็นจำนวนมาก</topic>
    <req>
      <req_rel>
        <re_ref>NF4</re_ref>
        <piority>1</piority>
      </req_rel>
      <req_rel>
        <re_ref>F2</re_ref>
        <piority>2</piority>
      </req_rel>
    </req>
    <detail>การจำหน่ายตั๋วจำเป็นต้องทันต่อความต้องการ แม้ว่าตั๋วอาจมีหลายลักษณะซึ่งการขายตั๋วแต่ละรูปแบบอาจมีความแตกต่างกัน ระบบจำหน่ายตั๋วก็ต้องยึดถือและสนับสนุนการจำหน่ายตั๋วที่มีนัยของจำนวนตั๋วที่ขายเป็นสำคัญ</detail>
```

```
</relationship>
<relationship seq="2">
  <topic>การบันทึกข้อมูลและสรุปรายงาน</topic>
  <req>
    <req_rel>
      <re_ref>F3</re_ref>
      <piority>1</piority>
    </req_rel>
    <req_rel>
      <re_ref>F4</re_ref>
      <piority>1</piority>
    </req_rel>
  </req>
  <detail>ข้อมูลที่ทำกรบันทึกและรายงานสรุปผลต้องมีความสอดคล้องกัน ซึ่งความต้องการเหล่านี้ต้องพิจารณาไปคู่กัน เพื่อสามารถเก็บข้อมูลได้สอดคล้องกับข้อมูลที่จะนำมาเป็นรายงาน และเมื่อพิจารณาจากรายงานก็จะช่วยให้เห็นข้อมูลที่จะต้องจัดเก็บ</detail>
```

```
</relationship>
</ReqRelationshipConstrain>
<StakeholderViewpoint>
  <stakeholder seq="1">
    <role>ผู้โดยสาร</role>
    <views>
      <req>
        <re_ref>F1</re_ref>
        <view>ไม่ต้องการที่จะซื้อตั๋วซื้อตั๋วนาน และหากเป็นไปได้ก็ต้องการใบเสร็จในการซื้อตั๋วโดยสารด้วย</view>
```

```

    </req>
    <req>
      <re_ref>F2</re_ref>
      <view>การซื้อตั๋วโดยสารทุกประเภทน่าจะทำได้ผ่านจุดจำหน่ายตั๋วจุดเดียวกัน</view>
    </req>
  </views>
</stakeholder>
<stakeholder seq="2">
  <role>พนักงานจำหน่ายตั๋วที่สถานีโดยสาร</role>
  <views>
    <req>
      <re_ref>F1</re_ref>
      <view>ระบบในการจำหน่ายตั๋วควรจะใช้งานง่าย และควรสามารถแก้ปัญหาเมื่อมีการยกเลิกการ
ขายที่ผิดพลาดหรือคืนตั๋วได้</view>
    </req>
    <req>
      <re_ref>F2</re_ref>
      <view>การที่มีตั๋วโดยสารหลายประเภท ระบบควรจะนำมาช่วยในจุดนี้เพื่อลดความสับสนในการ
จำหน่ายตั๋วได้ และไม่ว่าจะเป็นการจำหน่ายตั๋วแบบใดการใช้งานก็ไม่ควรจะแตกต่างกันมากนัก</view>
    </req>
    <req>
      <re_ref>F3</re_ref>
      <view>ในการบันทึกข้อมูลจากการจำหน่ายตั๋ว การบันทึกและผลของการบันทึกน่าจะแสดงออกมา
ให้ทราบด้วย อาจแสดงสรุปยอดขายจากข้อมูลที่บันทึกระหว่างการจำหน่ายตั๋วด้วย</view>
    </req>
  </views>
</stakeholder>
<stakeholder seq="3">
  <role>เจ้าหน้าที่วางแผนระบบขนส่ง</role>
  <views>
    <req>
      <re_ref>F3</re_ref>
      <view>การบันทึกข้อมูลต้องคงความถูกต้อง หากระบบมีปัญหาข้อมูลที่มีการบันทึกควรจะคง
อยู่</view>
    </req>
    <req>
      <re_ref>NF4</re_ref>
      <view>ระบบจะต้องรองรับผู้โดยสารในช่วงปกติได้โดยไม่มีปัญหา และในช่วงที่มีผู้โดยสารมาก
เป็นพิเศษการทำงานของระบบจะต้องเพียงพอ และควรมีการล่าช้าได้เล็กน้อยเท่านั้น</view>
    </req>
    <req>
      <re_ref>F4</re_ref>
      <view>การสรุปยอดการจำหน่ายตั๋วจะต้องเป็นไปตามมุมมองที่สามารถนำมาใช้วิเคราะห์ระบบ
ขนส่งได้ โดยข้อมูลที่ได้นี้ต้องมีความถูกต้องครบถ้วนตามที่แจ้งรูปแบบรายงาน</view>
    </req>
  </views>
</stakeholder>
</StakeholderViewpoint>
<ProcessPatterns />

```

```

    <DomainExample />
  </DomainPattern>
  <ProcessPattern PatternID="Ppt001">
    <CreateDate>2006-03-21</CreateDate>
    <ModifyDate>2006-03-21</ModifyDate>
    <Author>เอกชัย แซ่ตั้ง</Author>
    <Category>
      <cat>elicitation</cat>
    </Category>
    <Keywords>
      <key>define environment</key>
      <key>กำหนดสภาพแวดล้อม</key>
    </Keywords>
    <Comment>สร้างขึ้นเพื่อใช้ในการทดลอง</Comment>
    <Agreements>
      <agree>3</agree>
      <disagree>0</disagree>
    </Agreements>
    <PatternState>pattern</PatternState>
    <PatternName>Define Environments</PatternName>
    <PatternIntroduction />
    <Force>
      <F seq="1">ไม่มีการกำหนดระบบหรือสภาพแวดล้อมที่เกี่ยวข้องกับระบบอย่างแท้จริง</F>
      <F seq="2">มีความเป็นไปได้ว่าจะมีการเปลี่ยนแปลงในอนาคต</F>
    </Force>
    <ResultContext>ระบบที่พัฒนามีความชัดเจนมากขึ้นในขอบเขตและข้อจำกัดในการพัฒนา สามารถ
ประเมินความต้องการคุณลักษณะสภาพแวดล้อมที่กำหนดขึ้น มีความชัดเจนและที่นำไปในเรื่องของผู้กำหนดและ
จุดประสงค์ของสภาพแวดล้อมของระบบ ระบบที่สร้างขึ้นตามสภาพแวดล้อมที่กำหนดจะสามารถทำงานเข้ากันได้กับส่วน
อื่นๆหรือระบบอื่นๆที่เกี่ยวข้อง</ResultContext>
    <KnownUse />
    <RelatedPatterns />
    <Objective>
      <obj seq="1">เพื่อกำหนดสภาพแวดล้อม ซึ่งยังไม่ได้มีการกำหนดไว้</obj>
      <obj seq="2">เพื่อลดความเปลี่ยนแปลงของคุณลักษณะของสภาพแวดล้อม</obj>
      <obj seq="3">เพื่อกำหนดสภาพแวดล้อมที่เหมาะสมในการพัฒนา</obj>
    </Objective>
    <ProcessProblem>
      <PCP seq="1">
        <problem>จะกำหนดสภาพแวดล้อมได้อย่างไร</problem>
        <detail>เนื่องจากสภาพแวดล้อมอาจไม่ได้มีการกำหนดไว้เมื่อเริ่มต้นการพัฒนา การกำหนดสภาพ
        แวดล้อมของระบบจึงตกเป็นหน้าที่ของผู้พัฒนาในการเก็บรายละเอียดของความต้องการเพื่อกำหนดเป็นสภาพแวดล้อมที่
        เหมาะสม ซึ่งการที่จะกำหนดสภาพแวดล้อมขึ้นมานั้นอาจได้รับสิทธิ์เต็มที่จากผู้สนับสนุนโครงการซึ่งก็จะเป็นเรื่องง่าย
        ของทีมพัฒนา แต่หากบางส่วนผู้สนับสนุนโครงการหรือผู้ใช้งานมีการกำหนดสภาพแวดล้อมบางอย่างเอาไว้ การจะ
        สรุปรายละเอียดของสภาพแวดล้อมสำหรับระบบจึงเป็นปัญหาที่กระทบไปยังส่วนอื่นๆตลอดการพัฒนา
        </detail>
      </PCP>
    </ProcessProblem>
  </ProcessPattern>

```



```

    <rootcause>ยังไม่มี ความชัดเจน ในการกำหนดสภาพแวดล้อม และอาจไม่ได้รับสิทธิและอำนาจเต็ม
ในการตัดสินใจเลือกสภาพแวดล้อม</rootcause>
  </PCP>
  <PCP seq="2">
    <problem>จะต้องพิจารณาสภาพแวดล้อมในด้านใดบ้าง</problem>
    <detail>การพัฒนา ระบบจะมี ส่วนของสภาพแวดล้อมที่มีส่วนที่เกี่ยวข้อง ในด้านต่างๆจำนวนมาก ทำให้เกิด
ปัญหาว่า มีจุดใดบ้างของสภาพแวดล้อมที่ควร จะกำหนดเพื่อนำมาใช้ การเก็บความต้องการ อาจไม่ได้รับ
รายละเอียดที่ครบถ้วน หากผู้ให้รายละเอียด ไม่ได้พิจารณาครอบคลุมปัจจัยของสภาพแวดล้อมทั้งหมดที่มีผลเกี่ยวเนื่อง
</detail>
    <rootcause>ไม่สามารถ แจกแจงให้ครอบคลุมสภาพแวดล้อมที่จะทำการกำหนดได้</rootcause>
  </PCP>
  <PCP seq="3">
    <problem>จะต้องกำหนดสภาพแวดล้อมให้ครอบคลุมระดับไหน เพื่อประโยชน์ในแง่ใดบ้าง
  </problem>
    <detail>การกำหนดสภาพแวดล้อมมักจะพิจารณาเพียงแค่เหตุการณ์เฉพาะหน้า และช่วงสั้นของการ
ดำเนินการพัฒนา ระบบ ทำให้การกำหนดสภาพแวดล้อมมีปัญหาหลังจากผ่านไป ในระยะเวลาหนึ่ง การกำหนด
สภาพแวดล้อมในหลายๆครั้งก็ ไม่ก่อให้เกิดประโยชน์ต่อการพัฒนา บ่อยครั้งที่การกำหนดสภาพแวดล้อม ไม่ได้สังเกตเห็น
ผลประโยชน์ในการนำไปใช้ ขอบเขตที่กำหนดก็มักจะครอบคลุมแค่สิ่งที่ผู้เก็บความต้องการนั้นมองเห็นเท่านั้น เช่น การ
ละเลยการกำหนดสภาพแวดล้อมเพื่อให้เอื้อไปถึงการพัฒนาและทดสอบระบบด้วย</detail>
    <rootcause>ขาดมุมมองขนาดใหญ่ ในการกำหนดสภาพแวดล้อมเพื่ออำนวยความสะดวกไปยังส่วนอื่นๆ
ของการพัฒนาซอฟต์แวร์</rootcause>
  </PCP>
</ProcessProblem>
<Situation>
  <sit seq="1">ระบบไม่มีการกำหนดสภาพแวดล้อมที่ชัดเจนในเบื้องต้น</sit>
  <sit seq="2">ระบบมีความเป็น ไปได้ที่จะมีความเปลี่ยนแปลงเกิดขึ้นในอนาคต</sit>
  <sit seq="3">ไม่ทราบรายละเอียดของระบบหรือส่วนประกอบอื่นๆที่ระบบจะต้องทำงานร่วมด้วย
</sit>
</Situation>
  <ApplicationArea>ระบบทั่วไปที่สามารถพัฒนาในสภาพแวดล้อมที่มีให้เลือกหรือเป็นไป ได้หลายรูปแบบ
</ApplicationArea>
  <Method>
    <process seq="1">กำหนดคุณลักษณะของฮาร์ดแวร์</process>
    <process seq="2">กำหนด operating System</process>
    <process seq="3">กำหนด Library ของ software</process>
    <process seq="4">กำหนดเวอร์ชันของ OS และ Library</process>
    <process seq="5">ตรวจสอบความเข้ากันได้และความเหมาะสมของ platform
(Hardware+OS+Library)</process>
    <process seq="6">พิจารณาและเลือก interface ที่เกี่ยวข้อง เช่น database driver,
ERP, Protocol</process>
    <process seq="7">จัดแจง interface ให้สามารถทำงานประสานกันได้</process>

```

<process seq="8">กำหนด software ที่จะมีการใช้งานเป็นพื้นฐานของระบบ เช่น ระบบต้องมีซอฟต์แวร์อ่านเอกสาร .pdf</process>

<process seq="9">กำหนดเวอร์ชันของซอฟต์แวร์ที่นำมาใช้ให้เหมาะสมกับระบบ</process>

<process seq="10">ตรวจสอบข้อมูลระบบส่วนอื่นๆที่เกี่ยวข้องกับการทำงาน</process>

<process seq="11">กำหนดสภาพการทำงาน (ในระบบที่เกี่ยวข้องกับลักษณะการทำงาน

ด้วย)</process>

</Method>

<Tools />

<Cost>

<money>มีค่าใช้จ่ายน้อย</money>

<peple>ต้องใช้ทีมงานที่วางโครงสร้างสถาปัตยกรรมระบบเป็นผู้กำหนด</peple>

<skill>ใช้ความสามารถในการวางโครงสร้างสถาปัตยกรรมระบบ</skill>

<time>ใช้เวลาไม่มากนัก</time>

</Cost>

<Expeiece>

การกำหนดสภาพแวดล้อมต้องพิจารณาไปถึงปัจจัยที่จะเข้ามามีบทบาททำให้ระบบการเปลี่ยนแปลงด้วย ซึ่งนอกจากกำหนดคอนคันของการพัฒนาโดยมีจุดประสงค์หลังให้สามารถพัฒนาระบบได้ง่ายสภาพแวดล้อมมีความชัดเจน ซึ่งก็มักจะกำหนดสภาพแวดล้อมให้มีความยืดหยุ่นในระดับหนึ่ง เพื่อว่าระบบเปลี่ยนแปลงไปตามลักษณะที่คาดว่าจะเป็นไปได้ที่จะเกิดขึ้น สภาพแวดล้อมที่ได้พิจารณาเอาไว้แล้วจะสามารถรองรับได้โดยไม่มีการเปลี่ยนแปลง ปัญหาสำคัญในการกำหนดสภาพแวดล้อมคือต้องคุยกับเจ้าของโครงการให้เข้าใจว่าเหตุใดต้องการสภาพแวดล้อมแบบนี้ และเหตุใดจึงกำหนดแบบนี้ไม่ได้ ทั้งนี้บางครั้งการกำหนดบางอย่างอาจถูกกำหนดไว้ตั้งแต่การเริ่มโครงการแล้ว ซึ่งก็ควรที่จะกลับไปแก้ไขการกำหนดคอนคันเสีย แม้อาจจะดูไม่สมควรแต่ด้วยเหตุผลก็ควรจะสามารถทำได้เพื่อให้เกิดประโยชน์รวมทั้งสองส่วนและในระยะยาว

</Expeiece>

<ProcessExample />

</ProcessPattern>

<SafetyRelatedPattern Pattern_ID="SPT001">

<CreateDate>2006-03-21</CreateDate>

<ModifyDate>2006-03-21</ModifyDate>

<Author>เอกชัย แซ่ตั้ง</Author>

<Category>

<cat>ATP</cat>

<cat>Train Control</cat>

</Category>

<Keywords />

<Comment>สร้างขึ้นเพื่อใช้ในการทดลอง</Comment>

<Agreements>

<agree>3</agree>

<disagree>0</disagree>

</Agreements>

<PatternState>pattern</PatternState>

<PatternName>ATP: Safety Train separation</PatternName>

<PatternIntroduction>Safety train separate เป็นการกำหนดระยะห่างหรือช่องว่างที่สามารถป้องกันหรือลดความเสี่ยงจากอุบัติเหตุที่เกิดขึ้น โดยเป็นการตั้งค่าเพื่อกำหนดการเว้นระยะขบวนที่รองรับขบวน

รูปแบบต่างๆ ซึ่งจะทำงานสอดคล้องกับระบบที่เกี่ยวข้องจากภายนอกหรือแม้แต่รถไฟที่ไม่อยู่ในการควบคุมด้วยระบบอัตโนมัติ โดยการทำงานจะอิงตามรูปแบบการหยุดอย่างปลอดภัย</PatternIntroduction>

```

<Force>
  <F seq="1">ระบบรถไฟที่ควบคุมด้วยระบบอัตโนมัติหรือเรียกอีกอย่างว่ารถไฟอัตโนมัติ</F>
  <F seq="2">เป็นระบบที่ใช้การบริหารงานโดยการอิงระยะห่างของราง (ไม่ใช่รางแบบแบ่งช่วงของการใช้งาน) </F>
</Force>
<ResultContext>ระบบที่เว้นระยะรถไฟตามความเร็วและตำแหน่งในขณะเวลานั้น ทำให้ใช้รางได้อย่างมีประสิทธิภาพ ระบบซึ่งสามารถหยุดหรือชะลอความเร็วของรถไฟโดยอัตโนมัติเพื่อให้เกิดความปลอดภัยในระยะการเคลื่อนที่
</ResultContext>
<KnownUse />
<RelatedPatterns />
<Goal>
  <G seq="1">
    <detail>การเว้นระยะห่างที่เหมาะสมจะทำให้เกิดความปลอดภัยจากระยะที่สามารถหยุดหรือหลีกเลี่ยงการชนของรถไฟได้ โดยรถไฟจะต้องสามารถไปอยู่ในตำแหน่งปลอดภัยได้เมื่อเกิดเหตุการณ์ใดๆขึ้น โดยอาศัยการกำหนดควบคุมระยะห่างที่เหมาะสมจากเหตุร้ายที่ตามมา</detail>
    <eval>ระยะห่างระหว่างรถไฟจะถูกกำหนดให้อยู่ในช่วงที่ปลอดภัยเสมอ</eval>
  </G>
  <G seq="2">
    <detail>เมื่อระบบเกิดเหตุการณ์ใดๆขึ้น ซึ่งเป็นผลให้รถไฟต้องหยุดเพื่อความปลอดภัย เช่น มีรถจอดเสีย หรือเข้าสู่ทางที่ไร้ร่วม ระบบจะต้องคำนวณระยะปลอดภัยในการหยุด และเมื่อเข้าสู่ระยะที่วิกฤตเข้าสู่อันตรายได้ระบบก็จะดำเนินการในส่วนของการหยุดโดยทันที</detail>
    <eval>ระบบจะนำไปสู่การหยุดได้อย่างทันทีทั้งที่</eval>
  </G>
</Goal>
<DomainProblem>
  <DMP seq="1">
    <problem>ระยะห่างที่เหมาะสมจะพิจารณาจากอะไร</problem>
    <type>improve</type>
    <detail>การพิจารณาระยะห่างที่เหมาะสมต้องพิจารณาไปตามลักษณะของของอีกสิ่งหนึ่งประกอบ เช่น เป็นรถไฟด้วยกัน เป็นสุดทาง เป็นทางชำรุด และยังมีปัจจัยที่ขึ้นกับตัวรถไฟ ณ ขณะเวลา ความเร็ว และตำแหน่งที่แตกตัวกันออกไปด้วย</detail>
    <from>
      <who>ฝ่ายวิศวกรที่ออกแบบระบบ</who>
      <when>ทำการกำหนดระยะปลอดภัย</when>
    </from>
    <rootcause>การกำหนดระยะห่างมีปัจจัยจากหลายส่วน</rootcause>
  </DMP>
</DomainProblem>
<DevelopmentProblem>
  <DEVP seq="1">

```



```

    <problem>การคำนวณระยะห่างที่ปลอดภัยควรทำในขณะใด</problem>
    <type>performance</type>
    <detail>การคำนวณหาระยะห่างที่มีปัจจัยที่เกี่ยวข้องหลายส่วน จึงเป็นเรื่องยากที่จะกำหนดเป็น
ระยะห่างที่คงตัว ดังนั้นระยะห่างที่ปลอดภัยจึงเกิดจากการคำนวณตามปัจจัยที่ถูกต้องในช่วงเวลานั้นๆ ซึ่งเป็นปัญหาว่า
การคำนวณควรเกิดขึ้นเมื่อไรหรือตลอดเวลา</detail>
  </DEVP>
  <DEVP seq="2">
    <problem>ระบบข้อมูลและอุปกรณ์ตรวจสอบค่าต่างๆจะถูกต้องเพียงพอหรือไม่</problem>
    <type>technology</type>
    <detail>การคำนวณเกี่ยวกับระยะจะต้องอาศัยข้อมูลจากหลายๆส่วนประกอบกันเพื่อคำนวณระยะห่างที่
จะปลอดภัย ข้อมูลที่จะนำมาใช้ในการคำนวณจึงมีความสำคัญมาก และมีความเป็นไปได้ที่จะได้ข้อมูลไม่ครบหรือไม่
ถูกต้อง ในการพัฒนาต้องเอาประเด็นเหล่านี้มาใช้ประกอบด้วย ต้องวิเคราะห์ถึงลักษณะของอุปกรณ์และความแม่นยำ
ประกอบด้วย โดยอุปกรณ์ในตรวจวัดค่าบางอย่างอาจต้องการความแม่นยำมากกว่าปกติด้วย</detail>
  </DEVP>
</DevelopmentProblem>
<BusinessModel>
  <diagram seq="1">
    <name>ลักษณะของระยะห่างที่ปลอดภัย</name>
    <description>แสดงระยะห่างที่รถไฟจะสามารถหยุดได้อย่างปลอดภัยโดยไม่ชน (มีระยะเพียงพอ
ในการหยุด)</description>
    <source>SPt001-01.jpg</source>
  </diagram>
  <diagram seq="2">
    <name>การสั่งการหยุดรถไฟเมื่อถึงจุดเสี่ยง</name>
    <description>ระบบจะดำเนินการส่วนของระบบหยุดรถไฟโดยอัตโนมัติเมื่อรถไฟเข้าสู่จุดที่ระยะห่างที่
เหลืออาจเกิดอันตราย (ระยะไม่เพียงพอในการหยุด)</description>
    <source>SPt001-02.jpg</source>
  </diagram>
  <topicDetail>
    <topic>การพิจารณาระยะหยุด</topic>
    <description>ระยะหยุดรถจะพิจารณาโดยอ้างอิงตามรูปแบบการหยุดอย่างปลอดภัย (Safety
Braking Model)</description>
  </topicDetail>
</BusinessModel>
<Environments>
  <ent seq="1">
    <type>system</type>
    <detail>ระบบกำกับการเดินรถไฟ (Train Supervisions) ซึ่งจะเป็นตัวกำหนดการใช้รางการ
ปล่อยรถ ซึ่งมีผลต่อระยะห่างและการใช้รางของรถไฟแต่ละขบวน</detail>
  </ent>
</Environments>
<SystemConstrain>
  <const seq="1">การทำงานจะสามารถนำไปสู่การหยุดได้อย่างทันทั่วทั้งที่</const>
  <const seq="2">ยกเลิกการทำงานในกรณีดำเนินการเพื่อทำให้ระบบที่เสียกลับมาใช้งานได้</const>

```

```

    <const seq="3">ค่าความไม่แน่นอนของตำแหน่ง ค่าความเร็ว ความคลาดเคลื่อน จะต้องมีมา
พิจารณาด้วย</const>
  </SystemConstrain>
  <Requirements>
    <functionReq Req_ID="F1">
      <from>ผู้จัดการระบบ</from>
      <detail>จะต้องคำนวณระยะปลอดภัยได้อย่างถูกต้อง</detail>
    </functionReq>
    <functionReq Req_ID="F2">
      <from>ผู้จัดการระบบ</from>
      <detail>จะต้องครอบคลุมจุดที่มีรถเสียอยู่</detail>
    </functionReq>
    <functionReq Req_ID="F3">
      <from>ผู้จัดการระบบ</from>
      <detail>จะต้องทำงานร่วมกับรถไฟที่ไม่อยู่ในระบบได้</detail>
    </functionReq>
    <functionReq Req_ID="F4">
      <from>ผู้จัดการระบบ</from>
      <detail>จะต้องควบคุมในจุดที่มีการใช้รางร่วมกันได้</detail>
    </functionReq>
    <functionReq Req_ID="F5">
      <from>ผู้จัดการระบบ</from>
      <detail>จะต้องนำไปสู่การหยุดรถหากมีการเคลื่อนที่เกินระยะปลอดภัย</detail>
    </functionReq>
    <non-functionReq Req_ID="NF1">
      <from>ผู้จัดการระบบ</from>
      <detail>จะต้องทำงานได้ทั้งกับรถไฟที่มีความยาวแบบคางคกและแบบไม่แน่นอน</detail>
      <type>product_req</type>
    </non-functionReq>
    <non-functionReq Req_ID="NF2">
      <from>ผู้จัดการระบบ</from>
      <detail>จะต้องสอดคล้องกับรูปแบบการหยุดรถอย่างปลอดภัย (Safety braking
model)</detail>
      <type>interface_req</type>
    </non-functionReq>
  </Requirements>
  <DomainSolution>
    <sol seq="1">
      <topic>การคำนวณระยะห่าง</topic>
      <detail>ระยะทางที่ปลอดภัยสำหรับการหยุดรถจะขึ้นอยู่กับตำแหน่งของรถกับวัตถุ ลักษณะวาง
ความเร็วและความสามารถในการหยุดของรถ ซึ่งการคำนวณระยะที่ปลอดภัยจะเป็นการพิจารณาระยะที่เพียงพอใน
การหยุดในขณะนั้น หรือการประมาณระยะที่ไม่ควรเข้าไปใกล้รถขบวนหน้าจนเกินไป</detail>
      <deal>วิธีการหนึ่งที่มักถูกนำมาใช้คือการกำหนดระยะของเส้นทางเป็นช่วงๆ เช่น ช่วงละ 100 เมตร
โดยการคำนวณจะคำนวณเมื่อรถไฟเข้าสู่ช่วงถัดไปของทาง ระยะทางก็จะถูกคำนวณจากความเร็วสูงสุด (ตามที่

```

กำหนดไว้ได้) หรือความเร็วรถไฟขณะนั้นเพื่อคำนวณระยะหยุด และเมื่อเปรียบเทียบระยะทางที่ใช้กับทางของราง ก็จะได้ช่วงของรางที่รถจะหยุดได้ทัน โดยหากเกิดกรณีช่วงของรางนั้นมันช่วงทับกับช่วงการหยุดของรถไฟขบวนอื่น หรือไม่อยู่ในช่วงที่ต้องการก็จะถือว่าเกิดกรณีที่จะต้องดำเนินการแจ้งเพื่อดำเนินการหยุดทั้งแบบปกติและแบบฉุกเฉิน ในทางปฏิบัติการแบ่งช่วงจะพิจารณาตาม track circuit</deal>

```
<conclude>พิจารณาโดยแบ่งรางเป็นช่วง คำนวณระยะขณะนั้นเพื่อตรวจสอบ</conclude>
<reqs>F1</reqs>
</sol>
<sol seq="2">
  <topic>การกำหนดและควบคุมการตอบสนอง</topic>
  <detail>เมื่อรถไฟไม่อยู่ในขีดจำกัดของการกำหนดระยะ นั้นคือระยะและความเร็วเลขขอบเขตที่
กำหนดเป็นระยะที่มีความปลอดภัย ระบบจะต้องทำการตอบสนองเพื่อป้องกันผลที่ตามมา</detail>
  <deal>ระบบจะต้องทำการตอบสนองเมื่อ ระยะหยุดจะเหลือน้อยกว่าระยะปลอดภัย (ไม่ยอมให้น้อย
กว่า) ความเร็วเกินกว่าความเร็วที่ใช้คำนวณตามระยะปลอดภัย และเกิดการกีดขวางฉับพลันในส่วนระยะปลอดภัย
โดยระบบจะต้องดำเนินการแจ้งให้ผู้ควบคุมรถไฟทราบในทันที โดยการเตรียมระบบหยุดรถไฟพร้อมทำงาน ไม่ยอม
ให้เพิ่มความเร็ว ดำเนินการลดความเร็ว รวมไปถึงดำเนินการหยุดแบบฉุกเฉิน</deal>
```

```
<conclude>กำหนดปัจจัยที่ต้องตอบสนองและวิธีการตอบสนองที่จะดำเนินการตาม</conclude>
<reqs>F5</reqs>
</sol>
</DomainSolution>
<ReqRelationshipConstrain>
  <relationship seq="1">
    <topic>การหยุดเมื่อไม่อยู่ในขอบเขตตามรูปแบบการหยุดอย่างปลอดภัย</topic>
    <req>
      <req_rel>
        <re_ref>F5</re_ref>
        <piority>1</piority>
      </req_rel>
      <req_rel>
        <re_ref>NF2</re_ref>
        <piority>1</piority>
      </req_rel>
    </req>
    <detail>การทำงานของระบบจะอิงกับรูปแบบการหยุดอย่างปลอดภัย ซึ่งจะเป็นรายละเอียดของ
ระยะที่จะต้องทำการหยุดตามความเร็วในขณะนั้น เมื่อเห็นว่าอยู่ในระยะที่ไม่ปลอดภัยแล้ว การหยุดก็จะถูก
ดำเนินการขึ้นในทันที โดยบวกค่าระยะที่เผื่อไว้ตามที่มีการกำหนดด้วย การหยุดและการควบคุมตามรูปแบบการหยุด
อย่างปลอดภัยจะต้องพิจารณาควบคู่กันไป</detail>
```

```
</relationship>
<relationship seq="2">
  <topic>การคำนวณระยะที่ปลอดภัยเกี่ยวกับรถไฟที่อยู่และไม่อยู่ในระบบ</topic>
  <req>
    <req_rel>
      <re_ref>F2</re_ref>
      <piority>1</piority>
    </req_rel>
    <req_rel>
      <re_ref>F3</re_ref>
```



```

    <piority>1</piority>
  </req_rel>
  <req_rel>
    <re_ref>F1</re_ref>
    <piority>2</piority>
  </req_rel>
  <req_rel>
    <re_ref>F4</re_ref>
    <piority>2</piority>
  </req_rel>
</req>
<detail>เมื่อรางในระบบมีการใช้งานร่วมกับรถไฟอื่นๆที่ไม่ได้ควบคุมโดยระบบ เช่น รถไฟที่
เคลื่อนที่เป็นกรณีพิเศษ รถไฟที่ไม่มีระบบกึ่งอัตโนมัติในการควบคุม เมื่อเป็นรถไฟในระบบข้อมูลต่างๆมีความ
สมบูรณ์กว่าและสามารถควบคุมและกำหนดการทำงานของรถไฟได้ แต่เมื่อทำงานร่วมกับรถไฟที่ไม่อยู่ในระบบอาจทำ
ให้เกิดปัญหาในการควบคุมรถไฟที่อยู่นอกระบบ การใช้รางร่วมกันจึงต้องให้ความสำคัญกับรถไฟที่อยู่นอกระบบด้วย
และเนื่องจากขาดความสามารถในการจัดการอย่างสมบูรณ์ ในการควบคุมระยะห่างซึ่งไม่สามารถกำหนดที่รถไฟที่อยู่นอกระบบได้
รถไฟในระบบจึงต้องได้รับการควบคุมอย่างเคร่งครัดและให้การเว้นระยะเป็นไปตามรถไฟนอกระบบเมื่อมีการ
ใช้รางร่วมกัน</detail>
</relationship>
</ReqRelationshipConstrain>
<StakeholderViewpoint>
  <stakeholder seq="1">
    <role>พนักงานควบคุมไฟ</role>
    <views>
      <req>
        <re_ref>F3</re_ref>
        <view>หากมีรถไฟขบวนอื่นๆที่นอกเหนือจากที่ถูกควบคุมอยู่โดยระบบ ระบบควรสามารถบอกให้
ทราบได้ด้ว เนื่องจากการควบคุมรถไฟเหล่านี้เป็นลักษณะ manual จึงอาจจะต้องติดต่อกับศูนย์ควบคุมและอาจต้อง
ระมัดระวังเป็นพิเศษ</view>
      </req>
      <req>
        <re_ref>NF2</re_ref>
        <view>เมื่อรถไฟกำลังจะเข้าสู่ระยะที่ทำการจำกัดก็ควรจะเตือนในทันที เช่น หากวิ่งด้วยความเร็ว
ปัจจุบันต่อไปอีก 5 นาที จะทำให้รถชนกับรถขบวนข้างหน้า และหากเข้าสู่ระยะที่วิกฤติแล้วระบบก็ควรจะสามารถ
ชะลอความเร็วเองได้ หรือทำการหยุดอย่างกะทันหันโดยอัตโนมัติเมื่อเกิดเหตุการณ์ฉุกเฉิน</view>
      </req>
    </views>
  </stakeholder>
</StakeholderViewpoint>
<ProcessPatterns />
<DomainExample />
<SafetyAcceptance>
  <acceptance seq="1">
    <detail>ระยะห่างอยู่ในช่วงไม่ปลอดภัย</detail>
    <cataloglevel>Serious</cataloglevel>
    <operatelevel>ผู้บริหาร</operatelevel>
  </acceptance>
  <acceptance seq="2">

```

```

    <detail>การหยุดหรือชะลอความเร็วโดยอัตโนมัติ</detail>
    <cataloglevel>Hight</cataloglevel>
    <operatelevel>หัวหน้าโครงการ</operatelevel>
  </acceptance>
  <acceptance seq="3">
    <detail>การควบคุมไฟรดไฟให้เว้นระยะอย่างเหมาะสม</detail>
    <cataloglevel>Hight</cataloglevel>
    <operatelevel>พนักงานควบคุมรถไฟ</operatelevel>
  </acceptance>
</SafetyAcceptance>
<SafetyRequirements>
  <sre SR_ID="SR1">
    <type>protect</type>
    <detail>ระยะห่างระหว่างรถไฟน้อยเกินกว่าจะปลอดภัยจากอุบัติเหตุ</detail>
  </sre>
  <sre SR_ID="SR2">
    <type>protect</type>
    <detail>ระยะจากรถไฟถึงสิ่งกีดขวางหรือจุดสิ้นสุดของรางน้อยเกินกว่าจะหยุดรถได้ทัน</detail>
  </sre>
  <sre SR_ID="SR3">
    <type>detect</type>
    <detail>การเพิ่มความเร็วที่ทำให้ระยะห่างระหว่างรถไฟไม่เพียงพอต่อการหยุดรถ</detail>
  </sre>
  <sre SR_ID="SR4">
    <type>react</type>
    <detail>ระยะที่เหลือไม่เพียงพอสำหรับการเว้นระยะที่ปลอดภัย</detail>
  </sre>
  <sre SR_ID="SR5">
    <type>react</type>
    <detail>การเพิ่มความเร็วที่ทำให้ระยะห่างระหว่างรถไฟไม่เพียงพอต่อการหยุดรถ</detail>
  </sre>
</SafetyRequirements>
<SafetySignificant>
  <ssig seq="1">
    <sreq>การคำนวณระยะห่างผิดพลาด</sreq>
    <probability>probable</probability>
    <severity>critical</severity>
  </ssig>
  <ssig seq="2">
    <sreq>ความคลาดเคลื่อนของการวัดความเร็วและตำแหน่ง</sreq>
    <probability>probable</probability>
    <severity>marginal</severity>
  </ssig>
</SafetySignificant>
<AccidentIncident>
  <ac seq="1">
    <type>hardware</type>
    <detail>ตัวตรวจจับความเร็วของรถทำงานผิดพลาด</detail>
  </ac>
  <ac seq="2">
    <type>hardware</type>

```

```

    <detail>ตัวตรวจจับตำแหน่งของรถทำงานผิดพลาด</detail>
  </ac>
  <ac seq="3">
    <type>software</type>
    <detail>การคำนวณระยะห่างไม่ถูกต้อง</detail>
  </ac>
  <ac seq="4">
    <type>human</type>
    <detail>ทำการควบคุมรถโดยไม่ระมัดระวัง</detail>
  </ac>
  <ac seq="5">
    <type>software</type>
    <detail>ระบบเตือนการหยุดรถหรือการชะลอรถไม่ทำงาน</detail>
  </ac>
</AccidentIncident>
<RiskHazard>
  <rh seq="1">
    <detail>รถไฟชนกัน</detail>
    <effect>รถไฟเสียหายโดยอาจทำให้รถไฟดกรางอีกด้วย ทำให้ผู้โดยสารได้รับอันตรายโดยเฉพาะการ
ชนกันที่มีความเร็วหรือในลักษณะประสานงาหรือคกราง ทำให้ระบบหรือเครือข่ายหยุดชะงัก และอาจก่อให้เกิด
อุบัติเหตุอื่นๆตามมา</effect>
  </rh>
  <rh seq="2">
    <detail>รถไฟชนสิ่งกีดขวางหรือวิ่งเลี้ยว</detail>
    <effect>รถไฟเสียหายและคกราง ทำให้ผู้โดยสารได้รับอันตราย</effect>
  </rh>
</RiskHazard>
<SafetyRelatedExample />
</SafetyRelatedPattern>
</RequirementsPatternSystem>

```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

คำถามประกอบการทดลอง

ก.1 แนวทางในการตั้งคำถาม

ตอนที่ 1 คำถามปรนัย

ประกอบด้วยคำถามแบบตัวเลือกจำนวน 10 ข้อ แต่ละข้อมี 5 ตัวเลือก (ก. ถึง จ.)

คำถามข้อที่ 1 ทดสอบว่าผู้ตอบคำถามเข้าใจเป้าหมายของงานหรือไม่

คำถามข้อที่ 2 ทดสอบว่าผู้ตอบคำถามเข้าใจปัญหาที่เกิดขึ้นในแง่ของผู้ให้ความต้องการ

และผู้พัฒนา

คำถามข้อที่ 3 ทดสอบว่าผู้ตอบคำถามสามารถระบุข้อจำกัดของระบบได้

คำถามข้อที่ 4 ทดสอบว่าผู้ตอบคำถามทราบว่าใครคือผู้ใช้และผู้ให้ความต้องการของ

ระบบบ้าง

คำถามข้อที่ 5 ทดสอบว่าผู้ตอบคำถามสามารถหาความต้องการได้

คำถามข้อที่ 6 ทดสอบว่าผู้ตอบคำถามเล็งเห็นปัจจัยและความเกี่ยวข้องอื่นๆ ในการทำ

Requirements

คำถามข้อที่ 7 ทดสอบว่าผู้ตอบคำถามสามารถหาความต้องการความปลอดภัยได้

คำถามข้อที่ 8 ทดสอบว่าผู้ตอบคำถามสามารถระบุนอันตรายที่อาจเกิดขึ้นได้

คำถามข้อที่ 9 ทดสอบว่าผู้ตอบคำถามเข้าใจระดับการยอมรับทางด้านความปลอดภัย

คำถามข้อที่ 10 ทดสอบว่าผู้ตอบคำถามสามารถเลือกเทคนิควิธีการที่จะนำมาใช้ในงานได้

เป็นอย่างดี

ตอนที่ 2 คำถามอัตนัย

ประกอบด้วยคำถาม 5 ข้อ ดังนี้

คำถามข้อที่ 11 ทดสอบว่าผู้ตอบคำถามเล็งเห็นปัญหาสามารถแก้ปัญหาที่มีอยู่ใน

กระบวนการวิศวกรรมความต้องการซอฟต์แวร์

คำถามข้อที่ 12 ทดสอบว่าผู้ตอบคำถามมีความเข้าใจและเห็นความสัมพันธ์ระหว่างความ

ต้องการ

คำถามข้อที่ 13 ทดสอบว่าผู้ตอบคำถามสามารถหาและสรุปต้นเหตุที่นำไปสู่อันตรายได้

คำถามข้อที่ 14 ทดสอบว่าผู้ตอบคำถามสามารถระบุความต้องการที่ไม่ใช่หน้าที่ได้

คำถามข้อที่ 15 ทดสอบว่าผู้ตอบคำถามสามารถเชื่อมโยงไปสู่ขั้นการออกแบบระบบอย่าง

ปลอดภัยได้

ตอนที่ 3 คำถามที่ใช้ประเมินส่วนเนื้อหาในแบบรูป

ใช้ในการประเมินความถูกต้องของแบบรูป เมื่อมีการใช้แบบรูปในการทำความเข้าใจความต้องการเท่านั้น โดยให้คะแนนเป็น 1 ถึง 5 แทนความหมาย น้อยมาก น้อย ปานกลาง ดี และดีมาก โดยประกอบด้วยคำถาม 5 ข้อ ดังนี้

คำถามข้อที่ 1 ความครบถ้วนสมบูรณ์ของเนื้อหาในแบบรูป

คำถามข้อที่ 2 ความถูกต้องของเนื้อหาในแบบรูป

คำถามข้อที่ 2 ความง่ายในการใช้และทำความเข้าใจแบบรูป

คำถามข้อที่ 3 ความชัดเจนของเนื้อหาในแบบรูป

คำถามข้อที่ 4 ความมีประสิทธิภาพจากการใช้งานแบบรูป

ชุดคำถาม ประเมินและแสดงความเห็นเกี่ยวกับแบบรูป

คำถามในส่วนนี้จะประเมินความเห็น โดยให้ลำดับความเห็นเป็น น้อยที่สุด น้อย ปานกลาง มาก และมากที่สุด และให้คะแนนการประเมินเป็น 1 ถึง 5 ตามลำดับ

1. ประเมินเพื่อวัดความเห็นต่อการทำงาน โดยใช้เป็นคำถามข้อ 1 ถึง 4
2. ประเมินเพื่อวัดความเห็นเกี่ยวกับโครงสร้างและองค์ประกอบ โดยใช้เป็นคำถามข้อ 5 ถึง 7
3. ประเมินเพื่อวัดความรู้สึของผู้ใช้งาน โดยใช้เป็นคำถามข้อ 8 ถึง 10

ก.2 ชุดคำถามที่ 1 Safety Train Separate

ตอนที่ 1 จงเลือกคำตอบที่ถูกต้องที่สุดเพียง 1 ข้อ

1. ข้อใดคือเป้าหมายของการทำงานในส่วนของ Safety train separate
 - ก. กำหนดระยะห่างระหว่างรถไฟ
 - ข. ควบคุมระยะห่างให้สามารถหยุดอย่างปลอดภัย
 - ค. ควบคุมตำแหน่งรถไฟให้มีความปลอดภัย
 - ง. ข้อ ก. และ ข.
 - จ. ข้อ ก. และ ค.
2. ข้อใดไม่ใช่ปัญหาในงานวิศวกรรมความต้องการสำหรับ safety train separate
 - ก. ผู้ให้ความต้องการ ไม่สามารถระบุปัจจัยที่ใช้เป็นตัวกำหนดระยะห่างได้อย่างสมบูรณ์
 - ข. การกำกับเวลาหรือจังหวะที่จะมีการคำนวณเกี่ยวกับระยะห่าง
 - ค. ความถูกต้องของระบบข้อมูลและตัวอุปกรณ์

- ง. ค่าความกลาดเคลื่อนที่เกิดขึ้นในการคำนวณระยะห่าง
 - จ. ข้อ ข. และ ง.
3. ข้อใดไม่ใช่ข้อจำกัด (System Constraint) ในส่วนของ safety train separate
 - ก. การทำงานจะนำไปสู่การหยุดรถที่ปลอดภัย
 - ข. การขอมให้ยกเลิกการทำงานในกรณีที่เกี่ยวข้องกับการทำให้ระบบที่เสียกลับมาใช้งาน
 - ค. การเว้นระยะห่างเป็นไปตามการกำกับการเดินรถไฟ
 - ง. พิจารณาค่าความไม่แน่นอนของตำแหน่งรถไฟให้อยู่ในช่วงที่กำหนด
 - จ. พิจารณาค่าความไม่แน่นอนของความเร็วรถไฟให้อยู่ในช่วงที่กำหนด
4. ข้อใดคือกลุ่มคนที่เกี่ยวข้องกับความต้องการของ Safety train separate
 - ก. พนักงานควบคุมรถไฟ
 - ข. พนักงานควบคุมรถไฟ และหัวหน้าโครงการ
 - ค. พนักงานควบคุมรถไฟ ผู้บริหาร และหัวหน้าโครงการ
 - ง. พนักงานกำหนดเส้นทาง
 - จ. พนักงานกำหนดเส้นทาง และหัวหน้าโครงการ
5. ข้อใดไม่จัดว่าเป็นความต้องการของ safety train separate
 - ก. จะต้องคำนวณระยะปลอดภัยได้อย่างถูกต้อง
 - ข. จะต้องทำงานร่วมกับรถไฟที่ไม่ได้มีการควบคุมได้
 - ค. จะต้องทำงานได้ทั้งกับรถไฟที่มีความยาวคงที่และไม่คงที่
 - ง. จะต้องนำไปสู่การหยุดหรือชะลอรถหากระยะห่างเข้าสู่จุดที่ไม่ปลอดภัย
 - จ. จะต้องช่วยในการกำหนดขีดความเร็วของรถไฟที่วิ่งได้
6. ข้อใดไม่มีความเกี่ยวข้องกับการพิจารณาความต้องการของ safety train separate
 - ก. ความยาวรถไฟ
 - ข. เส้นทางตัดใช้ร่วมกับทางอื่น
 - ค. สถานีโดยสารหรือชานชาลา
 - ง. ตำแหน่งสิ้นสุดของราง
 - จ. ตำแหน่งของรถไฟ
7. ข้อใดคือหน้าที่ที่เกี่ยวข้องกับความปลอดภัยของ safety train separate
 - ก. ใช้คำนวณระยะห่างระหว่างรถไฟ
 - ข. ใช้กำหนดระยะที่เพียงพอในการหยุดอย่างปลอดภัย
 - ค. ใช้ควบคุมการใช้รางร่วมกันของรถไฟ
 - ง. ใช้ป้องกันไม่ให้รถไฟชนกัน

- จ. ใช้กำหนดควบคุมความเร็วไฟรัศให้เหมาะสม
8. ข้อใดเป็นอันตรายที่อาจเกิดขึ้นจากความผิดพลาดของ safety train separate โดยตรง
- รถไฟวิ่งเลยจุดสิ้นสุดของราง และรถไฟชนสิ่งกีดขวาง
 - รถไฟชนกันเอง และรถไฟตกราง
 - ระยะห่างระหว่างรถไฟน้อยเกินไป
 - รถไฟชนสิ่งกีดขวาง และรถไฟตกราง
 - รถไฟวิ่งด้วยความเร็วสูงจนอันตราย
9. ข้อใดคือผลที่ไม่สามารถยอมรับได้และมีความสำคัญมากสำหรับ safety train separate
- การควบคุมระยะของรถไฟที่ไม่เหมาะสม
 - การมีระยะห่างรถไฟอยู่ในช่วงอันตราย
 - การหยุดหรือชะลอความเร็วโดยอัตโนมัติไม่ทำงาน
 - การคำนวณความเร็วคลาดเคลื่อนจากความเป็นจริง
 - การคำนวณตำแหน่งคลาดเคลื่อนจากความเป็นจริง
10. ข้อใดต่อไปนี้เป็นเทคนิคที่มีความเหมาะสมเป็นพิเศษในการนำมาใช้ในงานวิศวกรรมความต้องการสำหรับ safety train separate
- การบันทึกที่มาของความต้องการ
 - การวิเคราะห์ลำดับความสำคัญของความต้องการ
 - การใช้ทีมที่มีความหลากหลายในการตรวจสอบความต้องการ
 - การแยกแยะและให้คำแนะนำผู้ใช้งานระบบ
 - การกำหนดวัฒนธรรมด้านความปลอดภัยในองค์กร

ตอนที่ 2 จงตอบคำถามต่อไปนี้

- จงยกตัวอย่างปัญหาในงานวิศวกรรมความต้องการที่พบในระบบที่กล่าวถึง พร้อมทั้งอธิบายวิธีการในการจัดการกับปัญหานั้น
- จงยกตัวอย่างความต้องการที่มีความสัมพันธ์กันพร้อมทั้งอธิบาย
- จงยกตัวอย่างความผิดพลาดจาก Software, hardware และ human ที่ส่งผลต่อความปลอดภัย มา 3 ข้อ
- จงยกตัวอย่างความต้องการที่ไม่ใช่หน้าที่ มา 3 ข้อ
- จงยกตัวอย่างอันตรายที่นำไปพิจารณาประกอบเมื่อถึงขั้นตอนการออกแบบระบบ มา 3 ข้อ

ตอนที่ 3 คำถามที่ใช้ประเมินส่วนเนื้อหาในแบบรูป

ให้ตอบคำถามในตอนที่ 3 เมื่อมีการใช้แบบรูปประกอบการทดสอบเท่านั้น

ข้อ 16 – 20 ให้คะแนน 1 – 5 แทนความหมาย น้อยมาก น้อย ปานกลาง ดี และดีมาก ตามลำดับ

16. ท่านคิดว่าแบบรูปที่ท่านใช้มีความครบถ้วนสมบูรณ์ระดับใด
17. ท่านคิดว่าแบบรูปที่ท่านใช้มีความถูกต้องในระดับใด
18. ท่านคิดว่าแบบรูปที่ท่านใช้มีความง่ายในการใช้และทำความเข้าใจระดับใด
19. ท่านคิดว่าแบบรูปที่ท่านใช้มีความชัดเจนในระดับใด
20. ท่านคิดว่าแบบรูปที่ท่านใช้ช่วยแก้ปัญหาการทำความเข้าใจความต้องการนี้ในระดับใด

ค.3 ชุดคำถามที่ 2 Door open control protection interlocks

ตอนที่ 1 จงเลือกคำตอบที่ถูกต้องที่สุดเพียง 1 ข้อ

1. ข้อใดเป็นเป้าหมายของ Door Open Control Protection Interlocks
 - ก. มีการควบคุมการเปิดประตู
 - ข. เปิดประตูด้วยเงื่อนไขที่ปลอดภัย
 - ค. สามารถเปิดประตูในกรณีฉุกเฉินได้
 - ง. ถูกเฉพาะข้อ ก. และ ข.
 - จ. ถูกทุกข้อ
2. ข้อใดคือปัญหาในงานวิศวกรรมความต้องการสำหรับ Door Open Control Protection Interlocks
 - ก. การหาปัจจัยที่เกี่ยวข้องกับความปลอดภัย
 - ข. รูปแบบหรือประเภทของประตู(เปิดทางเดียว เปิดสองทาง)
 - ค. ความยุ่งยากในการกำหนดความสัมพันธ์กับอุปกรณ์ที่เกี่ยวข้องกับการประตู
 - ง. ข้อ ก. และ ข.
 - จ. ข้อ ก. และ ค.
3. ข้อใดคือข้อจำกัด (System Constraint) ในส่วนของ Door Open Control Protection Interlocks
 - ก. ตำแหน่งการเปิดประตู
 - ข. ความเร็วในการเปิดประตู
 - ค. สัญญาณแจ้งการเปิดประตู
 - ง. การเปิดประตูต้องทำขณะที่รถหยุด
 - จ. การเปิดประตูฉุกเฉิน
4. ข้อใดคือกลุ่มคนที่เกี่ยวข้องกับความต้องการของ Door Open Control Protection Interlocks
 - ก. พนักงานควบคุมรถไฟ และผู้โดยสาร
 - ข. พนักงานควบคุมรถไฟ ผู้จัดการระบบ และผู้โดยสาร
 - ค. พนักงานกำหนดเส้นทางเดินรถ พนักงานควบคุมรถไฟ และผู้โดยสาร
 - ง. ผู้จัดการระบบ และผู้โดยสาร

- จ. พนักงานประจำสถานีโดยสารรถไฟ พนักงานควบคุมรถไฟ และผู้โดยสาร
5. ข้อใดไม่จัดว่าเป็นความต้องการใน Door Open Control Protection Interlocks
- ต้องสามารถเปิดประตูได้
 - ต้องสามารถเปิดประตูกรณีฉุกเฉินได้
 - ต้องมีการแจ้งการเปิดประตู
 - ต้องมีการตรวจสอบความปลอดภัยก่อนจะเปิดประตูได้
 - ต้องสามารถควบคุมการเปิดประตูจากภายนอกได้ด้วย
6. ข้อใดไม่มีความเกี่ยวข้องกับการพิจารณาความต้องการของ Door Open Control Protection Interlocks
- ความยาวรถไฟ
 - ตำแหน่งจอดเทียบชานชาลา
 - ระยะเวลาการจอดรถ
 - การเคลื่อนที่ของรถไฟ
 - ตำแหน่งการจอดรถไฟ
7. ข้อใดคือหน้าที่ที่เกี่ยวข้องกับความปลอดภัยของ Door Open Control Protection Interlocks
- การกำหนดระยะเวลาการเปิดประตู
 - การป้องกันการชนกันของรถไฟ
 - การเตือนการเปิดประตู
 - การป้องกันการเปิดประตูขณะรถยังวิ่งอยู่
 - การเปิดประตูกรณีฉุกเฉิน
8. ข้อใดเป็นอันตรายที่อาจเกิดขึ้นจากความผิดพลาดของ Door Open Control Protection Interlocks โดยตรง
- ประตูหนีบผู้โดยสารที่ผ่านเข้าออก
 - ผู้โดยสารออกจากรถไฟบริเวณนอกสถานีโดยสาร
 - ประตูเปิดผิดทาง (เช่น ประตูเปิดข้างใต้ชานชาลาอยู่ขวา)
 - รถไฟหยุดเนื่องจากมีความพยายามในการเปิดประตูฉุกเฉิน
 - ผู้โดยสารพึงประตูขณะที่กำลังเปิด
9. ข้อใดคือผลที่ไม่สามารถยอมรับได้และมีความสำคัญมากสำหรับ Door Open Control Protection Interlocks
- การเปิดประตูไม่สามารถทำได้
 - การเปิดประตูทำให้เกิดอันตราย

- ก. รถหยุดอันเนื่องมาจากการเปิดประตูฉุกเฉิน
 - ง. ตำแหน่งการเปิดประตูไม่ตรงตามที่ขานขาลาระบุไว้
 - จ. การควบคุมการเปิดประตูจากภายนอกไม่สามารถทำได้
10. ข้อใดต่อไปนี้เป็นเทคนิคที่มีความเหมาะสมเป็นพิเศษในการนำมาใช้ในงานวิศวกรรมความต้องการสำหรับ Door Open Control Protection Interlocks
- ก. การบันทึกที่มาของความต้องการ
 - ข. การวิเคราะห์ลำดับความสำคัญของความต้องการ
 - ค. การใช้ทีมที่มีความหลากหลายในการตรวจสอบความต้องการ
 - ง. การแยกแยะและให้คำแนะนำผู้ใช้งานระบบ
 - จ. รายการตรวจสอบความปลอดภัย

ตอนที่ 2 จงตอบคำถามต่อไปนี้

11. จงยกตัวอย่างปัญหาในงานวิศวกรรมความต้องการที่พบในระบบที่กล่าวถึง พร้อมทั้งอธิบายวิธีการในการจัดการกับปัญหานั้น
12. จงยกตัวอย่างความต้องการที่มีความสัมพันธ์กันพร้อมทั้งอธิบาย
13. จงยกตัวอย่างความผิดพลาดจาก Software, hardware และ human ที่ส่งผลต่อความปลอดภัย มา 3 ข้อ
14. จงยกตัวอย่างความต้องการที่ไม่ใช่น้ำที่ มา 3 ข้อ
15. จงยกตัวอย่างอันตรายที่นำไปพิจารณาประกอบเมื่อถึงขั้นตอนการออกแบบระบบ มา 3 ข้อ

ตอนที่ 3 คำถามที่ใช้ประเมินส่วนเนื้อหาในแบบรูป

ให้ตอบคำถามในตอนที่ 3 เมื่อมีการใช้แบบรูปประกอบการทดลองเท่านั้น

- ข้อ 16 – 20 ให้คะแนน 1 – 5 แทนความหมาย น้อยมาก น้อย ปานกลาง ดี และดีมาก ตามลำดับ
16. ท่านคิดว่าแบบรูปที่ท่านใช้มีความครบถ้วนสมบูรณ์ระดับใด
 17. ท่านคิดว่าแบบรูปที่ท่านใช้มีความถูกต้องในระดับใด
 18. ท่านคิดว่าแบบรูปที่ท่านใช้มีความง่ายในการใช้และทำความเข้าใจระดับใด
 19. ท่านคิดว่าแบบรูปที่ท่านใช้มีความชัดเจนในระดับใด
 20. ท่านคิดว่าแบบรูปที่ท่านใช้ช่วยแก้ปัญหาการทำความเข้าใจในระดับใด

ก.4 ชุดคำถามที่ 3 Train Location and Speed

ตอนที่ 1 จงเลือกคำตอบที่ถูกต้องที่สุดเพียง 1 ข้อ

1. ข้อใดเป็นเป้าหมายของ Train Location and Speed Determination
 - ก. กำหนดตำแหน่ง ความเร็ว อัตราเร่ง และทิศทางได้

- ข. ตรวจสอบและติดตามตำแหน่ง ความเร็ว อัตราเร่ง และทิศทาง ได้อย่างถูกต้อง
 - ค. ควบคุมให้รถไฟใช้ความเร็วตามโปรแกรมที่กำหนดไว้
 - ง. ข้อ ก. และ ข.
 - จ. ถูกทุกข้อ
2. ข้อใดคือปัญหาในงานวิศวกรรมความต้องการสำหรับ Train Location and Speed Determination
- ก. การกำหนดค่าความปลอดภัยของตำแหน่งและความเร็ว
 - ข. การอธิบายวิธีการหาหรือคำนวณตำแหน่งของรถไฟ
 - ค. การพยายามคำนวณความเร็วของรถไฟให้มีความแม่นยำ
 - ง. การกำหนดการ โปรแกรมการใช้ความเร็วของรถไฟ
 - จ. การหาวิธีการ ในการกำหนดตำแหน่งของรถไฟ
3. ข้อใดไม่ใช่ข้อจำกัด (System Constraint) ในส่วนของ Train Location and Speed Determination
- ก. ความคลาดเคลื่อนในการประเมินตำแหน่งรถไฟ
 - ข. ค่าความแม่นยำในการวัดตำแหน่งรถไฟ
 - ค. ความคลาดเคลื่อนในการประเมินความเร็วรถไฟกรณีปกติ
 - ง. ค่าความแม่นยำในการวัดความเร็วรถไฟ
 - จ. ค่าความแม่นยำในการวัดอัตราเร่งของรถไฟ
4. ข้อใดคือกลุ่มคนที่เกี่ยวข้องกับความต้องการของ Train Location and Speed Determination
- ก. พนักงานควบคุม และผู้โดยสาร
 - ข. พนักงานควบคุม และผู้จัดการระบบ
 - ค. พนักงานควบคุม ผู้จัดการระบบ และผู้โดยสาร
 - ง. พนักงานกำหนดเส้นทาง ผู้จัดการระบบ และผู้โดยสาร
 - จ. พนักงานควบคุม และผู้กำหนดเส้นทางเดินรถ
5. ข้อใดไม่จัดว่าเป็นความต้องการ ใน Train Location and Speed Determination
- ก. ต้องควบคุมตำแหน่ง ความเร็ว และทิศทางได้
 - ข. ต้องเริ่มทำงานโดยอัตโนมัติเมื่อรถไฟเข้าสู่ระบบ
 - ค. ต้องพิจารณาความคลาดเคลื่อนประกอบการคำนวณด้วย
 - ง. ต้องกำหนดตำแหน่งทั้งหน้าและหลังรถไฟได้อย่างถูกต้อง
 - จ. ไม่มีข้อใดที่ไม่ใช่ความต้องการสำหรับ Train Location and Speed Determination
6. ข้อใดไม่มีความเกี่ยวข้องกับการพิจารณาความต้องการของ Train Location and Speed Determination
- ก. ตำแหน่งวางรถไฟ

- ข. สถานีโดยสารและชานชาลา
 - ค. ความยาวของรถไฟ
 - ง. ความสามารถในการทำความเร็วของรถไฟ
 - จ. การเคลื่อนที่ของรถไฟ
7. ข้อใดคือหน้าที่ที่เกี่ยวข้องกับความปลอดภัยของ Train Location and Speed Determination
- ก. ทำการป้องกันการคำนวณความเร็วรถไฟผิดพลาด
 - ข. การป้องกันการไถลของรถไฟ
 - ค. การเตือนเมื่อมีการใช้ความเร็วเกินกำหนด
 - ง. การป้องกันการใช้รางร่วมกับรถไฟอื่นทั้งที่ไม่ได้รับอนุญาต
 - จ. การสนองตอบเมื่อตำแหน่งของรถไฟยังไม่มีกำหนด
8. ข้อใดเป็นอันตรายที่อาจเกิดขึ้นจากความผิดพลาดของ Train Location and Speed Determination โดยตรง
- ก. รถไฟวิ่งผิดทางโดยวิ่งสวนกับอีกทางหนึ่งจนชนกับอีกขบวนที่สวนกัน
 - ข. รถไฟอยู่ในตำแหน่งที่ไม่คาดคิดจากความผิดพลาดในการระบุตำแหน่งและทำให้ชนกันขึ้น
 - ค. รถไฟใช้ความเร็วเกินกว่าที่กำหนดจนเป็นผลให้รถไฟตกราง
 - ง. ข้อ ก. และ ข.
 - จ. ข้อ ข. และ ค.
9. ข้อใดคือผลที่ไม่สามารถยอมรับได้และมีความสำคัญมากสำหรับ Train Location and Speed Determination
- ก. การกำหนดรวมทั้งค่าประเมินความคลาดเคลื่อนสำหรับรถไฟมีความผิดพลาด
 - ข. การวัดตำแหน่งและความเร็วของรถไฟคลาดเคลื่อนเกินที่กำหนด
 - ค. การควบคุมรถไฟให้สามารถจอดตามจุดที่กำหนดได้อย่างแม่นยำ
 - ง. ข้อ ก. และ ข.
 - จ. ข้อ ข. และ ค.
10. ข้อใดต่อไปนี้ เป็นเทคนิคที่มีความเหมาะสมเป็นพิเศษในการนำมาใช้ในงานวิศวกรรมความต้องการสำหรับ Train Location and Speed Determination
- ก. การบันทึกที่มาของความต้องการ
 - ข. การวิเคราะห์ลำดับความสำคัญของความต้องการ
 - ค. การใช้ทีมที่มีความหลากหลายในการตรวจสอบความต้องการ
 - ง. การแยกแยะและให้คำแนะนำผู้ใช้งานระบบ

จ. ไม่มีเทคนิคใดที่เหมาะสมเป็นพิเศษ

ตอนที่ 2 จงตอบคำถามต่อไปนี้

11. จงยกตัวอย่างปัญหาในงานวิศวกรรมความต้องการที่พบในระบบที่กล่าวถึง พร้อมทั้งอธิบายวิธีการในการจัดการกับปัญหานั้น
12. จงยกตัวอย่างความต้องการที่มีความสัมพันธ์กันพร้อมทั้งอธิบาย
13. จงยกตัวอย่างความผิดพลาดจาก Software, hardware และ human ที่ส่งผลต่อความปลอดภัย มา 3 ข้อ
14. จงยกตัวอย่างความต้องการที่ไม่ใช่น้ำที่ มา 3 ข้อ
15. จงยกตัวอย่างอันตรายที่นำไปพิจารณาประกอบเมื่อถึงขั้นตอนการออกแบบระบบ มา 3 ข้อ

ตอนที่ 3 คำถามที่ใช้ประเมินส่วนเนื้อหาในแบบรูป

ให้ตอบคำถามในตอนที่ 3 เมื่อมีการใช้แบบรูปประกอบการทดลองเท่านั้น

ข้อ 16 ถึง 20 ให้คะแนน 1 ถึง 5 แทนความหมาย น้อยมาก น้อย ปานกลาง ดี และดีมาก ตามลำดับ

16. ท่านคิดว่าแบบรูปที่ท่านใช้มีความครบถ้วนสมบูรณ์ระดับใด
17. ท่านคิดว่าแบบรูปที่ท่านใช้มีความถูกต้องในระดับใด
18. ท่านคิดว่าแบบรูปที่ท่านใช้มีความง่ายในการใช้และทำความเข้าใจระดับใด
19. ท่านคิดว่าแบบรูปที่ท่านใช้มีความชัดเจนในระดับใด
20. ท่านคิดว่าแบบรูปที่ท่านใช้ช่วยแก้ปัญหาการทำความเข้าใจความต้องการนี้ในระดับใด

ค.5 ชุดคำถาม ประเมินและแสดงความเห็นเกี่ยวกับแบบรูป

ให้คะแนน 1 - 5 แทน น้อยที่สุด - มากที่สุด

1. ท่านคิดว่าการนำแบบรูปมาใช้กับงานวิศวกรรมความต้องการซอฟต์แวร์จะช่วยให้การทำงานง่ายขึ้นมากกว่าการที่ไม่ใช้แบบรูปมากนักเพียงใด
2. ท่านคิดว่าการใช้แบบรูปมาใช้กับงานวิศวกรรมความต้องการซอฟต์แวร์จะช่วยให้การทำงานมีประสิทธิภาพมากกว่าการที่ไม่ใช้แบบรูปมากนักเพียงใด
3. ท่านคิดว่าการใช้แบบรูปมาใช้กับงานวิศวกรรมความต้องการซอฟต์แวร์จะช่วยให้การลดเวลาในการทำงานมากกว่าการที่ไม่ใช้แบบรูปมากนักเพียงใด
4. ท่านคิดว่าการนำแบบรูปมาใช้กับความต้องการความปลอดภัยสัมพันธ์มีความเหมาะสมมากน้อยเพียงใด
5. ท่านคิดว่าโครงสร้างของแบบรูปมีความเหมาะสมมากน้อยเพียงใด
6. ท่านคิดว่าองค์ประกอบและเนื้อหาในแต่ละส่วนขององค์ประกอบที่ใช้ในแบบรูปมีความเหมาะสมมากน้อยเพียงใด

7. ท่านคิดว่าการนำแบบรูปมาใช้กับงานวิศวกรรมความต้องการซอฟต์แวร์โดยเฉพาะเมื่อเปรียบเทียบกับการนำกลับมาใช้ใหม่วิธีอื่นๆที่ท่านรู้จักมีความเหมาะสมมากน้อยเพียงใด อย่างไร
8. ท่านรู้สึกดีหรือพึงพอใจในการใช้งานแบบรูปมากน้อยเพียงใด
9. ท่านรู้สึกว่าแบบรูปที่เสนอนี้ก่อให้เกิดการนำไปใช้อย่างต่อเนื่องมากน้อยเพียงใด
10. ท่านรู้สึกถึงความคุ้มค่าในการใช้แบบรูปที่เสนอนี้น้อยเพียงใด



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายเอกชัย แซ่ตั้ง เกิดเมื่อวันที่ 6 กันยายน 2524 ที่จังหวัดอุบลราชธานี สำเร็จการศึกษาระดับปริญญาตรี สาขาวิศวกรรมศาสตรบัณฑิต จากภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2545 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2546



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย