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 มาแก้ปัญหาที่เกิดขึ้นใหม่ที่มีสภาพแวดล้อม (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 ได้นำแบบรูปมาใช้ในการออกแบบส่วนต่อประสานผู้ใช้โดยภาษาสมอล์ลทอล์ค (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) สภาวะผลลัพธ์ (Result Context) แบบรูปที่เกี่ยวข้อง (Related Patterns) รวมทั้งองค์ประกอบสนับสนุนที่จะช่วยในการอธิบายผลเฉลยของแบบรูปนั้นๆ เช่น ตัวอย่าง (Example) แผนภาพคลาส (Class Diagram) ซึ่งใช้ในแบบรูปการออกแบบ เป็นต้น องค์ประกอบเหล่านี้จะขึ้นอยู่กับการนำแบบรูปไปใช้งาน โดยมีจุดประสงค์เพื่อที่สามารถอธิบาย สนับสนุนการใช้ หรือช่วยในการทำความเข้าใจปัญหาหรือการแก้ปัญหาที่นำเสนอได้ชัดเจนขึ้น

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) คอล์ลาโบเรชัน (Collaboration)

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

สำหรับงานวิทยานิพนธ์นี้ได้นำเสนอโครงสร้างและองค์ประกอบของแบบรูปความต้องการ ดังนั้นจึงมีองค์ประกอบที่แตกต่างออกไปจากแบบรูปการออกแบบ โดยองค์ประกอบสำหรับแบบรูปความต้องการนั้นจะเกี่ยวข้องกับงานวิศวกรรมความต้องการซอฟต์แวร์เป็นหลัก โดยจะนำไปใช้ในกระบวนการพัฒนาซอฟต์แวร์ในช่วงการเก็บความต้องการ การวิเคราะห์ความต้องการ หรือขั้นตอนอื่นๆ ในวิศวกรรมความต้องการซอฟต์แวร์ดังหัวข้อถัดไป

2.1.1.4 การพิจารณาองค์ประกอบพื้นฐานและตัวภาษาของแบบรูป

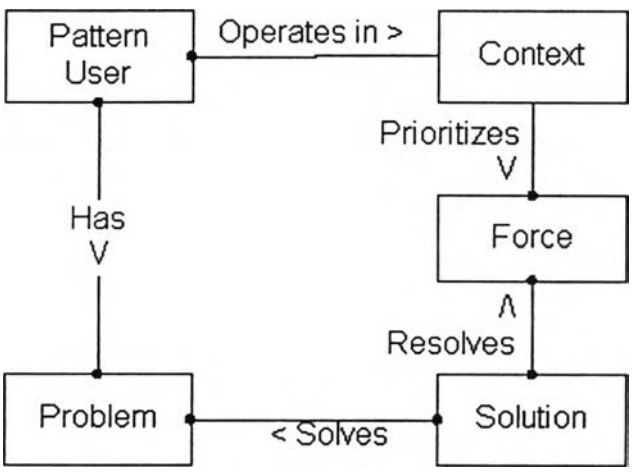
การพิจารณาองค์ประกอบพื้นฐาน และการพิจารณาตัวภาษาของแบบรูป คือ การพิจารณาในส่วนที่เป็นองค์ประกอบที่เป็นแกนหลักของแบบรูป ซึ่งประกอบด้วย ปัญหา สภาพแวดล้อม และผลเฉลย ควบคู่ไปกับภาษาแบบรูป

การนำเสนอผลเฉลยสำหรับปัญหาจำเป็นต้องให้ความสำคัญกับมุมมองของการนำกลับมาใช้ใหม่ ซึ่งจะต้องพิจารณาเพื่อหาวิธีที่ง่ายในการเสนอผลเฉลยโดยเฉพาะที่มีความซับซ้อน

บนสภาพแวดล้อมที่แตกต่างกัน ซึ่งจะส่งผลให้แต่ละแบบรูปมีการแก้ปัญหาที่เฉพาะเจาะจงและชัดเจน โดยแบบรูปควรเป็นเอกเทศ มีสมบูรณ์ในตัวเอง และจำกัดความสัมพันธ์ในการอ้างอิงเพื่อเชื่อมโยงไปยังแบบรูปอื่นๆ สำหรับการใช้งานหนึ่งๆ ให้น้อยที่สุด

2.1.1.5 การพิจารณาโครงสร้างแบบรูปให้เหมาะสม

การพิจารณาแบบรูปให้เหมาะสมอีกนัยหนึ่งคือ การกำหนดองค์ประกอบสำหรับแบบรูปให้มีความสมบูรณ์สอดคล้องกับการใช้งาน ข้อสำคัญในการกำหนดองค์ประกอบแบบรูปคือ การมั่นใจว่ามีข้อมูลสำคัญครบถ้วนเพียงพอ โดยองค์ประกอบที่อยู่ในแบบรูปจะพิจารณาเพื่อการนำแบบรูปไปใช้แก้ปัญหา โดยจะต้องไม่มากเกินความจำเป็นจนทำให้เกิดความยุ่งยากและซับซ้อนในการใช้ ซึ่งจะมีองค์ประกอบเป็นโครงสร้างหลักดังรูปที่ 2.1



รูปที่ 2.1 ความสัมพันธ์ระหว่างองค์ประกอบของแบบรูป [13]

องค์ประกอบแกนหลักของแบบรูปประกอบด้วย

1) สภาพแวดล้อม

2) ปัญหา

3) ผลเฉลย

4) แรงชักจูง

การใช้แรงชักจูงจะช่วยให้มั่นใจว่าทำการเลือกใช้แบบรูปได้อย่างถูกต้องตรงกับปัญหาและสภาพการณ์ที่เป็นอยู่ และยังช่วยให้สามารถละแบบรูปนี้ไปได้โดยง่ายหากเห็นว่าไม่ตรงกับการนำไปใช้

จากองค์ประกอบหลักของแบบรูป จะเห็นได้ว่ายังไม่เพียงพอในการนำไปใช้ จำเป็นต้องมีองค์ประกอบเสริมที่จะนำมาใช้ด้วย ดังนี้

1) สภาวะผลลัพธ์

2) แบบรูปที่เกี่ยวข้อง

3) ตัวอย่าง

4) การนำไปใช้ที่ทราบ

โดยองค์ประกอบเสริมที่ยกมาแสดงนั้นเป็นองค์ประกอบที่ใช้กันทั่วไป ซึ่งที่ยกมาเป็นองค์ประกอบที่มีความเหมาะสมและเพียงพอสำหรับการนำมาใช้งานในแบบรูปความต้องการ

2.1.1.6 การใช้ชื่อและการอ้างอิงแบบรูป

การเรียกชื่อองค์ประกอบนอกเหนือไปจากองค์ประกอบที่เป็นพื้นฐานในโครงสร้างแบบรูป ซึ่งเป็นองค์ประกอบที่มีลักษณะเฉพาะเจาะจงและมีจุดมุ่งหมายเพื่อใช้ในลักษณะพิเศษ การกำหนดชื่อองค์ประกอบเหล่านี้จึงจำเป็นต้องสัมพันธ์กับความหมายตามที่มักจะนำไปใช้จริง ชื่อที่กำหนดควรจะมีความหมายชัดเจนและเป็นตัวแทนของสิ่งที่แบบรูปจะพยายามอธิบาย

แบบรูปควรมีองค์ประกอบภายในที่สัมพันธ์และอ้างอิงถึงกันได้ เพื่อสามารถใช้งานทดแทนหรือประกอบกัน และควรอ้างอิงไปยังแบบรูปอื่นที่มีความเกี่ยวข้องกันโดยพิจารณาลักษณะความสัมพันธ์ภายนอก

การสร้างแบบรูปเพื่อใช้งานตามองค์ประกอบที่กำหนด จำเป็นต้องใช้ให้ตรงกับความหมาย การอ้างอิงและความสัมพันธ์

2.1.1.7 การทำแบบรูปให้เข้าใจง่าย

ในการออกแบบแบบรูปควรกำหนดองค์ประกอบให้ง่ายต่อการทำความเข้าใจ ซึ่งควรพิจารณาในด้านต่างๆ ควรจะพิจารณาในด้านต่อไปนี้

1) ความเข้าใจเฉพาะกลุ่ม

การใช้งานและสร้างแบบรูปควรระวังในเรื่องของความหมายหรือการกำหนดในสิ่งที่จำกัดเฉพาะในกลุ่มผู้เกี่ยวข้องหรือคุ้นเคย อย่างไรก็ดีไม่ควรที่จะใช้ความหมายที่เป็นกลางจนเกินไป จะทำให้ขาดความชัดเจนและมีความหมายกว้างเกินไป

2) การใช้ศัพท์

การใช้คำศัพท์จำเป็นต้องเป็นคำที่ง่ายต่อการเข้าใจ และควรเป็นคำศัพท์ที่รู้จักโดยทั่วไป

3) การใช้สัญลักษณ์

สัญลักษณ์เป็นสิ่งสำคัญที่ช่วยในการเข้าใจและจดจำ อาจอยู่ในรูปแบบสัญลักษณ์พิเศษ รูปภาพ หรือแผนภาพ ที่เป็นที่รู้จัก ไม่ควรใช้สัญลักษณ์เฉพาะกลุ่ม หรือนำมาจากงานทางด้านอื่นที่ไม่เกี่ยวข้องกัน

2.1.1.8 การกำหนดโครงสร้างแบบรูป

โครงสร้างแบบรูป คือ การสรุปภาพรวมของแบบรูปที่สร้างขึ้นในกรณีที่แบบรูปที่ออกแบบมีความเป็นระบบ ซึ่งมีการทำงานร่วมกันของแบบรูปมากกว่า 1 แบบรูป โดยมีแบบรูปที่มีความสัมพันธ์อ้างอิงถึงกัน ควรสร้างส่วนสรุปแบบรูปเพื่อช่วยผู้ใช้งานแบบรูปทำความเข้าใจและเลือกใช้แบบรูปได้อย่างมีประสิทธิภาพ

การสรุปแบบรูปแบ่งออกเป็น 2 ส่วน คือ

1) สรุปภาษาแบบรูป

การสรุปภาษาแบบรูป คือ การสรุปลักษณะองค์ประกอบและการใช้งานของแต่ละแบบรูปเพื่ออธิบายความหมายขององค์ประกอบส่วนต่างๆ ซึ่งจะช่วยให้เข้าใจแบบรูปและสามารถใช้แบบรูปตามองค์ประกอบที่มีได้อย่างถูกต้อง

2) สรุปปัญหาและผลเฉลย

การสรุปปัญหาและผลเฉลย คือ การสรุปในภาพรวมของแบบรูป เมื่อแบบรูปในระบบมีจำนวนมากขึ้น จำเป็นต้องจัดทำสรุปและแจกแจงแบบรูปเพื่อให้เลือกใช้ง่ายขึ้น

2.1.2 วิศวกรรมความต้องการซอฟต์แวร์ (Software Requirements Engineering)

วิศวกรรมความต้องการซอฟต์แวร์เป็นส่วนหนึ่งของวิศวกรรมซอฟต์แวร์ ถูกกำหนดขึ้นในช่วงการเริ่มต้นของกระบวนการพัฒนาซอฟต์แวร์ โดยมีจุดหมายในการให้ได้มาซึ่งความต้องการด้านซอฟต์แวร์ที่ถูกต้องและชัดเจนเพื่อนำไปใช้ในการกำหนดระบบที่จะทำการพัฒนา 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 ความต้องการที่เก็บมามีความสอดคล้องกันหรือไม่ 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 ที่ไม่อาจยอมรับความผิดพลาดในตัวระบบได้ตลอดจนให้ความสำคัญกับทุกเหตุการณ์ที่อาจเกิดขึ้น

กระบวนการพัฒนาระบบปลอดภัยเชิงวิกฤต [21] มีความแตกต่างจากกระบวนการพัฒนาระบบทั่วๆไปค่อนข้างมาก 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 ลักษณะได้มีการพัฒนาระบบปลอดภัยเชิงวิกฤติในรูปแบบของระบบฝังตัว ดังนั้นปัญหาความผิดพลาดในระบบปลอดภัยเชิงวิกฤตจึงไม่ได้พิจารณาที่ความผิดพลาดของซอฟต์แวร์อย่างเดียว แต่รวมไปถึงความผิดพลาดในตัวอุปกรณ์ฮาร์ดแวร์ ลักษณะการใช้งานที่ผิดๆ ความผิดพลาดของมนุษย์ เช่น ความประมาท การหลงลืม รวมทั้งภัยธรรมชาติที่เกิดขึ้นด้วย

อย่างไรก็ดีในการพัฒนาระบบปลอดภัยเชิงวิกฤตโดยเฉพาะในส่วนที่เกี่ยวข้องกับวิศวกรรมความต้องการซอฟต์แวร์ จะทำการพิจารณาสิ่งที่จะเป็นอันตรายต่อชีวิตโดยละเอียด ซึ่งมักจะใช้การวิเคราะห์อันตรายเป็นหลัก ระหว่างการพัฒนาซอฟต์แวร์จะทำการระมัดระวังผลที่เกิดตามความเสี่ยงเหล่านี้ด้วย โดยในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์สำหรับระบบปลอดภัยเชิงวิกฤตยังมีจุดประสงค์ที่สำคัญอีกอย่าง คือ การให้ได้มาซึ่งข้อกำหนดซอฟต์แวร์ที่สามารถป้องกัน ตรวจจับ หรือตอบสนองต่อเหตุการณ์ที่เป็นอันตรายได้

การวิเคราะห์อันตรายเป็นขั้นตอนที่ต้องทำตลอดการพัฒนาระบบปลอดภัยเชิงวิกฤต และเป็นส่วนสำคัญในการวางแผนและกำหนดลักษณะของระบบในการเริ่มต้นการพัฒนา ซึ่งเกี่ยวข้องและใกล้ชิดกับกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ ดังนั้นการวิเคราะห์อันตรายจึงเป็นส่วนหนึ่งที่เกี่ยวข้องโดยตรงกับงานวิจัยนี้

การวิเคราะห์อันตรายมีด้วยกันหลายวิธีขึ้นกับจุดประสงค์ในการนำไปใช้ และอาจมีการใช้หลายๆ วิธีร่วมกัน ตัวอย่างของวิธีการในการวิเคราะห์อันตราย ได้แก่

1) การวิเคราะห์ต้นไม้ความผิดพร่อง (Fault Tree Analysis)
2) การวิเคราะห์ต้นไม้เหตุการณ์ (Event Tree Analysis)
3) การวิเคราะห์อันตรายและการดำเนินการ (Hazards and Operability Analysis)
4) การวิเคราะห์หนทางขัดข้องและผลกระทบ (Failure Modes and Effects Analysis)
5) การวิเคราะห์อันตรายด้วยสเตทแมชชีน (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)

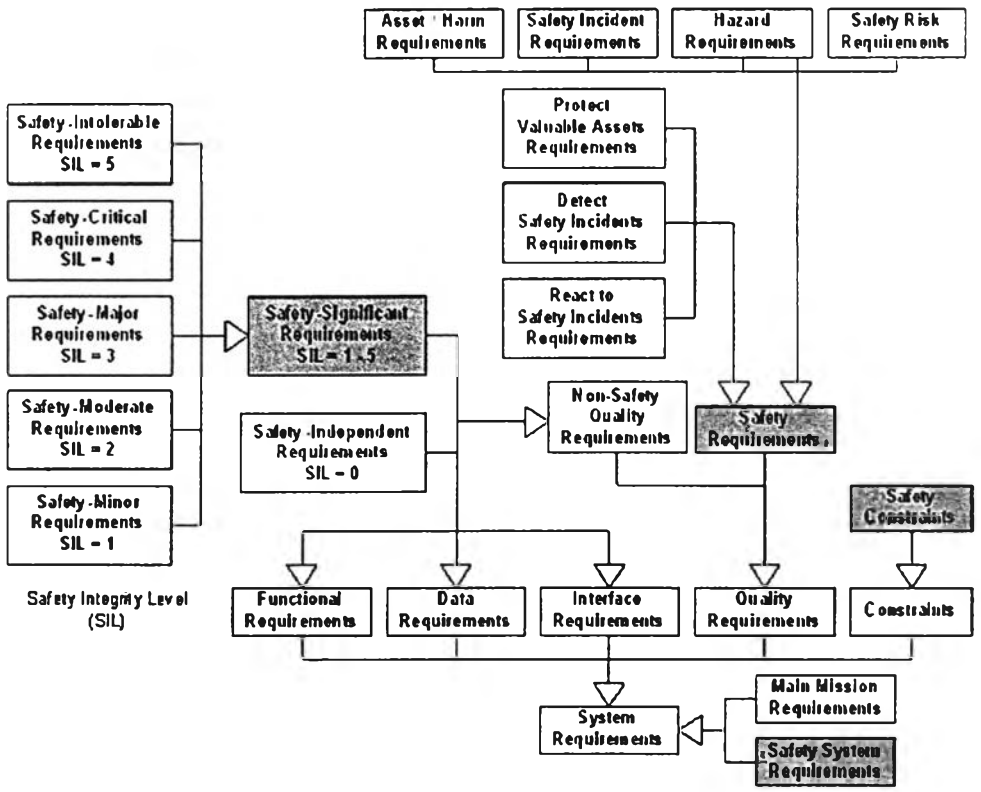
ความต้องการที่เกี่ยวข้องกับความปลอดภัย คือ ความต้องการในระบบปลอดภัยเชิงวิกฤต ซึ่งครอบคลุมทั้งส่วนของฮาร์ดแวร์ ซอฟต์แวร์ และการปฏิบัติงาน โดยเป็นความต้องการที่เกี่ยวข้องกับลักษณะความปลอดภัย ซึ่งเป็นตัวกำหนดระบบให้ปลอดภัยจากอันตรายและกำหนดหน้าที่และคุณภาพของซอฟต์แวร์ และได้ให้ความสำคัญกับอันตรายหรืออุบัติเหตุที่อาจเกิดขึ้นเป็นพิเศษ

นอกจากความต้องการทางผลิตผล (Product Requirements) ที่ครอบคลุมลักษณะซึ่งถือเป็นการกำหนดคุณสมบัติของระบบที่ได้ ความต้องการของระบบปลอดภัยเชิงวิกฤตยังให้ความสำคัญกับความต้องการทางกระบวนการ (Process Requirements) เช่น การมีเงื่อนไขบังคับให้ต้องทำการวิเคราะห์อันตรายเป็นต้น จะเห็นว่าความต้องการสำหรับระบบปลอดภัยเชิงวิกฤตมีขอบเขตที่กว้างและเจาะจงกว่าระบบทั่วๆ ไป ซึ่งมาตรฐาน [21] และคู่มือ [11] ในการพัฒนาระบบปลอดภัยเชิงวิกฤติก็มีการกำหนดรายละเอียดที่เกี่ยวข้องกับความต้องการที่ต้องมีความชัดเจนและครอบคลุมทั้งความต้องการทางผลิตผลและความต้องการทางกระบวนการ

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

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

การแบ่งประเภทของความต้องการที่เกี่ยวข้องกับความปลอดภัย ได้อาศัยการแบ่งประเภทตามแนวทางของ Donald Firesmiith [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 ระบบรถไฟฟ้าต้องป้องกันไม่ให้รถไฟวิ่งในระดับความเร็วที่เป็นอันตราย เป็นต้น โดยความต้องการนี้พิจารณาไปยังผลิตผลที่ได้ ถือได้ว่าเป็นความต้องการที่ไม่ใช้หน้าที่การทำงาน เนื่องจากไม่ได้เป็นการระบุหน้าที่ของระบบโดยตรง และจัดได้ว่าเป็นความต้องการเพื่อความวางใจ (Dependability Requirements) [25]

ความต้องการความปลอดภัยสามารถแบ่งตามลักษณะของความต้องการได้เป็น 3 ประเภท ดังนี้

1) การป้องกัน (Protection) คือ ความต้องการให้ระบบสามารถป้องกันความผิดพลาดได้ เช่น ระบบรถไฟฟ้าต้องป้องกันไม่ให้เกิดไฟฟ้าดับเกิน 30 นาที เป็นต้น
2) การตรวจหา (Detection) คือ ความต้องการในการตรวจพบความผิดพลาดที่เป็นความผิดพลาดที่คาดการหรือจดจำได้ เช่น ระบบรถไฟฟ้ามีความเป็นไปได้ที่เกิดปัญหาไฟฟ้าดับ และเมื่อเกิดเหตุการณ์นี้ขึ้นระบบจะต้องตรวจพบและสามารถดำเนินแก้ไขได้ทันท่วงที เป็นต้น
3) การตอบสนอง (Reaction) คือ การรายงานความผิดพลาดที่เกินขึ้น เป็นการเตือนหรือส่งผลออกมา เช่น เมื่อไฟฟ้าดับรถไฟฟ้าจะมีการแจ้งให้ผู้โดยสารทราบ และระบบช่วยเหลือจะดำเนินการโดยอัตโนมัติ เป็นต้น

จะเห็นว่าลักษณะการแบ่งความต้องการข้างต้น จะพิจารณาไปที่วิธีการที่จะทำให้เกิดความเชื่อมั่นในความปลอดภัยของระบบ และจากรูปที่ 2.2 นอกจากการแบ่งตามลักษณะการทำให้เกิดความเชื่อมั่น ยังสามารถแบ่งประเภทตามลักษณะมูลเหตุที่จำเป็นต้องทำการกำหนดความต้องการด้วย โดยความต้องการในส่วนนี้จัดได้ว่าเป็นความต้องการที่ไม่ใช่หน้าที่

2.1.4.2 ความต้องการนัยปลอดภัย (Safety-Significant Requirements)

ความต้องการนัยปลอดภัย คือ การประเมินความต้องการจากอันตรายที่เกิดขึ้น ซึ่งได้จากการวิเคราะห์อันตราย ซึ่งจะทำการวิเคราะห์อันตรายที่เกินทั้งในส่วนของ 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 ในกรณีที่ผู้ใช้งานระบบใช้งานผิดหรือไม่ระมัดระวังเพียงพอจนเกิดความผิดพลาดในการทำงานขึ้น 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) หายนะ (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 การประเมินค่าความเสี่ยงอุบัติภัย [26]

SEVERITY PROBABILITY	Catastrophic	Critical	Marginal	Negligible
Frequenct	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 หมวดหมู่การเสี่ยงอุบัติภัย [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 และถือเป็นความจำเป็นขั้นมูลฐานของตัวระบบปลอดภัยเชิงวิกฤต

ความต้องการระบบปลอดภัยถือได้ว่าเป็นเอกลักษณ์ของระบบปลอดภัยเชิงวิกฤต และมีความสำคัญสูงสุดในการพัฒนาระบบ ความต้องการระบบปลอดภัยไม่ได้กำหนดลักษณะในระดับผลผลิตที่ได้ออกมาเท่านั้น แต่ความต้องการระบบปลอดภัยจะทำการพิจารณาไปถึงกระบวนการที่จะได้ผลิตผลออกมาด้วย ไม่ว่าจะเป็นเรื่องของการกำหนดและควบคุมกระบวนการในการพัฒนา การรับประกันและดูแลระบบหลังจากการพัฒนาระบบเสร็จสิ้น เช่น การพัฒนาซอฟต์แวร์จะต้องทำการประเมินความปลอดภัย ต้องทำการอบรมพนักงานให้สามารถใช้งานระบบได้อย่างปลอดภัยหลังการส่งมอบ เป็นต้น

ขอบเขตของความต้องการระบบปลอดภัยให้ข้อจำกัดในวงกว้าง ตัวอย่างเช่น ความต้องการระบบปลอดภัยของระบบขึ้นบินและลงจอดเครื่องบิน ซึ่งบางส่วนมีการกำหนดว่าเครื่องบินต้องขึ้นหรือลงไม่ออกนอกทางวิ่ง ซึ่งเป็นความต้องการที่เกี่ยวข้องกับส่วนต่างๆของระบบ 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 ในองค์กรมีการกำหนดมาตรฐานซอฟต์แวร์ของตนว่าต้องมีความผิดพลาดในโค้ดน้อยกว่า 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 โดยตัดองค์ประกอบของแบบรูปในส่วนของ การทำให้เกิดผล และ ตัวอย่างโปรแกรมออกไป และเพิ่มข้อจำกัด (Constraints) พฤติกรรม (Behavior) และ แบบรูปการออกแบบ (Design Patterns) เข้ามา ซึ่งนำเสนอโดยใช้ยูเอ็มแอล (Unified Modeling Language: UML) พร้อมกับข้อความอธิบาย

ในงานวิจัยได้ทำการแบ่งประเภทของตัวแบบรูปความต้องการสำหรับระบบฝังตัวออกเป็นกลุ่มระบบทำงานย่อยส่วนต่างๆ และนำเสนอตัวอย่างแบบรูปความต้องการของระบบฝังตัว 2 ตัวอย่างด้วยกัน คือ

1) ตัวกระตุ้น-ตัวตรวจจับ (Actuator-Sensor) ซึ่งเป็นแบบรูปประเภทโครงสร้าง

2) การจัดการผลของความผิดพลาด (Fault Handler) ซึ่งเป็นแบบรูปประเภทพฤติกรรม

ตัวอย่างทั้ง 2 ที่แสดงในงานวิจัยนี้ทำให้เห็นภาพรวมของการนำแบบรูปไปใช้งาน ซึ่งเป็นแนวทางที่นำความต้องการที่ผ่านการวิเคราะห์ออกแบบมาใช้ใหม่

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 โดยพยายามนำเอาแบบรูปไปใช้ในการแก้ปัญหากับลักษณะของกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ โดยเฉพาะกับกระบวนการเก็บรวบรวมความต้องการ

งานวิจัยนี้สร้างแบบรูปโดยอาศัยมุมมองใน 2 ส่วน คือ

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

โดยงานวิจัยนี้พยายามชี้ให้เห็นว่าสถานการณ์จะนำไปสู่การแก้ปัญหาที่เกิดขึ้นด้วยแบบรูป หรือนำไปสู่สถานการณ์อื่นๆ ที่จะเป็นผลเฉลยให้ได้ 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 ซึ่งสนับสนุนกระบวนการวิศวกรรมความต้องการในโครงการพัฒนาซอฟต์แวร์อันได้แก่ 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) องค์ประกอบบางตัวของแบบรูปในงานวิจัยนี้เป็นองค์ประกอบที่เหมาะในการนำมาใช้ในแบบรูปความต้องการ ซึ่งได้แก่ จุดประสงค์ (Objective) ขอบเขตของโปรแกรมประยุกต์(Application Area) และ ประสบการณ์ (Experience)

2) แบบรูปที่นำเสนอในงานวิจัยนี้นำเสนอสำหรับกระบวนการและเทคนิคที่ใช้ในกระบวนการวิศวกรรมความต้องการซอฟต์แวร์ ซึ่งเป็นแนวทางที่จะนำมาประยุกต์ใช้กับการสร้างแบบรูปกระบวนการความต้องการ

แนวคิดของงานวิจัยนี้พิจารณาไปที่กระบวนการวิศวกรรมความต้องการซอฟต์แวร์เป็นหลัก ไม่ได้ให้ข้อสังเกตในการนำไปใช้กับโดเมนของระบบที่ต่างกันออกไป ตัวแบบรูปจึงเสมือนการนำเสนอแนวทาง 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. Johnton [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