บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันระบบซอฟต์แวร์ได้มีการขยายตัวทั้งในเชิงขนาดและความซับซ้อนมากยิ่งขึ้น ซึ่งในระบบเหล่านั้นต่างก็ต้องการซอฟต์แวร์ที่มีความเชื่อถือได้โดยเฉพาะในระบบที่มีความเสี่ยงสูง แต่การสร้างซอฟต์แวร์ให้มีความเชื่อถือได้นั้นเป็นปัญหาที่ยุ่งยากของอุตสาหกรรมการผลิตซอฟต์แวร์ และจะเป็นการล่าช้าเกินไปหากจะรอให้พบข้อผิดพลาดของการใช้งานโดยผู้ใช้หลังจากซอฟต์แวร์ได้ออกจำหน่ายไปแล้ว การตรวจสอบความถูกต้อง (verification) เป็นวินัยอย่างหนึ่งของกรรมวิธีการพัฒนาซอฟต์แวร์ที่สำคัญที่สามารถช่วยทำให้การพัฒนาซอฟต์แวร์มีคุณภาพมากยิ่งขึ้น

ในการอธิบายความต้องการของระบบนั้น ภาษาธรรมชาติซึ่งเป็นภาษาที่เราใช้ในการสื่อสารกันในชีวิตประจำวันมักจะถูกนำมาใช้เป็นภาษาหลัก ดังนั้นความเข้าใจถึงความต้องการของระบบจึงขึ้นอยู่กับความหมายของคำและรูปประโยคที่นำมาใช้ในการอธิบาย จากความซับซ้อนและกำกวมของภาษาธรรมชาติในความต้องการของระบบดังกล่าวทำให้เป็นการยากที่เราจะทำความเข้าใจและตรวจสอบความถูกต้อง จนอาจก่อให้เกิดความคลาดเคลื่อนในการสื่อสารระหว่างผู้ร่วมงาน โดยเฉพาะอย่างยิ่งหากระบบมีการขยายตัวและมีความซับซ้อนมากยิ่งขึ้นจะทำให้มีผลต่อประสิทธิภาพของการพัฒนา และด้วยความเป็นอรูปนัย (informal) ของความต้องการระบบนี้เอง ขั้นตอนที่สามารถดำเนินการได้โดยอัตโนมัติก็เป็นไปได้ยากด้วยเช่นกัน ดังนั้นความต้องการทางด้านวิธีรูปนัย (formal methods) จึงเพิ่มมากยิ่งขึ้นโดยเฉพาะในส่วนของการพัฒนาระบบการประมวลผลแบบกระจาย (distributed system) ระบบที่มีการทำงานแบบพร้อมกัน (concurrent system) และระบบการทำงานแบบเรียลไทม์ (real-time system) ซึ่งการนำเอาวิธีรูปนัยเข้ามาใช้จะช่วยเพิ่มความชัดเจนของการอธิบาย ทำให้ความสามารถในการเข้าถึงและเข้าใจความต้องการของระบบมีมากขึ้น อีกทั้งทำให้เราสามารถสร้างและเพิ่มประสิทธิภาพของขั้นตอนการตรวจสอบแบบอัตโนมัติได้ด้วย ดังนั้นการใช้งานวิธีรูปนัยจึงเป็นส่วนสำคัญส่วนหนึ่งที่คาดหมายว่าจะนำไปสู่การพัฒนาซอฟต์แวร์ที่มีประสิทธิภาพและมีความเชื่อถือได้

แผนภาพยูเอ็มแอล (UML: Unified Modeling Language) [1, 11] ซึ่งเป็นมาตรฐานที่ดูแลและควบคุมโดยโอเอ็มจี (OMG: Object Management Group) เป็นภาษาที่นำมาใช้ในการวิเคราะห์และออกแบบพัฒนาซอฟต์แวร์ในระบบเชิงวัตถุ โดยในปัจจุบันนี้ได้มีการนำไปใช้งานกันอย่างแพร่หลายในอุตสาหกรรมการพัฒนาซอฟต์แวร์ แผนภาพยูเอ็มแอลนี้เองประกอบด้วย

แผนภาพทั้งหมด 9 แผนภาพสำหรับอธิบายโครงสร้างและพฤติกรรมของระบบในมุมมองต่างๆ กัน โดยจะช่วยให้ผู้อ่านในแต่ละกลุ่มสามารถเข้าใจถึงตัวระบบได้ง่ายยิ่งขึ้น

แผนภาพสถานะ (state diagram) [4] เป็นหนึ่งในแผนภาพยูเอ็มแอลที่มีการใช้งานมากในการอธิบายการเปลี่ยนแปลงพฤติกรรมของระบบ โดยแต่ละแผนภาพจะอธิบายการเปลี่ยนแปลงพฤติกรรมของวัตถุเมื่อมีเหตุการณ์ต่างๆ เกิดขึ้น แต่เนื่องจากความเป็นกึ่งรูปนัย (semi-formal) ของตัวแผนภาพสถานะนั้น ทำให้การตรวจสอบและวิเคราะห์ด้วยกระบวนการทางรูปนัยนั้นเป็นไปได้ยาก ดังนั้นในงานวิจัยนี้จึงมีแนวความคิดที่จะเสนอวิธีการแปลงแผนภาพสถานะไปเป็นภาษารูปนัย (formal language) เพื่อรองรับและเพิ่มความสามารถในการตรวจสอบและวิเคราะห์แผนภาพ งานวิจัยชิ้นนี้ให้ความสนใจในคุณสมบัติการเปลี่ยนแปลงพฤติกรรมของระบบ (dynamic behavior) และการทำงานแบบพร้อมกัน (concurrent) ดังนั้นภาษารูปนัยที่นำมาใช้เป็นภาษาเป้าหมายจึงต้องมีความสามารถรองรับการอธิบายคุณสมบัติดังกล่าวเป็นอย่างดี ภาษาที่ได้เลือกมาใช้เป็นภาษาที่อธิบายลำดับการเกิดขึ้นของกระบวนการต่างๆ ซึ่งในที่นี้ คือ ซีอาร์อี (CRE: Concurrent Regular Expression) [3] และไพแคลคูลัส (π-Calculus) [7]

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อกำหนดขั้นตอนและกฎในการแปลงแผนภาพสถานะไปเป็น ซีอาร์อี และไพแคลคูลัส

1.2.2 เพื่อสร้างกฎในการตรวจสอบความสัมพันธ์ระหว่างแผนภาพสถานะ

1.3 ขอบเขตของการวิจัย

1.3.1 เพื่อกำหนดขั้นตอนและกฎในการแปลงแผนภาพสถานะไปเป็น ซีอาร์อี และไพแคลคูลัส โดยจะต้องได้ขั้นตอนและกฎที่มีความสามารถดังนี้

1.3.1.1 สามารถแปลงองค์ประกอบในการสร้างแผนภาพสถานะดังต่อไปนี้

1) สถานะทั่วไป (simple state) ซึ่งครอบคลุมถึงเหตุการณ์การเข้าสู่สถานะ (entry) การออกจากสถานะ (exit) กิจกรรมที่ต้องทำ (do activity) และการเปลี่ยนแปลงภายใน (internal transition)
2) สถานะประกอบ (composite state) ซึ่งครอบคลุมทั้ง สถานะย่อยแบบต่อเนื่อง (sequential substates) และสถานะย่อยแบบทำงานพร้อมกัน (concurrent substates) เป็นส่วนประกอบภายใน

3) สถานะซิง (synch state)

4) สถานะเริ่มต้นและสิ้นสุด (initial & final pseudostate)

5) สถานะประวัติ (history pseudostate) (เฉพาะไพแคลคูลัส)

6) สถานะแยกและเชื่อมสำหรับการทำงานพร้อมกัน (fork & join pseudostate)

7) สถานะตัวต่อและตัวเลือก (junction & choice Point)

8) เส้นการเปลี่ยนแปลง (transition) ซึ่งครอบคลุมทั้งส่วนของเหตุการณ์ (event) เงื่อนไข (condition) และ การกระทำ (actions)

9) แผนภาพย่อย (submachine) (เฉพาะไพแคลคูลัส)

1.3.1.2 สามารถแปลงการเชื่อมต่อระหว่างเส้นการเปลี่ยนแปลงและสถานะในแบบต่างๆ ดังนี้

1) สถานะที่มีเส้นการเปลี่ยนแปลงเชื่อมต่อไปยังอีกสถานะหนึ่งเพียงหนึ่งเส้น

2) สถานะที่มีเส้นการเปลี่ยนแปลงเชื่อมต่อไปยังสถานะอื่นๆ หลายเส้น

3) สถานะที่มีเส้นการเปลี่ยนแปลงที่เชื่อมต่อไปยังสถานะย่อยที่อยู่ภายในสถานะประกอบ ซึ่งครอบคลุมทั้งสถานะย่อยแบบต่อเนื่องและสถานย่อยแบบทำงานพร้อมกัน (เฉพาะไพแคลคูลัส)

4) สถานะที่มีเส้นการเปลี่ยนแปลงจากสถานะย่อยผ่านสถานะประกอบไปยังสถานะอื่นๆ (เฉพาะไพแคลคูลัส)

1.3.2 เพื่อสร้างกฎในการตรวจสอบความสัมพันธ์ระหว่างแผนภาพสถานะ โดยแบ่งออกเป็น 2 ส่วน คือ

1.3.2.1 การตรวจสอบความเท่าเทียมกันในด้านพฤติกรรมของวัตถุ คือ การตรวจสอบพฤติกรรมที่วัตถุตอบสนองต่อระบบว่าเท่าเทียมกันหรือไม่ โดยจะเกิดขึ้นมากในกรณีของการพัฒนาซอฟต์แวร์แบบอาศัยส่วนประกอบ (component based software development) ซึ่งจะพิจารณาการแทนที่วัตถุเก่าด้วยวัตถุใหม่ที่มีคุณสมบัติการทำงานเหมือนกัน หรือในกรณีของการออกแบบระบบในแต่ละระดับของรายละเอียด โดยสามารถเปรียบเทียบคุณสมบัติของระบบในระดับรายละเอียดย่อยและระดับนามธรรมว่าตรงกันได้

1.3.2.2 การตรวจสอบพฤติกรรมการทำงานของวัตถุต่างๆ ที่เกิดขึ้นเมื่อทำงานร่วมกับวัตถุอื่นภายในระบบ โดยสามารถตรวจสอบถึงพฤติกรรมที่ได้รับการออกแบบไว้ของวัตถุแต่ไม่ได้ถูกใช้งานเมื่อทำงานร่วมกับวัตถุอื่นภายในระบบได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

ผลของงานวิจัยนี้จะทำให้ได้ขั้นตอน และกฎในการแปลงแผนภาพสถานะไปเป็นการบรรยายในภาษารูปนัย คือ ซีอาร์อี และไพแคลคูลัส ทำให้สามารถนำวิธีการตรวจสอบและพิสูจน์ทางรูปนัยมาช่วยในขั้นตอนพัฒนาซอฟต์แวร์ได้ ส่งผลให้สามารถสร้างซอฟต์แวร์ที่มีคุณภาพได้มากยิ่งขึ้น

1.5 วิธีดำเนินการวิจัย

1.5.1 ศึกษาและทำความเข้าในการออกแบบซอฟต์แวร์ด้วยแผนภาพสถานะ รวมถึงศึกษาตัวอย่างการเขียนแผนภาพสถานะที่เป็นที่นิยม

1.5.2 ศึกษางานวิจัยที่เกี่ยวข้องกับการแปลงแผนภาพสถานะไปเป็นข้อกำหนดแบบรูปนัย

1.5.3 ศึกษาข้อกำหนดรูปนัยที่เหมาะสมเพื่อนำมาใช้เป็นภาษาเป้าหมายของการแปลงแผนภาพสถานะ

1.5.4 ออกแบบขั้นตอน และกฎการแปลงจากแผนภาพสถานะไปเป็นข้อกำหนดรูปนัยที่กำหนด

1.5.5 ออกแบบวิธีการในการตรวจสอบความสัมพันธ์ของแผนภาพจากข้อกำหนดรูปนัยที่ได้จากขั้นตอนการแปลง

1.5.6 สรุปผลการวิจัยและจัดทำรายงานวิทยานิพนธ์