หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ผลกระทบวงคลื่น

ผลกระทบวงคลื่น คือปรากฏการณ์หนึ่งที่เกิดขึ้นเมื่อมีการเปลี่ยนแปลงส่วนหนึ่งส่วนใดของโปรแกรมซึ่งจะส่งผลต่อส่วนอื่นๆ ของโปรแกรม [3] ตัวอย่างเช่น สมมติให้โปรแกรม P ประกอบไปด้วยส่วนย่อย 6 ส่วน คือ A B C D E และ F เมื่อมีการเปลี่ยนแปลงที่ส่วนย่อย A แล้วส่งผลให้ส่วนย่อย B และ C เกิดการเปลี่ยนแปลงตาม เช่น ไม่สามารถทำงานได้ หรือประสิทธิภาพของส่วนย่อยของโปรแกรมมีค่าลดลง การเกิดผลกระทบที่กระจายออกจากการเปลี่ยนแปลง ณ จุดใดจุดหนึ่งนี้ เรียกว่าผลกระทบวงคลื่น

ผลกระทบวงคลื่นสามารถแบ่งออกได้หลายแบบด้วยกัน เช่น งานวิจัยของ Yau และ Collofello [5] ได้แบ่งออกเป็น 2 แบบด้วยกัน คือ

1) ผลกระทบวงคลื่นเชิงตรรกะ (Logical Ripple Effect)
2) ผลกระทบวงคลื่นเชิงประสิทธิภาพ (Performance Ripple Effect)

นอกจากนี้ได้มีนักวิจัยทำการแบ่งผลกระทบวงคลื่นออกเป็นประเภทต่างๆ อีกดังนี้ [6]

1) ผลกระทบวงคลื่นเชิงการเขียนโปรแกรม (Coding Ripple Effect)
2) ผลกระทบวงคลื่นเชิงข้อมูล (Data Ripple Effect)
3) ผลกระทบวงคลื่นเชิงเอกสาร (Documentation Ripple Effect)
4) ผลกระทบวงคลื่นเชิงความต้องการ (Requirement Ripple Effect)
5) ผลกระทบวงคลื่นเชิงส่วนเชื่อมต่อ (Interface Ripple Effect)
6) ผลกระทบวงคลื่นเชิงสภาวะแวดล้อม (Environment Ripple Effect)
7) ผลกระทบวงคลื่นเชิงการจัดการ (Management Ripple Effect)

ในวิทยานิพนธ์นี้จะศึกษาโดยเน้นผลกระทบวงคลื่นเชิงตรรกะเป็นหลัก และมีนิยามคือ ผลกระทบจากการเปลี่ยนแปลงคลาสใดคลาสหนึ่งของโปรแกรมต่อการทำงานของคลาสอื่นๆ

2.1.2 ความเสถียร

ความเสถียรของซอฟต์แวร์ คือค่าความต้านทานของซอฟต์แวร์ต่อผลกระทบจากการเปลี่ยนแปลงที่เกิดขึ้นแก่ซอฟต์แวร์ [3] หรืออาจนิยามเป็นค่าความต้านทานของซอฟต์แวร์ต่อผลกระทบวงคลื่น ความเสถียรของซอฟต์แวร์สามารถแบ่งออกได้เป็น 2 แบบ ดังนี้

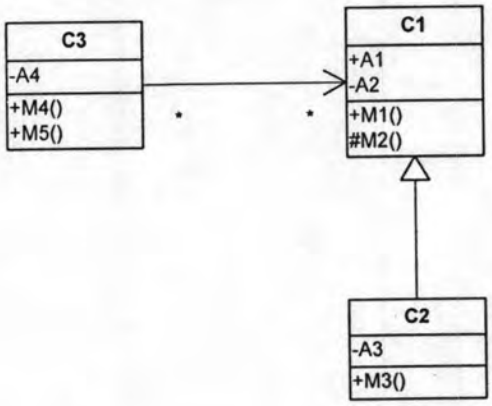
1) ความเสถียรเชิงตรรกะ
2) ความเสถียรเชิงประสิทธิภาพ

ความเสถียรเชิงตรรกะคือ โอกาสในการไม่ได้รับผลกระทบที่ส่งผลให้การทำงานผิดพลาดหรือไม่สามารถทำงานได้ ตัวอย่างของผลกระทบจากการเปลี่ยนแปลงที่ส่งผลเชิงตรรกะ เช่น จากรูปที่ 2.1 ให้เมทธอด M4() ของคลาส C3 มีการอ้างถึง เมทธอด M1() ของคลาส C1 และเมื่อมีการเปลี่ยนแปลงซิกเนเจอร์ของเมทธอด M1() แล้วเมทธอด M4() ที่ไม่ได้ทำการเปลี่ยนแปลงตามย่อมไม่สามารถเรียกใช้งานเมทธอด M1() ได้ เป็นต้น

ค่าความเสถียรเชิงตรรกะของคลาสใดๆ หาได้จาก จำนวนครั้งที่คลาสจะไม่ได้รับผลกระทบเชิงตรรกะ หรือผลกระทบที่ทำให้การทำงานผิดพลาดจากจำนวนการเปลี่ยนแปลงทั้งหมดของซอฟต์แวร์ [7] โดยความเสถียรเชิงตรรกะของคลาสใดๆ สามารถใช้อธิบายโอกาสในการไม่ได้รับผลกระทบเชิงตรรกะที่เกิดจากการเปลี่ยนแปลงซอฟต์แวร์ของคลาสได้ เช่น หากมีคลาสหนึ่งมีความเสถียรเชิงตรรกะเท่ากับ 0.68 หมายความว่าหากเกิดการเปลี่ยนแปลงขึ้นในซอฟต์แวร์ โอกาสที่คลาสนี้จะไม่ได้รับผลกระทบเชิงตรรกะมีค่าเท่ากับ 68% หรือในการเปลี่ยนแปลง 100 ครั้ง คลาสนี้จะไม่ได้รับผลกระทบเชิงตรรกะและยังคงสามารถทำงานได้เหมือนเดิม 68 ครั้ง เป็นต้น

ความเสถียรเชิงประสิทธิภาพคือ โอกาสในการไม่ได้รับผลกระทบที่ส่งผลให้ประสิทธิภาพในการทำงานเปลี่ยนไป ตัวอย่างของผลกระทบจากการเปลี่ยนแปลงที่ส่งผลต่อประสิทธิภาพ เช่น จากรูปที่ 2.1 ให้เดิมคลาส C3 ใช้เวลาในการประมวลผล 0.3 วินาที แต่เมื่อทำการเปลี่ยนแปลงเมทธอด M1() ซึ่งเป็นเมทธอดที่เมทธอด M4() ของคลาส C3 อ้างถึง แล้วส่งผลให้เวลาที่คลาส C3 ใช้ในการประมวลผลเพิ่มขึ้นเป็น 0.6 วินาที เป็นต้น เช่นเดียวกับค่าความเสถียรเชิงตรรกะ ค่าความเสถียรเชิงประสิทธิภาพของคลาสใดๆ ใช้อธิบายโอกาสในการไม่ได้รับผลกระทบเชิงประสิทธิภาพที่เกิดจากการเปลี่ยนแปลงซอฟต์แวร์ซึ่งส่งผลให้ประสิทธิภาพในการทำงานของคลาสนั้นๆ เปลี่ยนแปลงไป อย่างไรก็ตามการหาค่าความเสถียรเชิงประสิทธิภาพมีความซับซ้อนสูง ทั้งนี้เนื่องจากการหาผลกระทบเชิงประสิทธิภาพทำได้ยาก เพราะต้องกำหนดถึงประสิทธิภาพที่สนใจ และวิเคราะห์ถึงการเปลี่ยนแปลงว่าอาจเกิดผลกระทบกับประสิทธิภาพใดบ้าง นอกจากนี้

การหาค่าความเสถียรเชิงประสิทธิภาพต้องอาศัยการวิเคราะห์แบบพลวัต (Dynamic Analysis) ซึ่งเป็นการวิเคราะห์ระหว่างการทำงานจริงของซอฟต์แวร์อีกด้วย



รูปที่ 2.1 ตัวอย่างความสัมพันธ์ระหว่างคลาส

ค่าความเสถียรเชิงตรรกะของคลาสใดๆ คือ ความต้านทานต่อผลกระทบวงคลื่นเชิงตรรกะ โดยค่าผลกระทบวงคลื่นเชิงตรรกะคือ จำนวนครั้งที่คลาสที่สนใจจะได้รับผลกระทบจากการเปลี่ยนแปลงทุกประเภทของทุกๆ คลาสในโปรแกรมหารด้วยจำนวนการเปลี่ยนแปลงที่เป็นไปได้ทั้งหมด ดังสมการ [7]

$$\text{ผลกระทบวงคลื่นเชิงตรรกของคลาส} = \frac{\text{จำนวนครั้งที่คลาสได้รับผลกระทบจากการเปลี่ยนแบลง}}{\text{จำนวนการเปลี่ยนแปลงทั้งหมด}} \quad (1)$$

ดังนั้นความเสถียรเชิงตรรกะคือ จำนวนครั้งที่คลาสที่สนใจไม่ได้รับผลกระทบจากการเปลี่ยนแปลงทุกประเภทของทุกๆ คลาสในโปรแกรมหารด้วยจำนวนการเปลี่ยนแปลงที่เป็นไปได้ทั้งหมด สามารถคำนวณได้ดังสมการ [7]

$$\text{ความเสถียรเชิงตรรกของคลาส} = 1 - \text{ผลกระทบวงคลื่นเชิงตรรกะของคลาส} \quad (2)$$

หรือ

$$\text{ความเสถียรเชิงตรรกของคลาส} = 1 - \frac{\text{จำนวนครั้งที่คลาสได้รับผลกระทบจากการเปลี่ยนแบลง}}{\text{จำนวนการเปลี่ยนแปลงทั้งหมด}} \quad (3)$$

ตัวอย่างในการคำนวณความเสถียรเชิงตรรกะของคลาสใดๆ เป็นดังนี้ สมมติให้ซอฟต์แวร์หนึ่งประกอบด้วยคลาส 5 คลาส C_1 C_2 C_3 C_4 และ C_5 ในการหาค่าความเสถียรเชิงตรรกะของคลาส C_1 ทำได้โดย

1) เลือกคลาสทีละคลาสจากคลาสทั้งหมด
2) จำลองการเปลี่ยนแปลงกับคลาสที่เลือกมาถ้าสามารถทำการเปลี่ยนแปลงได้ให้เพิ่มค่าจำนวนการเปลี่ยนแปลงทั้งหมด
3) ตรวจสอบว่าคลาส C_1 ได้รับผลกระทบจากการเปลี่ยนแปลงหรือไม่ ถ้าได้รับผลกระทบให้เพิ่มค่าจำนวนครั้งที่ C_1 ได้รับผลกระทบ
4) ทำการตรวจสอบจนครบทุกรูปแบบการเปลี่ยนแปลงของคลาสที่เลือกมา โดยสำหรับการเปลี่ยนแปลงในระดับคลาสให้ทำการตรวจสอบทีละแอททริบิวต์และเมทธอด
5) ทำการเลือกคลาสที่เหลือขึ้นมาจนครบทุกคลาส

สมมติให้ผลที่ได้จากการตรวจสอบเป็นดังนี้

- จำนวนการเปลี่ยนแปลงทั้งหมดเป็น 257
- จำนวนครั้งที่คลาส C_1 ได้รับผลกระทบเป็น 154

ค่าผลกระทบวงคลื่นเชิงตรรกะของคลาส C_1 มีค่าเท่ากับ 154 หารด้วย 257 เท่ากับ 0.5992

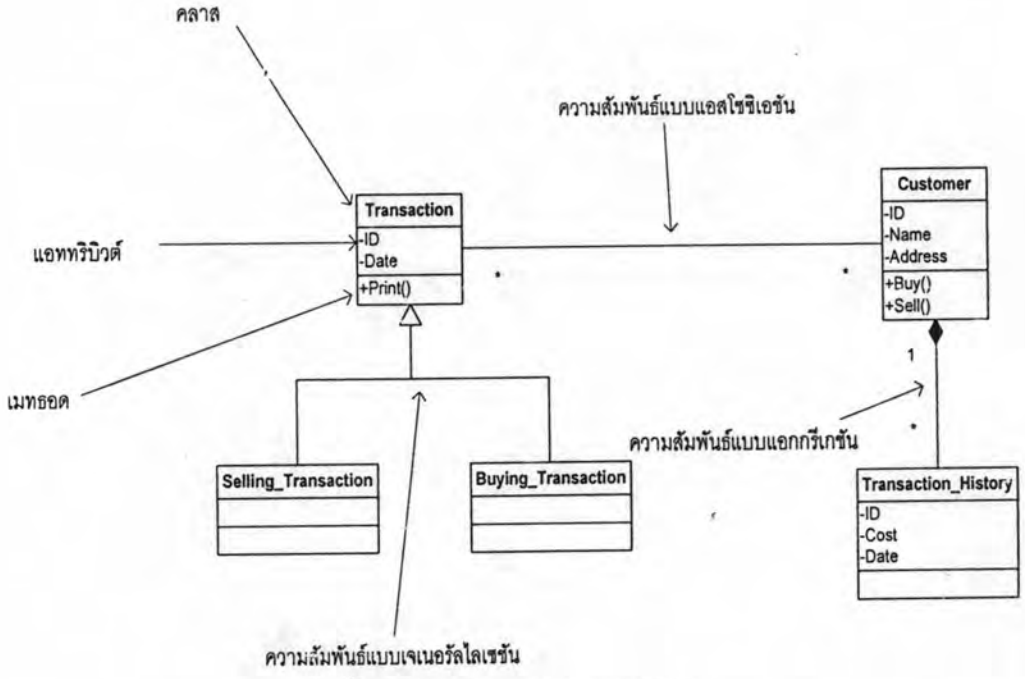
ดังนั้นค่าความเสถียรเชิงตรรกะของคลาส C_1 เท่ากับ 1 – 0.5992 เท่ากับ 0.4008 ซึ่งหมายความว่า เมื่อมีการเปลี่ยนแปลงขึ้นภายในซอฟต์แวร์คลาส C_1 จะมีโอกาสการไม่ได้รับผลกระทบเท่ากับ 40.08% หรืออาจกล่าวได้ว่าหากมีการเปลี่ยนแปลงกับซอฟต์แวร์เป็นจำนวน 100 ครั้ง คลาส C_1 จะไม่ได้รับผลกระทบ 40 ครั้ง

ในการคำนวณค่าความเสถียรเชิงตรรกะต้องทำการคำนวณค่าผลกระทบวงคลื่นเชิงตรรกะก่อนแล้วจึงทำการคำนวณความเสถียรเชิงตรรกะ ทั้งนี้เนื่องจากการคำนวณหาจำนวนครั้งที่คลาสได้รับผลกระทบสามารถกระทำได้ง่ายกว่า สำหรับอัลกอริทึมในการหาค่าความเสถียรเชิงตรรกะของคลาสมีรายละเอียดในภาคผนวก ก

2.1.3 แผนภาพคลาสและแผนภาพซีเควนซ์

แผนภาพคลาส [8]ใช้แสดงรายละเอียดของคลาสและความสัมพันธ์ระหว่างคลาสทั้งหมดของซอฟต์แวร์ซึ่งช่วยในการทำความเข้าใจระบบและการออกแบบระบบโดยรวม โดยคลาส

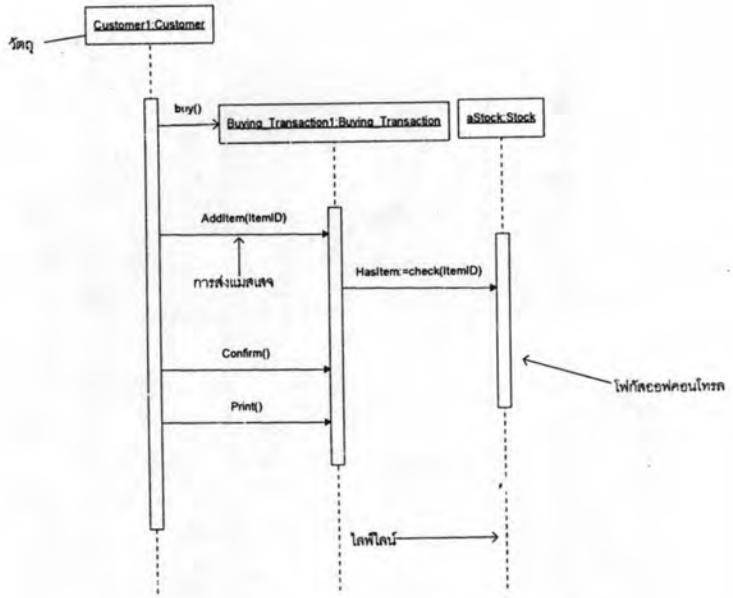
ประกอบด้วย แอททริบิวต์ (attribute) และเมทธอด (method) หรือโอเปอร์เรชัน (operation) และคลาสจะเชื่อมโยงกันด้วยความสัมพันธ์แบบต่างๆ เช่น แอสโซซิเอชัน (association) เป็นการเชื่อมโยงแบบปกติในกรณีที่คลาสมีการติดต่อเรียกใช้ซึ่งกันและกัน แอกกรีเกชัน (aggregation) เป็นการเชื่อมโยงของคลาสที่เป็นส่วนหนึ่งของอีกคลาส และเจเนอรัลไลเซชัน (generalization) เป็นความสัมพันธ์ระหว่างคลาสซึ่งคลาสหนึ่งเป็นคลาสย่อยของอีกคลาสหนึ่ง หรือเป็นความสัมพันธ์ระหว่างคลาสแม่กับคลาสลูก ตัวอย่างแผนภาพคลาสดังแสดงในรูปที่ 2.2



รูปที่ 2.2 ตัวอย่างแผนภาพคลาส [8]

แผนภาพซีเควนซ์ [8] แสดงการโต้ตอบกันระหว่างวัตถุ ซึ่งจะแสดงให้เห็นถึงลำดับของการส่งคำร้องขอ (Message) ถึงกันระหว่างวัตถุต่างๆ โดยปกติแผนภาพซีเควนซ์ใช้แสดงเค้าโครงเรื่อง (Scenario) หนึ่งๆ ของซอฟต์แวร์ และในการใช้งานจริงแผนภาพซีเควนซ์ใช้อธิบายเฉพาะสถานการณ์สำคัญเท่านั้น ทั้งนี้เพราะสำหรับซอฟต์แวร์หนึ่งๆ สามารถมีสถานการณ์ได้มากมาย

แผนภาพซีเควนซ์ประกอบไปด้วยองค์ประกอบหลักๆ ดังนี้ วัตถุ เส้นไลฟ์ไลน์ (lifeline) หรือช่วงที่วัตถุมีชีวิตอยู่ระหว่างการทำงาน และ โฟกัสออฟคอนโทรล (focus of control) หรือช่วงของเวลาที่วัตถุทำการรับส่งคำร้องขอตลอดเส้นไลฟ์ไลน์ ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 ตัวอย่างแผนภาพซีเควนซ์ [8]

วิทยานิพนธ์นี้ต้องการทำการประมาณค่าความเสถียรเชิงตรรกะโดยรวมของซอฟต์แวร์ จึงต้องทราบข้อมูลโดยภาพรวมของระบบในขณะที่แผนภาพซีเควนซ์ให้ข้อมูลได้เพียงบางส่วนเท่านั้น ดังนั้นงานวิจัยนี้จึงเน้นการศึกษาแผนภาพคลาสมากกว่าแผนภาพซีเควนซ์ อย่างไรก็ตามถึงแม้แผนภาพคลาสจะแสดงภาพรวมและรายละเอียดโดยรวมของซอฟต์แวร์ ทว่าหากพิจารณาถึงความละเอียดของข้อมูลจะพบว่ามีความละเอียดน้อย โดยจากแผนภาพคลาสสามารถทราบถึงความสัมพันธ์ระหว่างคลาสแต่ไม่ทราบถึงการเรียกใช้เมทธอดหรือแอททริบิวต์ต่างๆ จึงจำเป็นต้องนำแผนภาพซีเควนซ์มาประกอบเพื่อเพิ่มรายละเอียดและความถูกต้อง

2.1.4 **การวิเคราะห์ความถดถอยเชิงซ้อน** (Multiple Regression analysis)

การวิเคราะห์ความถดถอยเชิงซ้อน [9] เป็นการศึกษาความสัมพันธ์ระหว่างตัวแปรมากกว่า 2 ตัวขึ้นไป (หากเป็นการศึกษาความสัมพันธ์ระหว่างตัวแปร 2 ตัวจะเรียกว่า การวิเคราะห์ความถดถอยอย่างง่าย (Simple Regression Analysis)) ในการวิเคราะห์ความถดถอยจะมีตัวแปร 2 ประเภทด้วยกัน คือ ตัวแปรอิสระ และตัวแปรตาม

ตัวแปรอิสระคือ ตัวแปรที่ทราบค่าและไม่ขึ้นกับตัวแปรใดๆ

ตัวแปรตามคือ ตัวแปรที่ค่าของตัวแปรเปลี่ยนแปลงตามค่าของตัวแปรอิสระอื่นๆ

วัตถุประสงค์ของการวิเคราะห์ความถดถอยมีดังนี้

1) เพื่อศึกษาความสัมพันธ์ระหว่างตัวแปรว่ามีความสัมพันธ์กันมากน้อยเพียงใด

2) ใช้ความสัมพันธ์ที่ได้ในการประมาณค่าตัวแปรตาม เมื่อกำหนดค่าตัวแปรอิสระ

รูปแบบสมการความถดถอยเชิงซ้อนเมื่อมีตัวแปรอิสระ k ตัว ($X_1,X_2,...,X_k$) ที่มีความสัมพันธ์กับตัวแปรตาม Y โดยมีความสัมพันธ์แบบเชิงเส้นจะมีรูปแบบดังนี้

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k$$

โดย Y คือตัวแปรตาม

X_i คือตัวแปรอิสระ เมื่อ $i = 1,2,\ldots,k$

β_0 คือส่วนตัดแกน Y เมื่อกำหนดให้ $X_1 = X_2 = \ldots = X_k = 0$

β_i คือสัมประสิทธิ์ความถดถอยเชิงส่วน (Partial regression coefficient) เมื่อ $i = 1,2,\ldots,k$

และค่าประมาณของ Y คือ

$$\hat{Y} = b_0 + b_1 x + b_2 x_2 + \ldots + b_k x_k + e$$

โดย

$\hat{Y}$ คือค่าประมาณของ Y

x_i คือตัวแปรอิสระของสมการความถดถอย เมื่อ $i = 1,2,\ldots,k$

b_0 คือค่าคงที่

b_i คือค่าประมาณของสัมประสิทธิ์ความถดถอยเชิงส่วน เมื่อ $i = 1,2,\ldots,k$

e คือความคลาดเคลื่อนอย่างสุ่ม (random error)

ความหมายของสัมประสิทธิ์ความถดถอยเชิงส่วน คือ ค่าสัมประสิทธิ์ที่แสดงถึงระดับความสัมพันธ์ของตัวแปรอิสระที่มีต่อตัวแปรตาม เช่น b_i เป็นค่าซึ่งแสดงความสัมพันธ์ระหว่าง Y และ x_i หมายถึงถ้าตัวแปร x_i เพิ่มขึ้น 1 หน่วยจะทำให้ Y เปลี่ยนแปลง b_i หน่วย

การวิเคราะห์ความถดถอยมีขั้นตอนดังนี้

1. ทำการประมาณค่าสัมประสิทธิ์ความถดถอยเชิงส่วนโดยใช้วิธีกำลังสองน้อยที่สุด เมื่อได้ค่าประมาณสัมประสิทธิ์ความถดถอยเชิงส่วนจะสามารถสร้างสมการถดถอยได้
2. ทดสอบสมการความถดถอยโดยใช้การวิเคราะห์ความแปรปรวนจำแนกแบบทางเดียว เพื่อทดสอบว่าตัวแปรอิสระ X มีความสัมพันธ์กับตัวแปรตาม Y จริงหรือไม่ โดยจะทดสอบว่า $\beta_1 = \beta_2 = \ldots = \beta_n = 0$ หรือไม่ ถ้าไม่เท่ากับศูนย์หมายความว่ามี X_i อย่างน้อยหนึ่งตัวมีความสัมพันธ์กับ Y ในรูปแบบเชิงเส้นจริง
3. เมื่อทดสอบสมการความถดถอยแล้วพบว่ามี X_i อย่างน้อยหนึ่งตัวมีความสัมพันธ์กับ Y ในรูปแบบเชิงเส้นจริง จะต้องทดสอบเพื่อหาความสัมพันธ์ระหว่าง X_i แต่ละตัวกับ Y ว่ามีความสัมพันธ์กันจริงหรือไม่ โดยทดสอบเพื่อดูว่ามีค่า β_i ใดบ้างที่เป็น 0 และถ้า β_i ใดที่เป็น 0 แสดงว่าตัวแปร X_i นั้นไม่ได้มีความสัมพันธ์กับ Y สามารถตัดออกจากสมการได้
4. เมื่อได้สมการประมาณค่า Y แล้วจึงทำการหาค่าสัมประสิทธิ์การตัดสินใจเชิงซ้อน (Multiple Coefficient of Determination: R^2) ซึ่งเป็นค่าสัดส่วนที่ตัวแปรอิสระ ($X_1, X_2, \ldots, X_k$) สามารถอธิบายการเปลี่ยนแปลง Y ได้ หรือเป็นเปอร์เซ็นต์ความผันแปร Y ที่มีสาเหตุเนื่องจากการผันแปรของ$X_1, X_2, \ldots,$และ X_k และจากค่าสัมประสิทธิ์การตัดสินใจเชิงซ้อนจะสามารถคำนวณหาค่าสัมประสิทธิ์สหสัมพันธ์เชิงซ้อน (Multiple Coefficient of Correlation: R) ได้โดยการถอดรากที่ 2 สัมประสิทธิ์การตัดสินใจเชิงซ้อน สัมประสิทธิ์สหสัมพันธ์เชิงซ้อนแสดงถึงความสัมพันธ์ระหว่าง Y กับ $X_1, X_2, \ldots,$และ X_k โดยถ้า R มีค่าเข้าใกล้ 0 แสดงว่า Y มีความสัมพันธ์กับ $X_1, X_2, \ldots, X_k$ น้อยมาก แต่ถ้า R เข้าใกล้ 1 แสดงว่า Y มีความสัมพันธ์กับ $X_1, X_2, \ldots, X_k$ มาก

การวิเคราะห์ความถดถอยเชิงซ้อนเป็นการศึกษาความสัมพันธ์ระหว่างตัวแปรตาม 1 ตัว และตัวแปรอิสระหลายตัวนั้น มีข้อกำหนดว่าตัวแปรอิสระเหล่านั้นต้องไม่มีความสัมพันธ์กัน เนื่องจากถ้าตัวแปรอิสระมีความสัมพันธ์ซึ่งกันและกันจะทำให้เกิดปัญหา Multicollinearity ซึ่งจะทำให้เกิดปัญหาดังนี้

1. ทำให้ค่าเบี่ยงเบนมาตรฐานของสัมประสิทธิ์ความถดถอยมีค่าสูงมาก
2. ทำให้เครื่องหมายของสัมประสิทธิ์ความถดถอยตรงกันข้ามกับที่ควรจะเป็น
3. ทำให้ค่าสัมประสิทธิ์ความถดถอยไม่คงที่เมื่อมีตัวแปรอิสระเพิ่มขึ้น

การเกิดปัญหาเหล่านี้ในท้ายที่สุดแล้วจะทำให้โมเดลที่ได้ไม่ถูกต้อง ดังนั้นจึงต้องแก้ไขด้วยการใช้วิธีการเลือกตัวแปรอิสระแบบสเตปไวส์ซึ่งเป็นการเลือกตัวแปรอิสระเข้าสู่สมการความถดถอยครั้งละ 1 ตัว ถ้าตัวแปรอิสระที่นำเข้ามีความสัมพันธ์กับตัวแปรอิสระที่มีอยู่แล้ว จะทำการตัดตัวแปรอิสระตัวใดตัวหนึ่งทิ้งไป ซึ่งขั้นตอนของวิธีการเลือกตัวแปรอิสระแบบสเตปไวส์เป็นดังนี้

1. ทำการเลือกตัวแปรอิสระ 1 ตัวที่มีความสัมพันธ์กับตัวแปรตามอย่างมีนัยสำคัญที่สุดเข้าสมการ โดยดูจากค่าสัมประสิทธิ์สหสัมพันธ์เชิงส่วน (Partial correlation) ซึ่งแสดงระดับความสัมพันธ์ระหว่างตัวแปรตาม และตัวแปรอิสระแต่ละตัว
2. จากตัวแปรอิสระที่เหลือ k-1 เลือกตัวแปรอิสระที่มีความสัมพันธ์กับตัวแปรตามอย่างมีนัยสำคัญที่สุดเข้าสมการ จากนั้นทำการทดสอบความสัมพันธ์ระหว่างตัวแปรตามกับตัวแปรอิสระใหม่ และอาจจะทำการตัดตัวแปรอิสระตัวใดตัวหนึ่งออกไปถ้าตัวแปรอิสระนั้นมีความสัมพันธ์กับตัวแปรอิสระอื่นๆ
3. ทำซ้ำขั้นตอนที่ 2 จนกระทั่งไม่สามารถนำตัวแปรอิสระเข้าสู่สมการ หรือไม่มีตัวแปรอิสระที่สามารถตัดออกได้อีก

2.1.5 การทำนายที่ระดับแอล (Prediction at Level L)

การทำนายที่ระดับแอล [10] เป็นวิธีที่ใช้ในการประเมินผลโมเดลการประมาณค่า เพื่อประเมินความคลาดเคลื่อนในการประมาณค่าของโมเดลเมื่อเทียบกับค่าจริง โดยจะทำการคำนวณหาร้อยละความคลาดเคลื่อนในการประมาณค่าของโมเดลเทียบกับค่าจริง การทำนายที่ระดับแอลจะอาศัยขนาดความผิดพลาดสัมพันธ์ซึ่งมีสมการดังนี้

$$MRE = |RE| = \left|\frac{E - \hat{E}}{E}\right|$$

โดย

MRE คือ ขนาดความผิดพลาดสัมพันธ์ (Magnitude relative error)

RE คือ ความผิดพลาดสัมพันธ์ (Relative error)

E คือ ค่าความเสถียรเชิงตรรกะของคลาส

$\hat{E}$ คือ ค่าประมาณค่าความเสถียรเชิงตรรกะของคลาสที่ได้จากโมเดล

การทำนายที่ระดับแอลมีสมการดังนี้

$$PRED(l) = \frac{k}{n}$$

โดย

$PRED(l)$ คือ การทำนายที่ระดับแอล

l คือ ช่วงระดับของความคลาดเคลื่อน มีค่าตั้งแต่ 0 -1

k คือ จำนวนหน่วยตัวอย่างที่มีค่า $MRE \leq l$

n คือ จำนวนหน่วยตัวอย่างทั้งหมด

โดยผลการทำนายที่ระดับแอลสามารถอธิบายได้ดังนี้ ถ้า PRED(0.2)=0.85 หมายความว่า 85% ของค่าที่ประมาณได้จะมีความคลาดเคลื่อนจากค่าจริงไม่เกิน 20% และการนำผลการทำนายระดับแอลมาใช้ในการประเมินผลโมเดลสำหรับประมาณค่านั้น โดยทั่วไปจะยอมรับโมเดลสำหรับประมาณค่าที่ผลการทำนายระดับแอลที่ระดับ 0.25 มากกว่าหรือเท่ากับ 0.75 [10] หรือค่าประมาณที่ได้จะต้องคลาดเคลื่อนจากค่าจริงไม่เกิน 25% เป็นจำนวน 75% ขึ้นไป

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 Investigation of Metrics for Object-Oriented Design Logical Stability [7] โดย Mahmoud O. Elish , David Rine

งานวิจัยนี้ได้ทำการศึกษาถึงความสัมพันธ์ระหว่างค่าความเสถียรเชิงตรรกะของคลาสกับตัววัดของซอฟต์แวร์เชิงวัตถุของ Chidamber และ Kemerer ทั้ง 6 ตัว ได้แก่

- Weighted Methods per Class (WMC)
- Depth of Inheritance Tree (DIT)
- Coupling Between Object classes (CBO)
- Response For a Class (RFC)
- Lack Of Cohesion in Methods (LCOM)
- Number Of Children (NOC)

โดยค่าตัววัดทั้ง 6 หาจากซอร์สโคด (Source Code) ของซอฟต์แวร์เช่นเดียวกับค่าความเสถียรเชิงตรรกะ

งานวิจัยนี้ได้ให้นิยามของความเสถียรว่าเป็นค่าความต้านทานต่อการกระจายระหว่างคลาสเมื่อเกิดการเปลี่ยนแปลง โดยแบ่งความเสถียรออกเป็น 2 แบบ คือ ความเสถียรเชิงตรรกะและความเสถียรเชิงประสิทธิภาพ ซึ่งงานวิจัยนี้มุ่งเน้นที่ความเสถียรเชิงตรรกะโดยอาศัยการศึกษาถึงการเปลี่ยนแปลงในระดับคลาสที่เป็นไปได้ว่ามีผลกระทบต่อคลาสอื่นในรูปแบบใดและทำการคำนวณหาค่าความเสถียรเชิงตรรกะจากจำนวนครั้งที่คลาสที่สนใจได้รับผลกระทบกับจำนวนครั้งที่เกิดการเปลี่ยนแปลงทั้งหมด

งานวิจัยนี้ได้ทำการเลือกโปรแกรมเพื่อทำการศึกษามา 3 โปรแกรมด้วยกัน ได้แก่ Jxplorer เวอร์ชัน 2.1.001 JE(Just an Editor) เวอร์ชัน 1.65 และ Phex เวอร์ชัน 0.7.3 ซึ่งทั้ง 3 โปรแกรมเป็นโปรแกรมโอเพนซอร์ส (Open Source) ที่พัฒนาโดยภาษาจาวา งานวิจัยนี้คำนวณหาค่าความเสถียรเชิงตรรกะตามวิธีการที่ได้สร้างขึ้นและหาค่าตัววัดทั้ง 6 ของทั้ง 3 โปรแกรม แล้วนำมาคำนวณหาค่าความสัมพันธ์แล้วทำการสรุป

ผลสรุปของงานวิจัยนี้พบว่ามีตัววัด 5 ตัวมีความสัมพันธ์เชิงลบกับค่าความเสถียรเชิงตรรกะของคลาส ได้แก่ WMC , DIT , CBO , RFC , LCOM โดย CBO และ RFC มีความสัมพันธ์สูงสามารถนำไปใช้เป็นตัวบ่งชี้ได้ แต่ไม่พบความสัมพันธ์ระหว่างตัววัด NOC กับค่าความเสถียรเชิงตรรกะของคลาส อย่างไรก็ตามงานวิจัยนี้ทำการศึกษาความสัมพันธ์จากซอร์สโคดซึ่งจำเป็นต้องรอจนโปรแกรมพัฒนาเสร็จ

2.2.2 Ripple effect analysis for Software Maintenance [5] โดย S. S. Yau, J. S. Collofello, T. MacGregor

งานวิจัยนี้ทำการวิเคราะห์และเสนอแนะแนวทางในการวิเคราะห์ผลกระทบวงคลื่นที่อาจเกิดขึ้นในกระบวนการบำรุงรักษา โดยแบ่งผลกระทบวงคลื่นออกเป็น 2 แบบด้วยกันดังนี้

1. ผลกระทบวงคลื่นเชิงตรรกะ (Logical Ripple Effect) คือ ผลกระทบที่เกิดจากการเปลี่ยนแปลงส่วนหนึ่งส่วนใดของซอฟต์แวร์แล้วทำให้เกิดความขัดแย้งในการทำงานของส่วนอื่นๆ เช่นการเปลี่ยนชนิดของตัวแปร

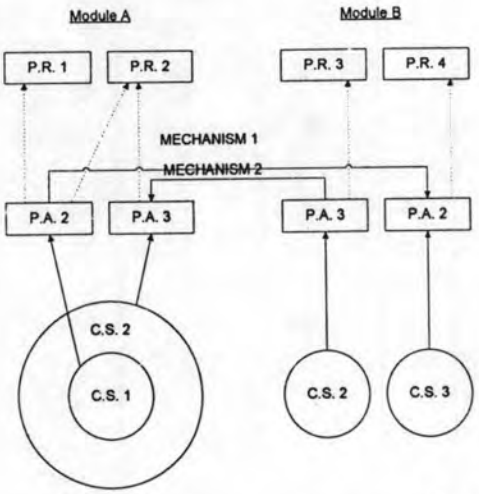
2. ผลกระทบวงคลื่นเชิงประสิทธิภาพ (Performance Ripple Effect) คือ ผลกระทบที่เกิดจากการเปลี่ยนแปลงส่วนหนึ่งส่วนใดของซอฟต์แวร์แล้วส่งผลให้ประสิทธิภาพของส่วนอื่นๆ ลดลง

ผลกระทบวงคลื่นเชิงตรรกะคือ การเกิดผลกระทบวงคลื่นต่อโปรแกรมในด้านการทำงานของโปรแกรม กล่าวคือเมื่อมีการเปลี่ยนแปลงส่วนหนึ่งส่วนใดของโปรแกรมแล้วทำให้ส่วนอื่นทำงานผิดพลาดไปจากเดิม หรือไม่สามารถทำงานได้ เช่น สมมติให้ส่วนย่อย A มีการประกาศตัวแปร X และส่วนย่อย B มีการเรียกใช้ตัวแปร X จาก A เมื่อมีการแก้ไขโดยเปลี่ยนประเภทของตัวแปร X แล้วทำให้ส่วนย่อย B ที่เรียกใช้ตัวแปร X ไม่สามารถทำงานได้ เป็นต้น

ผลกระทบวงคลื่นเชิงประสิทธิภาพ คือการเกิดผลกระทบวงคลื่นต่อโปรแกรมโดยมีผลกระทบต่อประสิทธิภาพ กล่าวคือเมื่อมีการเปลี่ยนแปลงส่วนหนึ่งส่วนใดของโปรแกรมแล้วส่งผลให้ประสิทธิภาพของส่วนอื่นของโปรแกรมลดลง เช่น สมมติให้ส่วนย่อย A มีการเปลี่ยนแปลงแล้วส่วนย่อย B จากเดิมสามารถประมวลผลเสร็จสิ้นในเวลา 1 วินาที ใช้เวลา 2 วินาทีในการประมวลผล เป็นต้น

ในงานวิจัยนี้ได้ทำการเสนอแนวทางในการวิเคราะห์ผลกระทบวงคลื่น สำหรับผลกระทบวงคลื่นเชิงตรรกะทำการวิเคราะห์โปรแกรมแล้วกำหนดแหล่งกำเนิดข้อผิดพลาดหลัก (Primary Error Source) ซึ่งคือจุดที่ทำการแก้ไขแล้วมีโอกาสทำให้เกิดความไม่สอดคล้องกันต่อส่วนอื่นๆ จากนั้นทำการหาแหล่งกำเนิดข้อผิดพลาดที่สอง (Secondary Error Source) ซึ่งคือจุดที่ได้รับผลกระทบจากการเปลี่ยนแปลงของแหล่งกำเนิดข้อผิดพลาดหลัก สำหรับการวิเคราะห์ผลกระทบวงคลื่นเชิงประสิทธิภาพทำได้โดยทำการวิเคราะห์ความต้องการเชิงประสิทธิภาพ (Performance Requirement หรือ P.R.) เพื่อสร้างเป็นคุณสมบัติทางประสิทธิภาพ (Performance Attribute หรือ P.A.) ซึ่งคุณสมบัติทางประสิทธิภาพเหล่านี้จะได้รับผลกระทบเมื่อมีการเปลี่ยนแปลงส่วนของโปรแกรมซึ่งจะส่งผลต่อความต้องการเชิงประสิทธิภาพต่อไป ส่วนของโปรแกรมที่เปลี่ยนแปลงแล้วส่งผลกระทบต่อคุณสมบัติทางประสิทธิภาพเรียกว่า ส่วนวิกฤต (Critical section หรือ C.S.)

นอกจากนี้ผลกระทบที่เกิดขึ้นต่อคุณสมบัติทางประสิทธิภาพหนึ่งยังส่งผลต่อคุณสมบัติทางประสิทธิภาพอื่นผ่านทางกลไกการทำงาน (Mechanism) ดังรูปที่ 2.4



รูปที่ 2.4 การส่งผลกระทบต่อกันของการเกิดผลกระทบวงคลื่นเชิงประสิทธิภาพ [5]

งานวิจัยนี้ได้เสนอแนะวิธีการในการทำนายความซับซ้อนของการเปลี่ยนแปลงซึ่งจะช่วยในการวิเคราะห์ถึงภาระงานของโปรแกรมเมอร์ที่ต้องการ อย่างไรก็ตามวิธีการทำนายที่สร้างขึ้นไม่ได้บอกวิธีการวัดอย่างชัดเจน

โดยสรุปงานวิจัยนี้ได้ทำการวิเคราะห์และเสนอวิธีการสำหรับวิเคราะห์ผลกระทบวงคลื่นที่อาจเกิดขึ้นโดยผลลัพธ์ที่ได้เป็นเพียงกลุ่มของส่วนโปรแกรมที่จะได้รับผลกระทบเท่านั้นไม่ได้เป็นค่าตัวเลขที่สามารถนำไปคำนวณได้ นอกจากนี้วิธีการวิเคราะห์ของงานวิจัยนี้เป็นวิธีการสำหรับโปรแกรมเชิงกระบวนคำสั่ง (Procedural Programming) เท่านั้น แต่ในวิทยานิพนธ์นี้จะมุ่งเน้นที่โปรแกรมเชิงวัตถุซึ่งเป็นที่แพร่หลายในปัจจุบัน

2.2.3 Some stability measures for software maintenance [3] โดย S. Yau , J. Collofello , T. MacGregor

งานวิจัยนี้ทำการศึกษาและเสนอแนะวิธีการในการวัดความเสถียรเชิงตรรกะของโปรแกรมเชิงกระบวนคำสั่ง โดยให้นิยามความเสถียรเป็นค่าความต้านทานของผลกระทบวงคลื่นที่อาจเกิดขึ้นเมื่อโปรแกรมมีการเปลี่ยนแปลง โดยทำการวิเคราะห์และเสนอแนะสมการและอัลกอริทึมสำหรับคำนวณความเสถียรเชิงตรรกะของโปรแกรม

งานวิจัยนี้ได้มองความเสถียรของโปรแกรมเป็นคุณสมบัติหนึ่งของโปรแกรมที่ส่งผลต่อกระบวนการบำรุงรักษา โดยโปรแกรมที่มีค่าความเสถียรสูงจะช่วยให้กระบวนการบำรุงรักษาทำ

ได้อย่างง่ายและลดค่าใช้จ่ายที่เกิดขึ้น ทั้งนี้เนื่องจากถ้าความเสถียรมีค่าต่ำผลกระทบที่เกิดขึ้นจากการเปลี่ยนแปลงจะมีความซับซ้อนสูงทำให้แก้ไขได้ยาก งานวิจัยนี้ได้แบ่งความเสถียรออกเป็น 2 แบบ คือ ความเสถียรเชิงตรรกะและความเสถียรเชิงประสิทธิภาพ โดยมุ่งเน้นที่การคำนวณความเสถียรเชิงตรรกะเท่านั้น

ในงานวิจัยนี้ได้แบ่งการคำนวณค่าความเสถียรเชิงตรรกะออกเป็น 2 ส่วนคือ การหาความเสถียรเชิงตรรกะของโมดูล และการหาความเสถียรเชิงตรรกะของโปรแกรม โดยความเสถียรเชิงตรรกะของโมดูลคือค่าความต้านทานต่อผลกระทบการเปลี่ยนแปลงที่อาจเกิดขึ้นเมื่อทำการเปลี่ยนแปลงโมดูลหนึ่งต่อโมดูลอื่นๆ ในโปรแกรมในแง่ของการทำงานทางตรรกะของโปรแกรม โดยอาศัยการเปลี่ยนแปลงอย่างง่ายและเล็กที่สุดคือการเปลี่ยนแปลงของตัวแปรในการคำนวณ

เมื่อมีการเปลี่ยนแปลงตัวแปรในโมดูลหนึ่งจะส่งผลกระทบต่อตัวแปรอื่นในโมดูลเดียวกัน ปรากฏการณ์นี้เรียกว่า การไหลของข้อผิดพลาดภายในโมดูล (Intra module error flow) และเช่นเดียวกันการเปลี่ยนแปลงนั้นจะส่งผลกระทบต่อตัวแปรของโมดูลอื่นๆ ด้วยเรียกว่าการไหลของข้อผิดพลาดระหว่างโมดูล (Inter module error flow)

วิธีการหาค่าความเสถียรเชิงตรรกะของโมดูลของงานวิจัยนี้ทำได้โดย

1) หาตัวแปรส่วนติดต่อภายนอกที่ได้รับผลกระทบจากการเปลี่ยนแปลงตัวแปรภายในโมดูลหนึ่งๆ ซึ่งคือการวิเคราะห์การไหลของข้อผิดพลาดภายในโมดูลที่เกิดจากการเปลี่ยนแปลงตัวแปรใดๆ
2) หาตัวแปรของโมดูลอื่นๆ ที่ได้รับผลกระทบที่เกิดขึ้นต่อตัวแปรส่วนติดต่อภายนอก ซึ่งคือการวิเคราะห์การไหลของข้อผิดพลาดระหว่างโมดูล
3) หาโมดูลทั้งหมดที่เกี่ยวข้องต่อการไหลของข้อผิดพลาดระหว่างโมดูล
4) คำนวณความซับซ้อนของการแก้ไขโมดูลนั้นๆ จากการเปลี่ยนแปลงตัวแปรใดๆ โดยใช้วิธีการคำนวณความซับซ้อนของแมคเคบ (McCabe's complexity) เป็นตัววัดความซับซ้อนในการแก้ไข
5) คำนวณค่าผลกระทบวงคลื่นเชิงตรรกะโดยอาศัยผลรวมของความน่าจะเป็นของการเกิดการเปลี่ยนแปลงตัวแปรใดๆ คูณกับความซับซ้อนของการแก้ไขโมดูลจากการเปลี่ยนแปลงตัวแปรนั้นๆ
6) คำนวณค่าความเสถียรเชิงตรรกะของโมดูลโดยให้เป็นส่วนกลับของค่าผลกระทบวงคลื่น

และจากค่าความเสถียรเชิงตรรกะของแต่ละโมดูลสามารถนำไปคำนวณค่าความเสถียรเชิงตรรกะของโปรแกรมได้ดังนี้

1) หาค่าผลกระทบวงคลื่นเชิงตรรกะของโปรแกรมโดยอาศัยผลรวมของผลคูณระหว่างความน่าจะเป็นที่จะเกิดการแก้ไขโมดูลใดๆ กับค่าผลกระทบวงคลื่นของโมดูลนั้นๆ
2) คำนวณค่าความเสถียรเชิงตรรกะของโปรแกรมโดยให้เป็นส่วนกลับของค่าผลกระทบวงคลื่นของโปรแกรม

ผลสรุปของงานวิจัยนี้คือ วิธีการและกระบวนการในการวัดค่าความเสถียรเชิงตรรกะของโปรแกรม ที่สามารถวัดค่าเป็นตัวเลขเพื่อนำมาใช้งานได้ อย่างไรก็ตามวิธีการของงานวิจัยนี้เป็นวิธีการสำหรับโปรแกรมเชิงกระบวนคำสั่งและมีความซับซ้อนค่อนข้างสูงในการวัด ทำให้เมื่อนำไปใช้กับระบบที่มีขนาดใหญ่จะทำให้เสียเวลามากและอาจเกิดข้อผิดพลาดได้

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย