แบบจำลองการวัดความเสถียรสำหรับระบบเซอร์วิซ

นายอัคคนัฐ จอมจุมพล

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

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

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2558
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

A Stability Measurement Model for Service System

Mr. Akkanat Chomchumpol

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Software Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2015

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	แบบจำลองการวัดความเสถียรสำหรับระบบเซอร์วิซ
โดย	นายอัคคนัฐ จอมจุมพล
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร. ทวิตีย์ เสนีวงศ์ ณ อยุธยา

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร. สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ
(รองศาสตราจารย์ ดร. พรศิริ หมื่นไชยศรี)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร. ทวิตีย์ เสนีวงศ์ ณ อยุธยา)

.....กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร. มฑุปายาส ทองมาก)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

อัคคนัฐ จอมจุมพล : แบบจำลองการวัดความเสถียรสำหรับระบบเซอร์วิซ (A Stability Measurement Model for Service System) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร. ทวิตีย์ เสนีวงศ์ ณ อยุธยา, 141 หน้า.

เทคโนโลยีเซอร์วิซ เช่น เว็บเซอร์วิซเป็นส่วนหนึ่งของเทคโนโลยีหลักในการพัฒนาซอฟต์แวร์ เนื่องจากคุณลักษณะเด่น เช่น การแยกส่วนต่อประสานออกจากการพัฒนา ความสามารถใช้ซ้ำได้ง่าย และความสามารถในการประกอบกันได้ง่าย เซอร์วิซมีลักษณะเหมือนซอฟต์แวร์ทั่วไปคือมีการปรับปรุงและบำรุงรักษาอยู่ตลอดเวลาอันเนื่องมาจากความต้องการหรือสภาพแวดล้อมการทำงานที่เปลี่ยนแปลงไปรวมถึงการแก้ไขข้อผิดพลาดที่เกิดขึ้น การเปลี่ยนแปลงของเซอร์วิซที่เกิดขึ้นนั้น โดยเฉพาะการเปลี่ยนแปลงส่วนต่อประสานอาจจะส่งผลกระทบต่อส่วนอื่น ๆ ของซอฟต์แวร์ ที่ใช้งานเซอร์วิซนั้น ดังนั้นการพิจารณาความเสถียรของเซอร์วิซ จะช่วยให้นักออกแบบเซอร์วิซทราบว่าระบบเซอร์วิซมีวิวัฒนาการที่ยังคงการออกแบบเดิมไว้อยู่ได้ดีเพียงใด วิทยานิพนธ์เล่มนี้เสนอแบบจำลองการวัดความเสถียรของระบบเซอร์วิซ แบบจำลองนี้มีพื้นฐานมาจากความเสถียรเชิงตรรกะของเว็บเซอร์วิซ ซึ่งวัดค่าความเสถียรจากระดับความไม่เปลี่ยนแปลงของส่วนต่อประสานของเซอร์วิซเมื่อเกิดวิวัฒนาการจากเวอร์ชันหนึ่งไปยังอีกเวอร์ชันหนึ่ง ค่าความเสถียรนั้นพิจารณาระหว่างคู่ของเวอร์ชันใด ๆ ของเซอร์วิซและของระบบเซอร์วิซ รวมทั้งความเสถียรโดยรวมของเซอร์วิซและของระบบเซอร์วิซที่มีวิวัฒนาการแบบเป็นเวอร์ชันคู่ขนาน ค่าความเสถียรที่วัดได้จะช่วยนักออกแบบเซอร์วิซในการบำรุงรักษาระบบเซอร์วิซ เนื่องจากเซอร์วิซเวอร์ชันใหม่ควรได้รับการออกแบบให้คงความเสถียรไว้เพื่อให้เกิดผลกระทบจากการเปลี่ยนแปลงน้อยที่สุดเท่าที่จะทำได้ นอกจากนี้นักออกแบบเซอร์วิซยังสามารถทราบได้ว่าเซอร์วิซและระบบเซอร์วิซมีความเสถียรเป็นอย่างไรตลอดช่วงชีวิต ซึ่งช่วยในการตัดสินใจนำเซอร์วิซมาใช้ซ้ำ หรือวางแผนต้นทุนและความพยายามในการบำรุงรักษาเซอร์วิซ

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมซอฟต์แวร์ ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ปีการศึกษา 2558

5670991321 : MAJOR SOFTWARE ENGINEERING
KEYWORDS: STABILITY / WEB SERVICE / SERVICE-ORIENTED SYSTEM / MEASUREMENT / SOFTWARE EVOLUTION

AKKANAT CHOMCHUMPOL: A Stability Measurement Model for Service System. ADVISOR: ASSOC. PROF. DR. TWITTIE SENIVONGSE, 141 pp.

Service technology, such as Web service, has been part of the mainstream of software development for some time due to its promising characteristics such as interface-implementation decoupling, reusability, and composability. Like other software, a service has to undergo maintenance during its lifetime to cope with changes in requirements and environment or to handle errors. Unfortunately certain changes of a service, especially service interface changes, can have effects on other parts of the system which use the service. It is therefore useful to determine service stability so that a service designer can determine how well the service-oriented system evolves while preserving the design. This thesis proposes a stability measurement model for service-oriented systems. The model is based on logical stability of a Web service which is measured by the degree of unchanged elements of the service interface during its evolution from one version to another. The thesis considers stability between any pair of service and system versions as well as the overall stability of the service and system which involve several parallel versions. The knowledge of service stability can help the service designer during maintenance of the service-oriented system since the new service version should be designed to preserve stability and cause as minimal impacts as possible. The service designer can also observe how stable the services and the whole system are over their lifetime when considering service reuse or planning service maintenance cost and effort.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Department:	Computer Engineering	Student's Signature	.....
Field of Study:	Software Engineering	Advisor's Signature	.....
Academic Year:	2015		

กิตติกรรมประกาศ

ข้าพเจ้าขอกราบขอบพระคุณ รองศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่สละเวลา ให้ความรู้ ให้คำแนะนำและให้คำปรึกษา ตลอดจนแนวทางในการจัดทำวิทยานิพนธ์ให้สำเร็จลุล่วงไปด้วยดี ขอขอบคุณ อาจารย์ทุกท่านที่ได้ให้ความรู้แก่ข้าพเจ้าตลอดระยะเวลาการศึกษา

ขอขอบพระคุณ รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี ประธานกรรมการสอบวิทยานิพนธ์ และผู้ช่วยศาสตราจารย์ ดร.มฑุปายาส ทองมาก กรรมการสอบวิทยานิพนธ์ ที่กรุณาให้คำแนะนำต่าง ๆ ทำให้วิทยานิพนธ์นี้มีความสมบูรณ์มากขึ้น

ขอขอบคุณ บริษัท โลคัส เทเลคอมมิวนิเคชัน อิงค์ สำหรับความอนุเคราะห์ที่อนุญาตให้ใช้ข้อมูลต่าง ๆ เพื่อการศึกษา และขอบคุณครอบครัวที่คอยให้กำลังใจและสนับสนุนตลอดการศึกษาจนสามารถสำเร็จลุล่วงด้วยดี

ข้าพเจ้าหวังเป็นอย่างยิ่งว่า วิทยานิพนธ์เล่มนี้จะเป็นประโยชน์ต่อผู้ที่สนใจ หากมีข้อผิดพลาดประการใด ข้าพเจ้าขอน้อมรับไว้เพื่อนำไปใช้ในการปรับปรุงต่อไป ความดีและประโยชน์จากวิทยานิพนธ์เล่มนี้ข้าพเจ้าขอมอบให้แก่ผู้มีพระคุณทุกท่าน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญตาราง 1
สารบัญภาพ 3
บทที่ 1 บทนำ 4
1.1 ที่มาและความสำคัญของปัญหา 4
1.2 วัตถุประสงค์ของงานวิจัย 5
1.3 ขอบเขตของงานวิจัย 5
1.4 ประโยชน์ของงานวิจัย 6
1.5 ขั้นตอนและวิธีดำเนินงานวิจัย 6
1.6 โครงสร้างของเนื้อหาในวิทยานิพนธ์ 7
1.7 ผลงานตีพิมพ์จากวิทยานิพนธ์ 7
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 8
2.1 ทฤษฎีที่เกี่ยวข้อง 8
2.1.1 ความเสถียรของซอฟต์แวร์ 8
2.1.1.1 ความเสถียรเชิงตรรกะของซอฟต์แวร์ 8
2.1.1.2 ความเสถียรเชิงประสิทธิภาพของซอฟต์แวร์ 8
2.1.2 เว็บเซอร์วิซ 9
2.1.3 วิสเดิล 10
2.2 งานวิจัยที่เกี่ยวข้อง 12

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

หน้า

2.2.1 Towards measuring object-oriented class stability 12
2.2.2 A Study of Design Characteristics in Software Using Stability as a Criterion 14
2.2.3 End-to-End Versioning Support for Web Services 17
2.2.4 Research on Maintainability Evaluation of Service-Oriented Software 18
2.2.5 A Controlled Experiment for Evaluating the Impact of Coupling on the Maintainability of Service-Oriented Software 19
2.2.6 A Quantitative Measurement and Validation of Granularity in Service Oriented Architecture 20
2.2.7 Cohesion Metrics for Predicting Maintainability of Service-Oriented Software 21
บทที่ 3 แบบจำลองการวัดความเสถียรสำหรับระบบเซอร์วิซ 22
3.1 การสร้างแบบจำลองการวัดความเสถียรระหว่างเวอร์ชัน 22
3.1.1 ความเสถียรของโอเปอเรชัน 24
3.1.2 ความเสถียรของเซอร์วิซเดี่ยว 28
3.1.3 ความเสถียรของระบบเซอร์วิซ 29
3.2 การสร้างแบบจำลองการวัดความเสถียรโดยรวม 30
3.2.1 ความเสถียรโดยรวมของโอเปอเรชัน 32
3.2.2 ความเสถียรโดยรวมของเซอร์วิซเดี่ยว 36
3.2.3 ความเสถียรโดยรวมของระบบเซอร์วิซ 38
บทที่ 4 การทดลองและประเมินแบบจำลอง 40
4.1 วัตถุประสงค์ของการประเมิน 40
4.2 การประเมินแบบจำลองการวัดความเสถียร 40

หน้า

4.2.1 การประเมินแบบจำลองการวัดความเสถียรของเซอร์วิซโดยการหาความสัมพันธ์กับขนาดของส่วนต่อประสาน 42
4.2.2 การประเมินแบบจำลองการวัดความเสถียรของเซอร์วิซโดยการหาความสัมพันธ์กับการเชื่อมติดกันของข้อมูลในส่วนต่อประสานของเซอร์วิซ 44
4.2.3 การประเมินแบบจำลองการวัดความเสถียรของเซอร์วิซโดยการหาความสัมพันธ์กับความสามารถในการบำรุงรักษา 45
4.3 สรุปผลการประเมิน 47
บทที่ 5 การพัฒนาเครื่องมือสนับสนุน 48
5.1 ความต้องการที่เป็นฟังก์ชัน 48
5.1.1 ส่วนการนำเข้าข้อมูล 48
5.1.2 ส่วนการประมวลผล 48
5.1.3 ส่วนการแสดงผล 49
5.2 ความสามารถของเครื่องมือ 49
5.3 การออกแบบเครื่องมือ 50
5.3.1 แผนภาพยูสเคส 50
5.3.2 แผนภาพคลาส 51
5.3.3 แผนภาพลำดับ 52
5.3.4 แผนภาพกิจกรรม 54
5.3.5 การออกแบบส่วนต่อประสาน 55
5.4 สภาพแวดล้อมในการพัฒนาเครื่องมือ 70
5.4.1 ฮาร์ดแวร์ 70
5.4.2 ซอฟต์แวร์ 70
5.5 ทดสอบเครื่องมือ 70
บทที่ 6 สรุปผลการวิจัย ข้อจำกัด และข้อเสนอแนะ 72

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

หน้า

6.1 สรุปผลการวิจัย 72
6.2 อุปสรรคและข้อจำกัด 72
6.3 ข้อเสนอแนะ 73
รายการอ้างอิง 74
ภาคผนวก 76
ภาคผนวก ก การใช้งานเครื่องมือสนับสนุนการวัดค่าความเสถียร 77
ภาคผนวก ข วิสเดิลที่ใช้ในการประเมินแบบจำลอง 82
ประวัติผู้เขียนวิทยานิพนธ์ 141

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญตาราง

ตารางที่ 2.1 ส่วนประกอบของวิสเดิล..... 10

ตารางที่ 2.2 วิธีการนับจำนวนคุณสมบัติของคลาสซึ่งไม่ถูกเปลี่ยนแปลง [4] 13

ตารางที่ 3.1 ปัจจัยและเงื่อนไขการนับของการวัดความเสถียรของโอเปอเรชัน (เอลิเมนต์เชิงฟังก์ชัน).....25

ตารางที่ 3.2 การคำนวณหาค่าความเสถียรของโอเปอเรชันระหว่างเวอร์ชัน 0 และ 1..... 26

ตารางที่ 3.3 ปัจจัยและเงื่อนไขการนับของการวัดความเสถียรของเอลิเมนต์ที่ไม่ใช่เชิงฟังก์ชัน 28

ตารางที่ 3.4 ค่าความเสถียรของแต่ละเอลิเมนต์เชิงฟังก์ชันของเวอร์ชันตั้งต้นและเวอร์ชันคู่ขนานสุดท้าย V_2..... 35

ตารางที่ 3.5 ค่าความเสถียรของแต่ละเอลิเมนต์เชิงฟังก์ชันของเวอร์ชันตั้งต้นและเวอร์ชันคู่ขนานสุดท้าย V_3..... 35

ตารางที่ 4.1 ความเสถียรของเซอร์วิซของวิสเดิลตัวอย่าง.....42

ตารางที่ 4.2 ผลการวัดค่าขนาดของส่วนต่อประสานและความเสถียรของเซอร์วิซ..... 42

ตารางที่ 4.3 ผลการวัดค่าการเชื่อมติดกันของข้อมูลในส่วนต่อประสานของเซอร์วิซและความเสถียรของเซอร์วิซ..... 44

ตารางที่ 4.4 ผลการวัดค่าจำนวนบรรทัดที่แก้ไขเฉลี่ยและความเสถียรของเซอร์วิซ..... 46

ตารางที่ 5.1 คำอธิบายฟังก์ชันการทำงาน.....51

ตารางที่ 5.2 รายละเอียดหน้าจอเลือกมุมมองการวัดค่าความเสถียร 56

ตารางที่ 5.3 รายละเอียดหน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของโอเปอเรชัน 57

ตารางที่ 5.4 รายละเอียดหน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของเซอร์วิซ..... 58

ตารางที่ 5.5 รายละเอียดหน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของระบบเซอร์วิซ..... 60

ตารางที่ 5.6 รายละเอียดหน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของ 63

ตารางที่ 5.7 รายละเอียดหน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของเซอร์วิซ 64

ตารางที่ 5.8 รายละเอียดหน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของระบบเซอร์วิซ 66

ตารางที่ 5.9 รายละเอียดหน้าจอแสดงผลการวัดค่าความเสถียรระหว่างเวอร์ชัน 68

ตารางที่ 5.10 รายละเอียดหน้าจอแสดงผลการวัดค่าความเสถียรโดยรวม 69

ตารางที่ 5.11 ผลการเปรียบเทียบค่าความเสถียรจากการคำนวณของผู้วิจัยและจากเครื่องมือ 71

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญภาพ

ภาพที่ 2.1 องค์ประกอบของเทคโนโลยีเว็บเซอร์วิซ 9

ภาพที่ 2.2 ตัวอย่างวิสเดิล [8] 11

ภาพที่ 2.3 คอร์ด (Chord) และความแตกต่าง (Variation) ของเวอร์ชันแบบลำดับและแบบคู่ขนานของระบบซอฟต์แวร์ [5] 16

ภาพที่ 2.4 การจำแนกประเภทของการเปลี่ยนแปลงเวอร์ชัน [9] 17

ภาพที่ 3.1 ภาพรวมวิธีการดำเนินงาน 22

ภาพที่ 3.2 การวัดความเสถียรแต่ละประเภท 24

ภาพที่ 3.3 ตัวอย่างบางส่วนวิสเดิลของเซอร์วิซ “MathService” [8] ในเวอร์ชันตั้งต้น 27

ภาพที่ 3.4 การเปลี่ยนแปลงประเภทข้อมูลของ “MathInput” ในเวอร์ชัน 1 27

ภาพที่ 3.5 ตัวอย่างเวอร์ชันคู่ขนาน 30

ภาพที่ 3.6 วิสเดิลที่ถูกเปลี่ยนแปลงเป็นเวอร์ชันคู่ขนานสุดท้าย V_2 34

ภาพที่ 3.7 วิสเดิลที่ถูกเปลี่ยนแปลงเป็นเวอร์ชันคู่ขนานสุดท้าย V_3 34

ภาพที่ 4.1 แผนภาพกระจัดกระจายของขนาดของส่วนต่อประสานและความเสถียรของเซอร์วิซ 43

ภาพที่ 4.2 แผนภาพกระจัดกระจายของการเชื่อมติดกันของข้อมูลในส่วนต่อประสานของเซอร์วิซและความเสถียรของเซอร์วิซ 45

ภาพที่ 4.3 แผนภาพกระจัดกระจายของจำนวนบรรทัดที่แก้ไขเฉลี่ยของวิสเดิลและความเสถียรของเซอร์วิซ 47

ภาพที่ 5.1 แผนภาพยูสเคส 50

ภาพที่ 5.2 แผนภาพคลาสเชิงตรรกะ 52

ภาพที่ 5.3 แผนภาพลำดับของการวัดความเสถียรของเซอร์วิซ 53

ภาพที่ 5.4 แผนภาพกิจกรรมภาพรวมการทำงานของเครื่องมือ 54

ภาพที่ 5.5 หน้าจอเลือกมุมมองของการวัดค่าความเสถียร 55

ภาพที่ 5.6 หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของโอเปอเรชัน 56

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ภาพที่ 5.7 หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของเซอร์วิซ 58
ภาพที่ 5.8 หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของระบบเซอร์วิซ 60
ภาพที่ 5.9 หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของโอเปอเรชัน 62
ภาพที่ 5.10 หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของเซอร์วิซ 64
ภาพที่ 5.11 หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของระบบเซอร์วิซ 66
ภาพที่ 5.12 ผลการวัดค่าความเสถียรระหว่างเวอร์ชัน 68
ภาพที่ 5.13 ผลการวัดค่าความเสถียรโดยรวม 69
ภาพที่ ก.1 หน้าต่างการเลือกมุมมองการวัดค่าความเสถียร 77
ภาพที่ ก.2 หน้าจอการนำเข้าและระบุข้อมูลให้แก่วิสเดิลในกลุ่มการวัดความเสถียรระหว่างเวอร์ชัน 79
ภาพที่ ก.3 หน้าจอการนำเข้าและระบุข้อมูลให้แก่วิสเดิลในกลุ่มการวัดความเสถียรโดยรวม 80
ภาพที่ ก.4 หน้าจอแสดงผลการวัดค่าความเสถียร 81

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 1
บทนำ

เนื้อหาในบทนี้กล่าวถึงภาพรวมของงานวิจัย ซึ่งประกอบด้วย ที่มาและความสำคัญของปัญหา วัตถุประสงค์ ขอบเขตของงานวิจัย ประโยชน์ ขั้นตอนและวิธีดำเนินการวิจัย และโครงสร้างของเนื้อหางานวิจัย ซึ่งมีเนื้อหาดังต่อไปนี้

1.1 ที่มาและความสำคัญของปัญหา

ขั้นตอนบำรุงรักษาถือเป็นขั้นตอนที่สำคัญกับทุกระบบซอฟต์แวร์ ไม่ว่าจะเป็นการบำรุงรักษาตามระยะเวลาเพื่อให้ซอฟต์แวร์ทำงานได้ปรกติหรือจะเป็นการปรับปรุงเพื่อให้ซอฟต์แวร์ทำงานได้ดียิ่งขึ้นหรือรองรับความต้องการที่เปลี่ยนแปลง ซึ่งขั้นตอนบำรุงรักษานี้มีต้นทุนค่อนข้างสูง ส่วนหนึ่งอาจเป็นผลเกี่ยวเนื่องมาจากขั้นตอนออกแบบตั้งแต่ระดับสถาปัตยกรรมไปจนถึงระดับการเขียนโปรแกรม

ใน ISO 9126 ความสามารถในการบำรุงรักษา (Maintainability) นั้นแยกย่อยได้เป็น Analysability, Changeability, Testability, Maintainability Compliance และ Stability ดังนั้นความเสถียรของซอฟต์แวร์ (Software Stability) จึงเป็นตัวชี้วัดหนึ่งที่บ่งบอกถึงคุณภาพของซอฟต์แวร์ (Software Quality) ซึ่งเป็นค่าความต้านทานต่อการได้รับผลกระทบที่เกิดจากการเปลี่ยนแปลงซอฟต์แวร์ (Software Change) [1] ความเสถียรแบ่งออกเป็น 2 ประเภทคือ ความเสถียรเชิงตรรกะ (Logical Stability) ซึ่งเป็นความเสถียรในเชิงโครงสร้างของซอฟต์แวร์ และความเสถียรเชิงประสิทธิภาพ (Performance Stability) ซึ่งเป็นความเสถียรของพฤติกรรมในขณะทำงานจริงของซอฟต์แวร์

ระบบเซอร์วิซ (Service System) เป็นระบบซอฟต์แวร์ที่ภายในประกอบด้วยการทำงานร่วมกันของเซอร์วิซต่าง ๆ โดยเทคโนโลยีที่นิยมใช้คือ เว็บเซอร์วิซ (Web Services) ระบบเซอร์วิซได้รับความนิยมอย่างแพร่หลายเนื่องจากแนวคิดที่มุ่งเน้นการนำเซอร์วิซกลับมาใช้ใหม่ (Reusable) ทำให้ช่วยลดต้นทุนในการพัฒนาซอฟต์แวร์และเช่นเดียวกับซอฟต์แวร์ทั่วไป การเปลี่ยนแปลงที่ส่งผลกับส่วนต่อประสานของเซอร์วิซ (Service Interface) จะส่งผลกระทบกับผู้เรียกใช้งานเซอร์วิซนั้น ๆ ดังนั้นการทราบข้อมูลความเสถียรของเซอร์วิซจะเป็นข้อมูลช่วยในการพิจารณาเลือกใช้เซอร์วิซของผู้เรียกใช้งาน นอกจากนี้ถ้าผู้พัฒนาระบบสามารถประเมินความเสถียรของระบบซึ่งได้รับการออกแบบใหม่เพื่อตอบสนองต่อการเปลี่ยนแปลง ข้อมูลความเสถียรนี้จะช่วยในการพิจารณาเลือกการออกแบบที่เหมาะสม เพื่อลดผลกระทบกับระบบอื่น ๆ ที่เรียกใช้งาน ส่งผลให้ช่วยลดต้นทุนในการบำรุงรักษา

ของระบบที่เรียกใช้งานด้วย ในมุมมองของผู้บริหารโครงการ การวัดค่าความเสถียรของระบบเซอร์วิซสามารถช่วยในการสังเกตแนวโน้มของระบบเซอร์วิซนั้น ๆ ว่ามีแนวโน้มของความเสถียรเป็นอย่างไร และข้อมูลนี้ช่วยในการบริหารจัดการความเสี่ยง หรือประมาณการทรัพยากรในการปรับปรุงระบบเซอร์วิซได้

หลายปีที่ผ่านมาได้มีผู้ค้นคว้าวิจัยเกี่ยวกับการวัดความเสถียรของซอฟต์แวร์ไม่ว่าจะเป็นการวัดความเสถียรเชิงโครงสร้าง [2] หรือจะเป็นการประมาณค่าความเสถียรจากการวิเคราะห์เชิงตรรกะ [3] หรือ การวิเคราะห์จากการเปลี่ยนแปลงของคุณสมบัติของคลาส [4] แต่ทั้งหมดนี้เป็นการประมาณค่าความเสถียรในมุมมองของคลาสเชิงวัตถุ ซึ่งยังไม่ครอบคลุมถึงการวัดความเสถียรในมุมมองสำหรับวิวัฒนาการของระบบเซอร์วิซ

งานวิจัยนี้จะเสนอแบบจำลองการวัดค่าความเสถียรในระบบเซอร์วิซโดยจะนำเอาหลักการวัดความเสถียรของซอฟต์แวร์เชิงวัตถุซึ่งวิเคราะห์การเปลี่ยนแปลงของคุณสมบัติของคลาส [4] มาประยุกต์ใช้เพื่อวัดความเสถียรของระบบเซอร์วิซและความเสถียรของเซอร์วิซเดี่ยวและโอเปอเรชันและประยุกต์การศึกษาวิวัฒนาการของซอฟต์แวร์โดยพิจารณาการเปลี่ยนแปลงของลักษณะการออกแบบ (Design Characteristic) [5] เพื่อวัดความเสถียรโดยรวมของระบบเซอร์วิซและของเซอร์วิซเดี่ยวและโอเปอเรชัน ในกรณีที่มีการพัฒนาแบบคู่ขนานตั้งแต่ 2 เส้นทางขึ้นไป

1.2 วัตถุประสงค์ของงานวิจัย

1) เพื่อสร้างแบบจำลองในการคำนวณค่าความเสถียรระหว่างเวอร์ชันของระบบเซอร์วิซและความเสถียรโดยรวมของระบบเซอร์วิซ
2) เพื่อสร้างเครื่องมือในการคำนวณค่าและแสดงผลความเสถียรของระบบเซอร์วิซ

1.3 ขอบเขตของงานวิจัย

1) การวัดความเสถียรระบบเซอร์วิซนั้นวิเคราะห์จากการไม่เปลี่ยนแปลงของส่วนประกอบในส่วนต่อประสานของเซอร์วิซในระบบในเวอร์ชันต่างๆ โดยครอบคลุมถึงความเสถียรต่อไปนี้
    - ความเสถียรระหว่างเวอร์ชัน
        - ความเสถียรของโอเปอเรชัน
        - ความเสถียรของเซอร์วิซเดี่ยว
        - ความเสถียรของระบบเซอร์วิซ
    - ความเสถียรโดยรวม
        - ความเสถียรโดยรวมของโอเปอเรชัน

- ความเสถียรโดยรวมของเซอร์วิซเดี่ยว
- ความเสถียรโดยรวมของระบบเซอร์วิซ

2) ส่วนต่อประสานของเซอร์วิซอธิบายด้วยวิสเดิล (WSDL)
3) พัฒนาเครื่องมือสำหรับการคำนวณค่าความเสถียรของระบบเซอร์วิซโดยจะแสดงค่าความเสถียรเท่านั้น มิได้เสนอแนวทางในการปรับปรุงแก้ไข ซึ่งเครื่องมือนั้นพัฒนาด้วยภาษาจาวาและทดสอบบนระบบปฏิบัติการวินโดวส์ (Windows) และไม่รองรับวิสเดิลที่อ้างอิง <types> ไปยังอีกวิสเดิลหนึ่ง
4) ประเมินแบบจำลองโดยทดลองความสอดคล้องกับตัววัดอื่นสำหรับเซอร์วิซ

1.4 ประโยชน์ของงานวิจัย

1) ได้แบบจำลองการคำนวณค่าความเสถียรของระบบเซอร์วิซซึ่งสามารถนำไปใช้เป็นแนวทางให้ผู้พัฒนาซอฟต์แวร์พิจารณาความเสถียรของระบบที่จะออกแบบก่อนการนำไปพัฒนาซึ่งจะช่วยลดต้นทุนในการบำรุงรักษาต่อไป รวมทั้งช่วยในการทำความเข้าใจวิวัฒนาการของระบบเซอร์วิซ
2) ได้เครื่องมือสำหรับการคำนวณค่าความเสถียรของระบบเซอร์วิซ

1.5 ขั้นตอนและวิธีดำเนินงานวิจัย

1) ศึกษาทฤษฎีพื้นฐานของความเสถียรของซอฟต์แวร์
2) ศึกษาทำความเข้าใจเกี่ยวกับสถาปัตยกรรมเชิงเซอร์วิซ (Service-Oriented Architecture: SOA)
3) ศึกษาและวิเคราะห์งานวิจัยที่จะนำมาใช้วัดความเสถียรของเซอร์วิซ ทั้งในเชิงความเสถียรของเซอร์วิซระหว่างเวอร์ชันและในเชิงภาพรวมของระบบเซอร์วิซ
4) สร้างแบบจำลองการคำนวณค่าความเสถียรของเซอร์วิซทั้งหมด 2 แบบคือการคำนวณความเสถียรของเซอร์วิซระหว่างเวอร์ชันและการคำนวณความเสถียรโดยรวมของเซอร์วิซ
5) ประเมินแบบจำลอง
6) วิเคราะห์ผลการทดลอง ออกแบบและพัฒนาเครื่องมือ
7) ทดสอบ ประเมินผลและปรับปรุงเครื่องมือ
8) สรุปผล ข้อเสนอแนะ งานวิจัยในอนาคต
9) จัดทำเอกสารผลงานวิชาการ

10) จัดทำวิทยานิพนธ์

1.6 โครงสร้างของเนื้อหาในวิทยานิพนธ์

เนื้อหาของวิทยานิพนธ์ฉบับนี้แบ่งออกเป็น 6 บทคือ บทที่ 1 บทนำ กล่าวถึงที่มาและความสำคัญ วัตถุประสงค์ ขอบเขตและประโยชน์ของงานวิจัยเป็นต้น บทที่ 2 กล่าวถึงทฤษฎีรวมถึงงานวิจัยต่าง ๆ ที่เกี่ยวข้อง บทที่ 3 เสนอแนวคิดของงานวิจัยและขั้นตอนในการสร้างแบบจำลองในมุมมองต่าง ๆ ได้แก่ 1) ความเสถียรระหว่างเวอร์ชันของโอเปอเรชัน เซอร์วิซเดี่ยว และระบบเซอร์วิซ 2) ความเสถียรโดยรวมของโอเปอเรชัน เซอร์วิซเดี่ยว และระบบเซอร์วิซ และการประมาณค่าความเสถียรของระบบเซอร์วิซในมุมมองนั้น ๆ ซึ่งประกอบด้วยขั้นตอนในการดำเนินการวิจัย รวมทั้งแนวคิดและรายละเอียดในการสร้างแบบจำลอง บทที่ 4 เป็นการประเมินผลแบบจำลอง โดยการเปรียบเทียบความสัมพันธ์กับตัววัดอื่นที่มีผลเกี่ยวกับความสามารถในการบำรุงรักษา บทที่ 5 อธิบายการออกแบบและพัฒนาเครื่องมือที่ใช้สนับสนุนการวัดความเสถียรในมุมมองต่าง ๆ และบทที่ 6 เป็นบทสุดท้าย ซึ่งจะเป็นบทสรุปของงานวิจัย ข้อจำกัด รวมทั้งข้อเสนอแนะงานวิจัยในอนาคต

1.7 ผลงานตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์ได้รับการตีพิมพ์ในบทความชื่อ “Stability measurement model for service-oriented systems” ในการประชุมระดับนานาชาติ “9th Malaysian Software Engineering Conference (MySEC2015)” ซึ่งจัดขึ้นที่กรุงกัวลาลัมเปอร์ ประเทศมาเลเซีย ระหว่างวันที่ 16-17 ธันวาคม พ.ศ. 2558

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องกับงานวิจัยนี้แบ่งออกเป็น 3 กลุ่มคือ ทฤษฎีเกี่ยวกับความเสถียรของซอฟต์แวร์, ทฤษฎีเกี่ยวกับเว็บเซอร์วิซ และทฤษฎีเกี่ยวกับวิสเดิล ซึ่งมีรายละเอียดดังต่อไปนี้

2.1.1 ความเสถียรของซอฟต์แวร์

ความเสถียรของซอฟต์แวร์ [1] เป็นความต้านทานต่อผลกระทบที่เกิดจากความเปลี่ยนแปลงที่เกิดขึ้นกับซอฟต์แวร์ ซึ่งเป็นคุณสมบัติหนึ่งที่บ่งบอกถึงคุณภาพของซอฟต์แวร์แบ่งออกเป็น 2 ประเภทดังนี้

2.1.1.1 ความเสถียรเชิงตรรกะของซอฟต์แวร์

ความเสถียรเชิงตรรกะของซอฟต์แวร์ คือโอกาสในการไม่ได้รับผลกระทบจากการเปลี่ยนแปลงซึ่งส่งผลให้การทำงานผิดพลาดหรือไม่สามารถทำงานได้ ซึ่งจำแนกออกได้เป็น 2 ประเภทคือ

1) ความเสถียรเชิงตรรกะของคลาส (Logical Stability of a Class) คือ ค่าความต้านทานการได้รับผลกระทบจากการเปลี่ยนแปลงของคลาสใด ๆ ในโปรแกรม เช่น คลาสหนึ่งมีค่าความเสถียรเชิงตรรกะเท่ากับ 0.70 จะหมายความว่าหากเกิดความเปลี่ยนแปลงขึ้นในคลาสอื่น ๆ โอกาสที่คลาสนี้จะไม่ได้รับผลกระทบเชิงตรรกะจะมีค่าเท่ากับ 70% หรือในการเปลี่ยนแปลง 100 ครั้ง คลาสนี้จะยังทำงานได้คงเดิมและไม่ได้รับผลกระทบเชิงตรรกะ 70 ครั้ง เป็นต้น

2) ความเสถียรเชิงตรรกะของโปรแกรม (Logical Stability of a Program) คือ ค่าความต้านทานการได้รับผลกระทบจากการเปลี่ยนแปลงของโปรแกรม

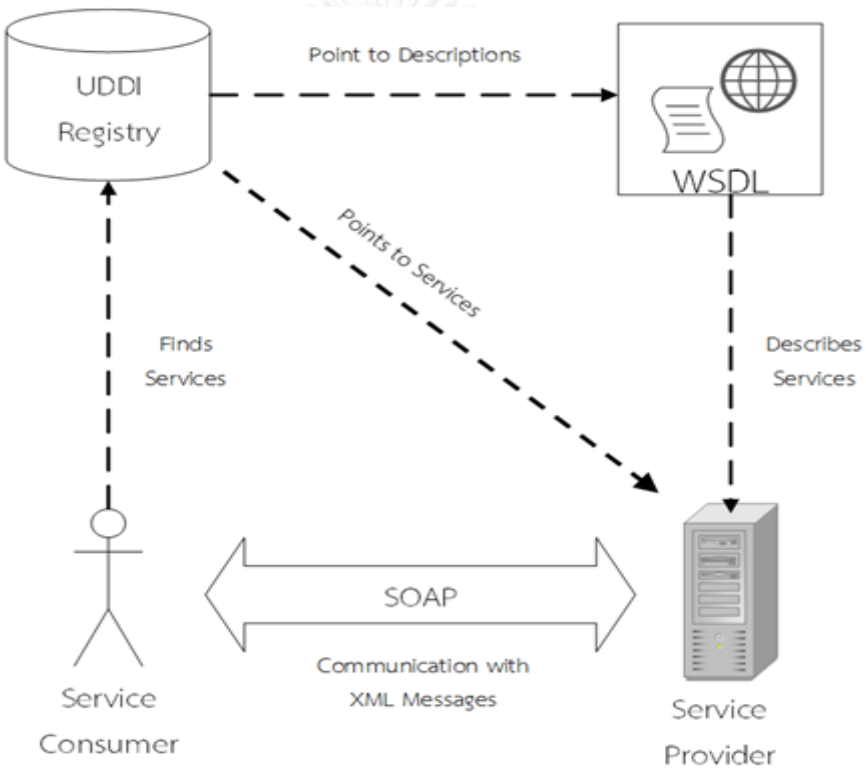
2.1.1.2 ความเสถียรเชิงประสิทธิภาพของซอฟต์แวร์

ความเสถียรเชิงประสิทธิภาพของซอฟต์แวร์คือ โอกาสในการไม่ได้รับผลกระทบจากการเปลี่ยนแปลงซึ่งส่งผลให้ประสิทธิภาพในการทำงานเปลี่ยนไป การหาค่าความเสถียรเชิงประสิทธิภาพของซอฟต์แวร์ต้องกำหนดประสิทธิภาพที่สนใจและมีการวิเคราะห์ที่ซับซ้อน การวัดค่าจะวัดได้เมื่อโปรแกรมทำงานจริง

ในงานวิจัยนี้จะเป็นการวัดความเสถียรเชิงตรรกะ ซึ่งจะพิจารณาส่วนต่อประสานของเซอร์วิซที่มีการเปลี่ยนแปลงระหว่าง 2 เวอร์ชัน รวมถึงการพิจารณาการเปลี่ยนแปลงของส่วนต่อประสานของเซอร์วิซของเวอร์ชันคู่ขนานที่มากกว่า 2 เวอร์ชันขึ้นไป

2.1.2 เว็บเซอร์วิซ

เว็บเซอร์วิซ (Web Services) [6] คือแอปพลิเคชันหรือโปรแกรมที่ทำงานอย่างใดอย่างหนึ่งในลักษณะให้บริการบนเว็บโดยจะถูกเรียกใช้งานจากแอปพลิเคชันอื่น ๆ ในรูปแบบ RPC (Remote Procedure Call) ซึ่งให้บริการและจะมีเอกสารที่อธิบายคุณสมบัติของบริการกำกับไว้ องค์ประกอบของเทคโนโลยีเว็บเซอร์วิซเป็นดังภาพที่ 2.1 ประกอบด้วยมาตรฐานดังต่อไปนี้



ภาพที่ 2.1 องค์ประกอบของเทคโนโลยีเว็บเซอร์วิซ

- โซป (SOAP: Simple Object Access Protocol) เป็นโปรโตคอลที่กำหนดรูปแบบข้อความที่แลกเปลี่ยนกับเว็บเซอร์วิซโดยข้อความเป็นแบบเอกซ์เอ็มแอล (XML) และการส่งข้อความทำโดยโปรโตคอลเอชทีทีพี (HTTP)
- วิสเดิล (WSDL: Web Service Description Language) เป็นภาษาที่ใช้อธิบายคุณลักษณะของเว็บเซอร์วิซและวิธีติดต่อกับเว็บเซอร์วิซ เป็นภาษาที่อยู่ในการดูแลของ W3C (World

Wide Web Consortium)

- ยูดีดีไอ (UDDI: Universal Description, Discovery and Integration) เป็นที่เก็บรวบรวมข้อมูลที่เกี่ยวกับเว็บเซอร์วิซต่าง ๆ ในอินเทอร์เน็ตไว้ในแหล่งเดียวกันเพื่อให้ผู้ใช้บริการสามารถค้นหาได้ง่าย

ผู้ให้บริการ (Service Provider) จะพัฒนาเว็บเซอร์วิซและลงทะเบียนเซอร์วิซที่ยูดีดีไอ (UDDI Registry) เพื่อเป็นการประกาศใช้งานซึ่งเอกสารอธิบายการให้บริการคือวิสเดิล (WSDL) ส่วนผู้ใช้บริการ (Service Consumer) จะเป็นผู้เรียกใช้เว็บเซอร์วิซโดยทำการค้นหาเซอร์วิซที่ยูดีดีไอแล้วทำการเรียกใช้เซอร์วิซจากผู้ให้บริการผ่านโปรโตคอล SOAP

2.1.3 วิสเดิล

วิสเดิล (WSDL: Web Service Description Language) [7] เป็นภาษาที่ใช้อธิบายการบริการของเว็บเซอร์วิซโดยใช้ภาษาเอกซ์เอ็มแอลในการกำหนด วิสเดิลจึงอธิบายส่วนต่อประสานของเว็บเซอร์วิซ (Service Interface) โดยมีส่วนประกอบดังตารางที่ 2.1 เอลิเมนต์ <PortType>, <message> และ <types> จัดเป็นเอลิเมนต์เชิงฟังก์ชัน (Functional Element) เนื่องจากเกี่ยวข้องกับฟังก์ชันการให้บริการ ส่วนเอลิเมนต์ <binding> และ <service> จัดเป็นเอลิเมนต์ที่ไม่ใช่เชิงฟังก์ชัน (Non-functional Element) ตัวอย่างของวิสเดิลของเว็บเซอร์วิซสำหรับบวกเลขแสดงในภาพที่ 2.2

ตารางที่ 2.1 ส่วนประกอบของวิสเดิล

เอลิเมนต์	ความหมาย
<portType>	อธิบายส่วนต่อประสานของเว็บเซอร์วิซ ซึ่งระบุ <operation> ที่เว็บเซอร์วิซมีให้บริการและ <message> ที่รับเข้าและส่งออกของแต่ละ <operation> เทียบได้กับไลบรารี มอดูล หรือคลาสในการเขียนโปรแกรม
<message>	อธิบายเอลิเมนต์ข้อมูล (Data elements) ของ <operation> แต่ละ <message> อาจมีมากกว่าหนึ่งส่วนเทียบได้กับพารามิเตอร์ของฟังก์ชันในการเขียนโปรแกรม
<types>	อธิบายชนิดข้อมูลภายใน <message> เพื่อความเป็นกลางวิสเดิลใช้เอกซ์เอ็มแอลสกีมา (XML Schema) ในการกำหนดชนิดข้อมูล
<binding>	อธิบายรูปแบบของข้อความโซปและโปรโตคอลที่ใช้ในการติดต่อเว็บเซอร์วิซ
<service>	อธิบาย <port> ที่ใช้ติดต่อเว็บเซอร์วิซในรูปแบบของยูอาร์แอล (URL) ของเว็บเซอร์วิซบนเว็บเซิร์ฟเวอร์

```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"  xmlns:y="http://example.org/math/"
  xmlns:ns="http://example.org/math/types/"
targetNamespace="http://example.org/math/">
   <types>
     <xs:schema targetNamespace="http://example.org/math/types/"
       xmlns="http://example.org/math/types/"
       elementFormDefault="unqualified" attributeFormDefault="unqualified">
<xs:complexType name="MathInput">
           <xs:sequence>
              <xs:element name="x" type="xs:double"/>
              <xs:element name="y" type="xs:double"/>
           </xs:sequence>
        </xs:complexType>
        <xs:complexType name="MathOutput">
           <xs:sequence>
              <xs:element name="result" type="xs:double"/>
           </xs:sequence>
        </xs:complexType>
         <xs:element name="Add" type="MathInput"/>
        <xs:element name="AddResponse" type="MathOutput"/>
     </xs:schema>
   </types>
   <message name="AddMessage">
     <part name="parameters" element="ns:Add"/>
   </message>
  <message name="AddResponseMessage">
     <part name="parameters" element="ns:AddResponse"/>
   </message>
   <portType name="MathInterface">
     <operation name="Add">
        <input message="y:AddMessage"/>
        <output message="y:AddResponseMessage"/>
     </operation>
   </portType>
   <binding name="MathSoapHttpBinding" type="y:MathInterface">
     <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
```

ภาพที่ 2.2 ตัวอย่างวิสเดิล [8]

```
    </operation>
  </portType>
<binding name="MathSoapHttpBinding" type="y:MathInterface">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="Add">
      <soap:operation soapAction="http://example.org/math/#Add"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="MathService">
    <port name="MathEndpoint" binding="y:MathSoapHttpBinding">
      <soap:address location="http://localhost/math/math.asmx"/>
    </port>
  </service>
</definitions>
```

ภาพที่ 2.2 ตัวอย่างวิสเดิล [8] (ต่อ)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.2 งานวิจัยที่เกี่ยวข้อง

เนื่องจากงานวิจัยนี้เกี่ยวข้องกับการวัดความเสถียรซึ่งความเสถียรมีความเกี่ยวเนื่องกับการเปลี่ยนแปลง ดังนั้นงานวิจัยที่เกี่ยวข้องจึงเป็นงานวิจัยที่เกี่ยวกับวิธีวัดความเสถียรของซอฟต์แวร์และการเปลี่ยนแปลงเวอร์ชันของซอฟต์แวร์

2.2.1 Towards measuring object-oriented class stability

งานวิจัยของ Alshayeb และคณะ [4] เสนอแบบจำลองวัดความเสถียรระดับคลาสเชิงวัตถุ โดยจำแนกคุณสมบัติซึ่งหากมีการเปลี่ยนแปลงจะมีผลต่อความเสถียรของคลาส 8 คุณสมบัติ ได้แก่

1. Class access-level
2. Class interface name

3. Inherited class name
4. Class variable
5. Class variable access-level
6. Method signature
7. Method access-level
8. Method body

โดยมีประเภทของการเปลี่ยนแปลงคุณสมบัติต่าง ๆ ดังนี้

1. Addition: คุณสมบัติที่ไม่มีในเวอร์ชัน i แล้วถูกเพิ่มในเวอร์ชัน i + 1
2. Deletion: คุณสมบัติที่มีในเวอร์ชัน i แล้วถูกลบในเวอร์ชัน i + 1
3. Modification: คุณสมบัติที่มีในเวอร์ชัน i แล้วถูกแก้ไขในเวอร์ชัน i + 1
4. Unchanged: คุณสมบัติที่มีในเวอร์ชัน i แล้วไม่ถูกแก้ไขในเวอร์ชัน i + 1

ในส่วนของ Unchanged จะเป็นตัวชี้วัดความเสถียรซึ่งหมายถึงคุณสมบัติใดไม่มีการเปลี่ยนแปลงในระหว่างเวอร์ชัน i+1 และ i คุณสมบัตินั้นมีความเสถียร การนับจำนวน Unchanged ของคุณสมบัติต่าง ๆ ระหว่าง เวอร์ชัน i+1 และ i เป็นไปตามตารางที่ 2.2

ตารางที่ 2.2 วิธีการนับจำนวนคุณสมบัติของคลาสซึ่งไม่ถูกเปลี่ยนแปลง [4]

	Metric		How to classify	Count
1	class access-level	unchanged	if the class access-level has not been modified or deleted from version *i* to version *i*+1	+1
2	class interface name	unchanged	if the interface name has not been modified or deleted from version *i* to version *i*+1	+1
3	inherited class name	unchanged	if the inherited class name has not been modified or deleted from version *i* to version *i*+1	+1
4	class variable	unchanged	if the class variable has not been modified or deleted from version *i* to version *i*+1	+1
5	class variable access-level	unchanged	if the class variable access-level has not been modified or deleted from version *i* to version *i*+1	+1
6	method signature	unchanged	if the method signature has not been modified or deleted from version *i* to version *i*+1	+1
7	method access-level	unchanged	if the method access-level has not been modified or deleted from version *i* to version *i*+1	+1
8	method body (code)	unchanged	if number of lines and the contents for a method did not change from version *i* to version *i*+1	+1

ค่าความเสถียรของคุณสมบัติ (Property Stability) คำนวณได้โดยใช้ตัววัด

$$Stab_{Property} = Unchanged_{Property} / Number_{Property}$$

โดยที่ $Stab_{Property}$ คือความเสถียรของคุณสมบัติใด ๆ ของคลาส

$Unchanged_{Property}$ คือจำนวนของ item ที่ไม่เปลี่ยนแปลงของคุณสมบัตินั้น ๆ

$Number_{Property}$ คือจำนวนรวมของ item ของคุณสมบัตินั้น ๆ

ดังนั้นค่าความเสถียรของคลาส (Class Stability) จึงได้จากการนำผลรวมของค่าความเสถียรของคุณสมบัติ 8 คุณสมบัติ หารด้วยจำนวนของคุณสมบัติที่กำหนด (ในงานวิจัยนี้ คือ 8 คุณสมบัติ) โดยใช้ตัววัด

$$Stability_{CLASS} = (Stab_{ClassAL}+Stab_{Interface}+Stab_{Inhr}+Stab_{Mthd} +Stab_{Var}+Stab_{VarAL}+Stab_{MthdAL}+Stab_{Body})/Properties_{CLASS}$$

ผู้วิจัยจะนำแบบจำลองการวัดของงานวิจัยนี้มาประยุกต์ในการวัดความเสถียรของเซอร์วิซระหว่างเวอร์ชันโดยคุณสมบัติที่จะนำมาพิจารณาคือเอลิเมนต์ต่าง ๆ ในวิสเดิล

2.2.2 A Study of Design Characteristics in Software Using Stability as a Criterion

งานวิจัยของ Kelly [5] ทำการศึกษาลักษณะการออกแบบ (Design Characteristic) ของซอฟต์แวร์ที่ถูกใช้งานมานาน ผ่านการแก้ไขมาแล้วหลายเวอร์ชันและเป็นซอฟต์แวร์ที่ประสบความสำเร็จ โดยหากลักษณะการออกแบบมีความเสถียรหรือเปลี่ยนแปลงน้อยในช่วงวิวัฒนาการของซอฟต์แวร์ จะถือว่าลักษณะการออกแบบนี้มีแนวโน้มที่จะส่งผลดีต่อความสามารถในการบำรุงรักษาของซอฟต์แวร์ งานวิจัยนี้มีสมมติฐานว่าลักษณะการออกแบบจะมีตัววัดการออกแบบ (Design Metric) ที่ให้ค่าที่สะท้อนถึงลักษณะการออกแบบนั้น (เช่น ตัววัด loc per module จะให้ค่าที่สะท้อนถึงลักษณะ module size) และพิจารณาความเสถียรของลักษณะการออกแบบจากระยะทาง (Distance) ระหว่างค่าตัววัดลักษณะการออกแบบซึ่งเปลี่ยนไปในแต่ละเวอร์ชันของซอฟต์แวร์

นิยาม: ลักษณะการออกแบบของซอฟต์แวร์จะเสถียรถ้าพิจารณาเวอร์ชันของซอฟต์แวร์นั้นตั้งแต่ 2 เวอร์ชันขึ้นไปแล้วค่าตัววัดของลักษณะการออกแบบนั้นมีความแตกต่างกันน้อย

ทฤษฎี: คอร์ด(Chord) คือ เส้นตรงภายในวงกลมโดยปลายทั้งสองจรดเส้นรอบวง และเส้นผ่านศูนย์กลางเป็นคอร์ดที่ยาวที่สุด

ในการวัดลักษณะของการออกแบบจะใช้หลักการของระยะทาง (Distance), คอร์ด (Chord) และความแตกต่าง (Variation) ซึ่งมีวิธีการคำนวณดังนี้

กำหนดให้เมตริกสเปซคือ เซต M ที่มีฟังก์ชันจำนวนจริง D(x, y) ซึ่งในที่นี้คือฟังก์ชันระยะทางที่กำหนดไว้สำหรับ x และ y ใด ๆ ที่เป็นสมาชิกของเซต M

ให้ $M = \{T_i, i = 0, 1,..., n\}$ โดยที่ n เป็นจำนวนธรรมชาติ และ T_i หมายถึงเวอร์ชันที่ i ของระบบซอฟต์แวร์ ในซอฟต์แวร์เวอร์ชัน T_i สามารถพิจารณาลักษณะการออกแบบได้หลายแง่มุม เช่น จำนวนบรรทัดในมอดูล ค่าใช้จ่ายในการพัฒนา เป็นต้น ลักษณะการออกแบบแต่ละอย่างในซอฟต์แวร์เวอร์ชัน T_i นี้กำหนดให้เป็น a และสมมติให้ลักษณะการออกแบบ a สามารถแทนได้ด้วยตัววัดที่ให้ค่า a ดังนั้นระยะทาง D ในเมตริกสเปซ M_a คือ

$$D(a(T_i), a(T_j)) = |a(T_i) - a(T_j)|$$

เนื่องจากเซต M เป็นเซตจำกัด ดังนั้นเส้นผ่านศูนย์กลางคือ

$$\max(D(a(T_i), a(T_j))) , T_i, T_j \in M$$

แต่เนื่องจากระยะทางที่จะพิจารณาเป็นระยะทางระหว่างจุดบางจุดที่สนใจของซับเซตของ M ดังนั้นสามารถนิยามคอร์ดได้คือ

$$\max(D(a(T_i), a(T_j))) ,$$ **เมื่อแต่ละ** T_i, T_j **เป็นสมาชิกของบางซับเซตที่สนใจใน** M

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

กำหนดให้ T_0 เป็นเวอร์ชันเริ่มต้นที่ทุกเวอร์ชันอื่น ๆ ในเซต M มีวิวัฒนาการต่อมา สองคอร์ดที่จะพิจารณาคือ

$$\delta_a = \max(D(a(T_i), a(T_j))) , i,j > 0,$$
$$\Delta_a = \max(D(a(T_0), a(T_j))) , j > 0.$$

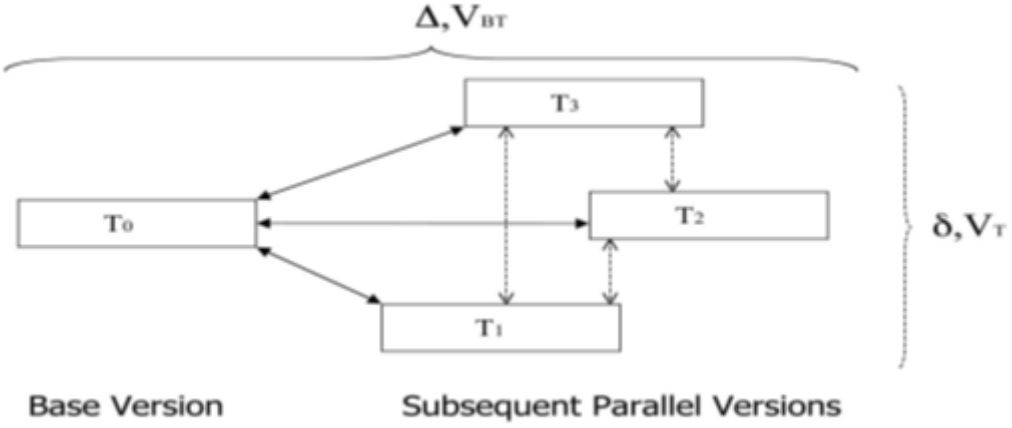
ทั้ง δ และ Δ จะมีหน่วยตามตัววัดของลักษณะการออกแบบ a เช่น จำนวนบรรทัดของโปรแกรมมีหน่วยเป็นบรรทัด, ค่าใช้จ่ายในการพัฒนามีหน่วยเป็นบาท เป็นต้น ดังนั้นจึงต้องทำการ Normalize เพื่อให้สามารถเปรียบเทียบค่าคอร์ดของแต่ละลักษณะการออกแบบที่มีหน่วยของตัววัดแตกต่างกันได้ นิยามของความแตกต่าง (Variation) ซึ่งหมายถึงคอร์ดที่ได้รับการ Normalize แล้วคือ

$V_T = \boldsymbol{\delta}_a / \max(a(T_i))$, $T_i \in M$

$V_{BT} = \Delta_a / \max(a(T_i))$, $T_i \in M$

แต่เนื่องจากในงานวิจัยนี้ผู้วิจัยสนใจตัววัดเพียงตัวเดียวคือความเสถียร จึงไม่ต้องทำการ Normalization โดยผู้วิจัยจะพิจารณาค่า $\boldsymbol{\delta}$ และ Δ เท่านั้น

ภาพที่ 2.3 แสดงถึงความสัมพันธ์ของ $\boldsymbol{\delta}_a$, Δ_a, V_T และ V_{BT} ด้วยเวอร์ชันที่พัฒนาแบบลำดับและแบบคู่ขนาน (เวอร์ชันที่แตกสาขามาจากเวอร์ชันตั้งต้น) 3 เวอร์ชันคือ T_1, T_2 และ T_3 โดยที่ T_0 เป็นเวอร์ชันตั้งต้น ในแนวตั้งค่า $\boldsymbol{\delta}_a$ และ V_T คือค่าความแตกต่างที่มากที่สุดระหว่างเวอร์ชันคู่ขนานด้วยกัน ส่วนในแนวนอนค่า Δ_a และ V_{BT} คือค่าความแตกต่างที่มากที่สุดระหว่างเวอร์ชันตั้งต้นไปยังเวอร์ชันคู่ขนาน หาก V_{BT} ของลักษณะที่พิจารณามีความเสถียรมาก (V_{BT} มีค่าน้อย) แล้วจะสามารถตีความได้ว่าลักษณะการออกแบบที่ออกแบบมาตั้งแต่ต้นนั้นเหมาะสมแล้วจึงไม่ค่อยเปลี่ยนแปลงทำให้สามารถบำรุงรักษาได้ง่าย ส่วน V_T จะพิจารณาระหว่างเวอร์ชันสุดท้ายของเวอร์ชันคู่ขนานด้วยกันเอง ถ้า V_T มีค่าน้อยแสดงว่าลักษณะการออกแบบนั้นเริ่มเข้าที่และเริ่มมาบรรจบกันในเวอร์ชันคู่ขนานสุดท้าย ส่งผลให้มีความแตกต่างของลักษณะการออกแบบเพียงเล็กน้อย



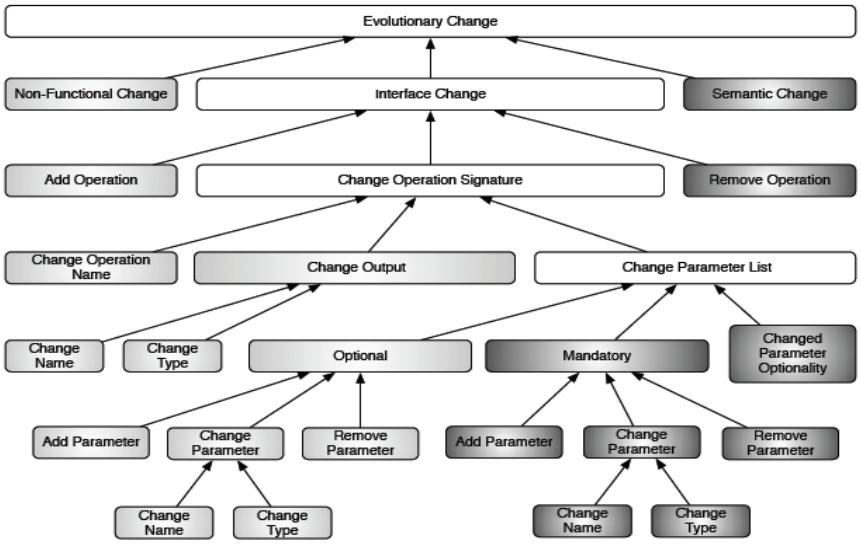
ภาพที่ 2.3 คอร์ด (Chord) และความแตกต่าง (Variation) ของเวอร์ชันแบบลำดับและแบบคู่ขนานของระบบซอฟต์แวร์ [5]

2.2.3 End-to-End Versioning Support for Web Services

งานวิจัยของ Philipp Leitner และคณะ [9] กล่าวถึงประเภทของการเปลี่ยนแปลงที่พบในระหว่างเวอร์ชันของเว็บเซอร์วิซโดยได้อธิบายประเภทของการเปลี่ยนแปลงส่วนต่อประสานเชิงฟังก์ชัน (Functional Interface) ของเว็บเซอร์วิซ ส่วนต่อประสานเชิงฟังก์ชันประกอบด้วยทุกโอเปอเรชันที่กำหนดไว้ในวิสเดิล ทุกโอเปอเรชันในส่วนต่อประสานเชิงฟังก์ชันประกอบด้วย ชื่อโอเปอเรชัน, รายการของพารามิเตอร์และเอาท์พุท ทั้งพารามิเตอร์และเอาท์พุทประกอบด้วยชื่อและประเภทข้อมูล ในส่วนของพารามิเตอร์นั้นมีทั้งแบบทางเลือก (Optional) และแบบจำเป็น (Mandatory) ส่วนอื่น ๆ ในวิสเดิลของเซอร์วิซ เช่น ที่อยู่ของเอนด์พ้อยต์ (Endpoint Address), สไตล์การเข้ารหัสของวิสเดิล (WSDL Encoding Style) เป็นต้น จัดอยู่ในส่วนต่อประสานที่ไม่ใช่เชิงฟังก์ชัน (Non-Functional Interface) ของเซอร์วิซ

ภาพที่ 2.4 เป็นแผนภาพการจำแนกประเภทของการเปลี่ยนแปลงโดยที่การเปลี่ยนแปลงระดับบนแบ่งออกเป็น 3 ประเภทคือ 1) การเปลี่ยนแปลงที่ไม่ใช่เชิงฟังก์ชัน (Non-Functional Change) คือการเปลี่ยนแปลงของส่วนต่อประสานที่ไม่ใช่เชิงฟังก์ชันของเว็บเซอร์วิซ 2) การเปลี่ยนแปลงของส่วนต่อประสาน (Interface Change) คือการเปลี่ยนแปลงของส่วนต่อประสานเชิงฟังก์ชันของเว็บเซอร์วิซ และ 3) การเปลี่ยนแปลงเชิงความหมาย (Semantic Change) ครอบคลุมการเปลี่ยนแปลงทั้งหมดที่ไม่ใช่การเปลี่ยนแปลงโครงสร้างของวิสเดิล

เนื่องจากงานวิจัยนี้พิจารณาการเปลี่ยนแปลงของส่วนต่อประสานตามวิสเดิล ผู้วิจัยจึงพิจารณาการเปลี่ยนแปลง 2 ประเภทคือ การเปลี่ยนแปลงของส่วนต่อประสานเชิงฟังก์ชันและที่ไม่ใช่เชิงฟังก์ชัน



ภาพที่ 2.4 การจำแนกประเภทของการเปลี่ยนแปลงเวอร์ชัน [9]

2.2.4 Research on Maintainability Evaluation of Service-Oriented Software

MA Zhe และ BEN Kerong [10] ได้เสนองานวิจัยเกี่ยวกับการประเมินความสามารถในการบำรุงรักษาของซอฟต์แวร์เชิงเซอร์วิซ รวมถึงได้วิเคราะห์ถึงปัจจัยที่มีผลต่อความสามารถในการบำรุงรักษา และได้แยกย่อยความสามารถในการบำรุงรักษาออกเป็น ความสามารถในการวิเคราะห์ (Analyzability), ความสามารถในการแก้ไข (Changeability), ความเสถียร (Stability) และความสามารถในการทดสอบ (Testability)

การบำรุงรักษานั้นเป็นส่วนที่ต้องปฏิบัติหลังจากที่พัฒนาซอฟต์แวร์เสร็จสิ้นและใช้งานแล้ว การบำรุงรักษานั้นอาจจะเกิดขึ้นจากการปรับปรุงประสิทธิภาพหรือแก้ไขข้อผิดพลาดหรือปรับปรุงเพื่อให้รองรับกับความต้องการใหม่ ถ้าหากความสามารถในการบำรุงรักษาต่ำหมายถึงบำรุงรักษายากจะส่งผลให้สิ้นเปลืองทรัพยากรในการบำรุงรักษา ในงานวิจัยนี้ได้กล่าวถึงการวัดความสามารถในการบำรุงรักษาของซอฟต์แวร์เชิงเซอร์วิซ (Maintainability Assessment Of Service-Oriented Software) ในส่วนของความเสถียร (Stability) ไว้ดังนี้

1. Service Update Rate

เนื่องจากซอฟต์แวร์เชิงเซอร์วิซโดยทั่วไปนั้นมีการผูกเซอร์วิซแบบไดนามิก (Dynamic Binding) ดังนั้นเมื่อมีการปรับปรุงเซอร์วิซและทำการประกาศใช้งาน ผู้เรียกใช้บริการอาจจะต้องทำการผูกเซอร์วิซใหม่ด้วย ดังนั้นอัตราการปรับปรุงแก้ไขเซอร์วิซจึงมีผลต่อความเสถียร

2. Data Encapsulation

เดิมนั้นการประมวลผลแบบกระจาย (Distributed Computing) นั้นมีลักษณะเสมือนเป็นการเรียกใช้งานฟังก์ชันจากภายนอกผ่านเครือข่าย ในระบบซอฟต์แวร์แบบเดี่ยว (Stand-Alone) นั้นการเรียกใช้ฟังก์ชันจะมีผลต่อความเร็วในการตอบสนองหรือความเสถียรไม่มาก แต่ในระบบเครือข่ายการเรียกฟังก์ชันบ่อย ๆ ส่งผลต่อการทำงานของระบบอย่างมาก ดังนั้นโดยทั่วไประบบเซอร์วิซจึงมีการแลกเปลี่ยนข้อมูลขนาดใหญ่และครบถ้วนเพื่อลดจำนวนการเรียกใช้ฟังก์ชัน ขนาดของข้อมูลและความสมบูรณ์ของข้อมูลที่แลกเปลี่ยนกับเซอร์วิซจึงมีผลต่อความเสถียรของระบบ

3. Independence of Change

ในระบบซอฟต์แวร์เชิงเซอร์วิซอาจมีบางเซอร์วิซมีการเรียกใช้งานเซอร์วิซอื่น ดังนั้นหากมีความต้องการที่จะเปลี่ยนแปลงในจุดที่ไม่มีความเกี่ยวเนื่องของเซอร์วิซก็จะส่งผลกระทบน้อย

งานวิจัยนี้แสดงถึงปัจจัยที่มีผลต่อความเสถียรแต่ไม่ได้ระบุมาตรวัดความเสถียรโดยตรง และแนวทางเหล่านี้ยังแตกต่างจากแนวทางของผู้วิจัยที่ต้องการวัดค่าความเสถียรในขั้นตอนออกแบบโดยวัดจากการเปลี่ยนแปลงของวิสเดิล

2.2.5 A Controlled Experiment for Evaluating the Impact of Coupling on the Maintainability of Service-Oriented Software

Mikhail Perepletchikov และคณะ [11] ได้เสนองานวิจัยโดยเสนอการทดลองเพื่อวิเคราะห์ความสัมพันธ์ระหว่างความขึ้นต่อกัน (Coupling) ของการออกแบบซอฟต์แวร์เชิงเซอร์วิซกับความสามารถในการบำรุงรักษาของซอฟต์แวร์ในมุมมองของคุณลักษณะย่อยเช่น ความสามารถในการวิเคราะห์ (Analyzability), ความสามารถในการแก้ไข (Changeability) และ ความเสถียร (Stability) ซึ่งในการวิจัยนี้ ได้กล่าวถึงมาตรวัดคุณลักษณะย่อยของความสามารถในการบำรุงรักษาซึ่งอ้างอิงมาจาก ISO/IEC TR 9126-1:2001 ได้แก่

- Analyzability Metric: Failure Analysis Efficiency (FAE)

  FAE = Sum(T)/N

  โดยที่ T คือ ผลรวมของเวลาที่ใช้วิเคราะห์ข้อผิดพลาด

  N คือ จำนวนข้อผิดพลาด

FAE มีค่าได้มากกว่า 0 โดยที่หาก FAE มีค่าเข้าใกล้ 0 จะหมายถึงสามารถวิเคราะห์ปัญหาได้ง่าย

- Changeability Metric: Modification Complexity (MC)

  MC = Sum(T)/N

  โดยที่ T คือ ผลรวมของเวลาที่ใช้ในการแก้ไขเปลี่ยนแปลง

  N คือ จำนวนของส่วนที่ต้องเปลี่ยนแปลง

MC มีค่าได้มากกว่า 0 โดยที่หาก MC มีค่าเข้าใกล้ 0 จะหมายถึงสามารถเปลี่ยนแปลงได้ง่าย

- Stability Metric: Modification Impact Localization (MIL)

  MIL = A/B

  โดยที่ A คือ จำนวนของความล้มเหลวที่เกิดขึ้นหลังจากมีการแก้ไข

  B คือ จำนวนของส่วนที่แก้ไข

MIL มีค่าได้ตั้งแต่ 0 โดยที่หาก MIL มีค่าเข้าใกล้ 0 จะหมายถึงมีความเสถียรสูง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

งานวิจัยนี้เสนอมาตรวัดความขึ้นต่อกันของซอฟต์แวร์เชิงเซอร์วิซ (Service-Oriented Coupling Metrics) ที่จะนำมาวัดและเปรียบเทียบผลกับคุณลักษณะย่อยของความสามารถในการบำรุงรักษา จากการทดลองพบว่าเมื่อค่าความขึ้นต่อกันสูงจะส่งผลไม่ดีกับ Analyzability และ Changeability เมื่อเทียบกับซอฟต์แวร์เชิงเซอร์วิซที่มีค่าความขึ้นต่อกันต่ำ แต่ Stability นั้นไม่มีความเกี่ยวเนื่องกับ Service-Oriented Coupling Metrics ที่ผู้วิจัยนำเสนอ

การวัดความเสถียรที่งานวิจัยดังกล่าวนำมาใช้นั้นเป็นการวัดความเสถียรจากการแก้ไขและปรับปรุงโปรแกรมและจำนวนความล้มเหลวที่เกิดขึ้นหลังจากมีการแก้ไขและปรับ ปรุงโปรแกรมไปแล้ว ซึ่งต่างจากแนวทางในการวิจัยที่ต้องการวัดความเสถียรในขั้นตอนออกแบบวิสเดิล

2.2.6 A Quantitative Measurement and Validation of Granularity in Service Oriented Architecture

T. Karthikeyan และ J. Geetha [12] นำเสนอการวัดและการทวนสอบของขนาดและความสมบูรณ์ในสถาปัตยกรรมเชิงเซอร์วิซ สืบเนื่องมาจากขนาดของเซอร์วิซเป็นคุณลักษณะพื้นฐานของการออกแบบเซอร์วิซ ดังนั้นจึงได้หาวิธีในการวัดขนาดของเซอร์วิซเพื่อที่จะนำค่าขนาดไปพิจารณาการออกแบบเซอร์วิซต่อไป โดยส่วนหนึ่งของการงานวิจัยได้นำเสนอตัววัดขนาดของส่วนต่อประสาน โดยสามารถวัดได้ดังนี้

ขนาดของส่วนต่อประสาน (Interface Granularity: IG) คำนวณได้จากสูตร

$$IG = \frac{\sum_{i=1}^{nip} Wi}{nip} + \frac{\sum_{i=1}^{nop} Wi}{nop}$$

โดยที่ nip คือ จำนวนของอินพุตพารามิเตอร์

nop คือ จำนวนของเอาต์พุตพารามิเตอร์

Wi คือ ค่าน้ำหนักของพารามิเตอร์แต่ละตัวตามประเภทข้อมูลของพารามิเตอร์ โดยที่ void มีค่าน้ำหนักเป็น 0.0, primitive type มีค่าน้ำหนักเป็น 0.25, user-defined type มีค่าน้ำหนักเป็น 0.5, complex type มีค่าน้ำหนักเป็น 1

ขนาดของส่วนต่อประสานมีค่าระหว่าง (0,2] โดยถ้ามีค่าสูงหมายถึงส่วนต่อประสานมีขนาดใหญ่และสมบูรณ์

เนื่องจากยังไม่มีตัววัดความเสถียรของเซอร์วิซที่ใกล้เคียงกับตัววัดความเสถียรที่ผู้วิจัยเสนอได้โดยตรง ผู้วิจัยจึงนำขนาดของส่วนต่อประสานมาเป็นส่วนหนึ่งในการประเมินแบบจำลองวัดความ

เสถียรของเซอร์วิซ โดยมีสมมติฐานว่าเซอร์วิซที่มีขนาดใหญ่หรือมีฟังก์ชันครอบคลุมอยู่แล้วน่าจะไม่เปลี่ยนแปลงมากนักหรือมีความเสถียรเมื่อมีวิวัฒนาการ

2.2.7 Cohesion Metrics for Predicting Maintainability of Service-Oriented Software

M. Perepletchikov และคณะ [13] ได้เสนอมาตรวัดความเชื่อมติดกันสำหรับทำนายความสามารถในการบำรุงรักษาซอฟต์แวร์เชิงเซอร์วิซโดยมีมาตรวัดย่อยที่วัดค่าการเชื่อมติดกันในหลาย ๆ ด้าน ซึ่งหนึ่งในมาตรวัดย่อยคือมาตรวัดการเชื่อมติดกันของข้อมูลของส่วนต่อประสาน (Service Interface Data Cohesion: SIDC) ซึ่งสามารถหาค่าได้จาก

$$SIDC(s) = \frac{|Common(Param(so \, \epsilon \, SO(si_s))|}{totalParamTypes}$$

โดยที่ Common(Param(so ϵ SO(si_s)) คือ ฟังก์ชันที่ส่งกลับเซตของโอเปอเรชันของเซอร์วิซที่มีประเภทข้อมูลของพารามิเตอร์เป็นประเภทเดียวกัน

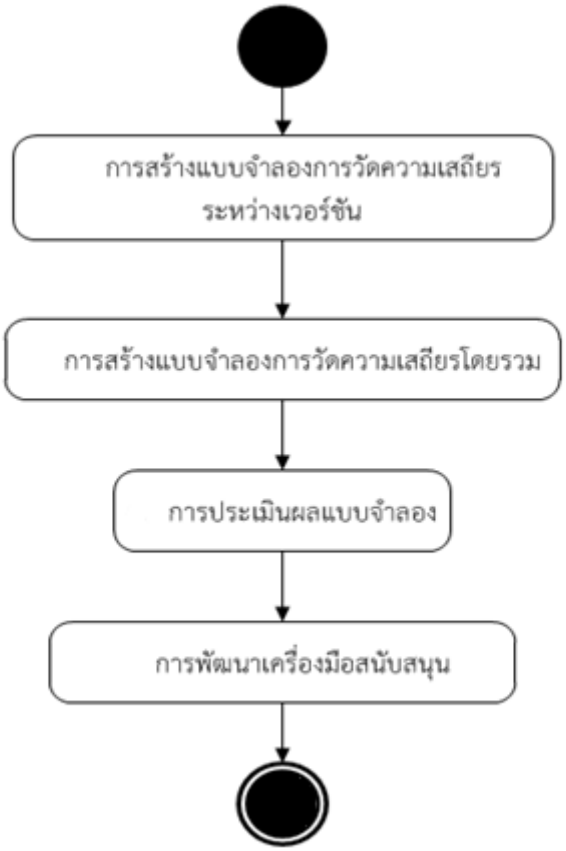
totalParamTypes คือ จำนวนประเภทข้อมูลของพารามิเตอร์ทั้งหมด

การเชื่อมติดกันของข้อมูลในส่วนต่อประสานมีค่าระหว่าง (0,1] โดยถ้ามีค่าสูงหมายถึงมีการเชื่อมติดกันของข้อมูลในส่วนต่อประสานมาก เนื่องจากโอเปอเรชันมีการใช้ประเภทข้อมูลร่วมกันมาก

ผู้วิจัยมีสมมติฐานว่าการออกแบบที่ดีควรมีการเชื่อมติดกัน (Cohesion) ของข้อมูลภายในส่วนต่อประสานสูง ถ้าภายในส่วนต่อประสานมีการออกแบบที่ดีอยู่แล้วก็จะส่งผลให้เกิดความเสถียรของเซอร์วิซเมื่อมีวิวัฒนาการ แต่อย่างไรก็ตามหากข้อมูลมีความเชื่อมติดกันสูง เมื่อจำเป็นต้องแก้ไขอาจมีผลกระทบต่อกันในวงกว้างเช่นกัน ด้วยสมมติฐานที่ได้กล่าวมาผู้วิจัยจึงได้นำมาตรวัดนี้มาทำการเปรียบเทียบกับความเสถียรของเซอร์วิซ

บทที่ 3
แบบจำลองการวัดความเสถียรสำหรับระบบเซอร์วิซ

จากที่มาและความสำคัญของปัญหา ผู้วิจัยได้ทำการศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องเพื่อหาวิธีการและสร้างแบบจำลองการวัดความเสถียรรวมทั้งสร้างเครื่องมือช่วยในการคำนวณและแสดงผลการวัด ภาพรวมขั้นตอนการทำงานเป็นดังภาพที่ 3.1 ซึ่งแต่ละขั้นตอนการดำเนินงานมีรายละเอียดดังนี้



ภาพที่ 3.1 ภาพรวมวิธีการดำเนินงาน

3.1 การสร้างแบบจำลองการวัดความเสถียรระหว่างเวอร์ชัน

จากงานวิจัยเรื่องการวัดความเสถียรของคลาสเชิงวัตถุโดยวิเคราะห์จากการเปลี่ยนแปลงคุณสมบัติของคลาส [4] ที่มีแนวคิดเกี่ยวกับการวัดความเสถียรโดยสนใจการเปลี่ยนแปลงของ

คุณสมบัติที่มีผลต่อความเสถียร ในงานวิจัยนี้ผู้วิจัยได้ประยุกต์แนวคิดของการวัดความเสถียรของคลาสเชิงวัตถุและได้แบ่งรูปแบบการวัดความเสถียรดังนี้

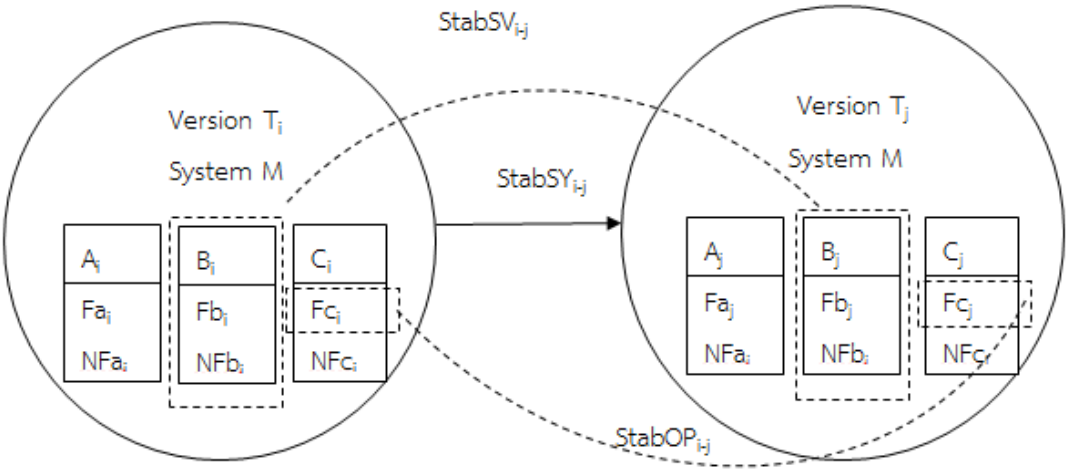
- ความเสถียรของโอเปอเรชันของเซอร์วิซระหว่างเวอร์ชัน i และ j
- ความเสถียรของเซอร์วิซเดี่ยวระหว่างเวอร์ชัน i และ j
- ความเสถียรของระบบเซอร์วิซทั้งระบบระหว่างเวอร์ชัน i และ j

จากภาพที่ 3.2 เป็นการแสดงระบบเซอร์วิซ M ในเวอร์ชันตั้งต้น T_i โดยในระบบเซอร์วิซ M ประกอบไปด้วยเซอร์วิซ A, B และ C ซึ่งในวิสเดิลของเซอร์วิซ A มีเอลิเมนต์เชิงฟังก์ชัน Fa และ เอลิเมนต์ที่ไม่ใช่เชิงฟังก์ชัน NFa, วิสเดิลของเซอร์วิซ B มีเอลิเมนต์เชิงฟังก์ชัน Fb และ เอลิเมนต์ที่ไม่ใช่เชิงฟังก์ชัน NFb และ วิสเดิลของเซอร์วิซ C มีเอลิเมนต์เชิงฟังก์ชัน Fc และ เอลิเมนต์ที่ไม่ใช่เชิงฟังก์ชัน NFc ต่อมาระบบเซอร์วิซนี้มีการเปลี่ยนแปลงเซอร์วิซต่าง ๆ จนกลายเป็นเวอร์ชัน T_j

- เมื่อพิจารณาการเปลี่ยนแปลงของเอลิเมนต์เชิงฟังก์ชันซึ่งมีความเกี่ยวข้องกับโอเปอเรชันของเซอร์วิซจากเวอร์ชัน T_i ไปยัง T_j ค่าความแตกต่างนั้นคือความเสถียรของโอเปอเรชัน (Operation Stability: StabOP) ระหว่างเวอร์ชัน i กับ j
- เมื่อพิจารณาการเปลี่ยนแปลงของเซอร์วิซใดเซอร์วิซหนึ่งจากเวอร์ชัน T_i ไปยัง T_j ค่าความแตกต่างนั้นคือความเสถียรของเซอร์วิซ (Service Stability: StabSV) ระหว่างเวอร์ชัน i กับ j
- เมื่อพิจารณาการเปลี่ยนแปลงของระบบเซอร์วิซทั้งระบบจากเวอร์ชัน T_i ไปยัง T_j ค่าความแตกต่างนั้นคือความเสถียรของระบบ (System Stability: StabSY) ระหว่างเวอร์ชัน i กับ j

ขั้นตอนการคำนวณค่าความเสถียรระหว่างเวอร์ชันนั้นแบ่งเป็น 3 ขั้นตอน ในขั้นตอนแรกจะต้องทำการนิยามความหมายของความเสถียร ส่วนที่สองคือการกำหนดปัจจัยที่ส่งผลต่อความเสถียร และส่วนที่สามคือวิธีการคำนวณค่าความเสถียร

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



ภาพที่ 3.2 การวัดความเสถียรแต่ละประเภท

3.1.1 ความเสถียรของโอเปอเรชัน

ความเสถียรของโอเปอเรชันของเซอร์วิซระหว่างเวอร์ชัน i และ j หมายถึงไม่มีการเปลี่ยนแปลงค่าในแอตทริบิวต์ต่าง ๆ ของเอลิเมนต์เชิงฟังก์ชันซึ่งมีความเกี่ยวข้องกับโอเปอเรชันได้แก่

- เอลิเมนต์ <operation>
- เอลิเมนต์ <input>
- เอลิเมนต์ <output>
- เอลิเมนต์ <message>
- เอลิเมนต์ <part> (ที่เกี่ยวข้องกับโอเปอเรชันที่พิจารณา)
- เอลิเมนต์ <types> (ที่เกี่ยวข้องกับโอเปอเรชันที่พิจารณา)

จากนิยามของความเสถียรของโอเปอเรชันของเซอร์วิซระหว่างเวอร์ชัน i และ j สามารถกำหนดปัจจัยที่มีผลต่อความเสถียร, เงื่อนไขและการนับเพื่อวัดค่าความเสถียรได้ดังตารางที่ 3.1

ตารางที่ 3.1 ปัจจัยและเงื่อนไขการนับของการวัดความเสถียรของโอเปอเรชัน (เอลิเมนต์เชิงฟังก์ชัน)

	ปัจจัย		เงื่อนไข	การนับ
1	<operation>	Unchanged	ถ้าไม่มีการเปลี่ยนแปลงแอตทริบิวต์หรือค่าของแอตทริบิวต์ของ <operation> จากเวอร์ชัน i ไปยัง j	+1
2	<input>	Unchanged	ถ้าไม่มีการเปลี่ยนแปลงแอตทริบิวต์หรือค่าของแอตทริบิวต์ของ <input> จากเวอร์ชัน i ไปยัง j	+1
3	<output>	Unchanged	ถ้าไม่มีการเปลี่ยนแปลงแอตทริบิวต์หรือค่าของแอตทริบิวต์ของ<output>จากเวอร์ชัน i ไปยัง j	+1
4	<message>	Unchanged	ถ้าไม่มีการเปลี่ยนแปลงแอตทริบิวต์หรือค่าของแอตทริบิวต์ของ<message> จากเวอร์ชัน i ไปยัง j	+1
5	<part>	Unchanged	ถ้าไม่มีการเปลี่ยนแปลงแอตทริบิวต์หรือค่าของแอตทริบิวต์ของ <part> จากเวอร์ชัน i ไปยัง j	+1
6	<types>	Unchanged	ถ้าไม่มีการเปลี่ยนแปลงแอตทริบิวต์หรือค่าของแอตทริบิวต์ของ Data type tag ต่าง ๆ ภายใต้ <types> จากเวอร์ชัน i ไปยัง j	+1

ในการคำนวณค่าความเสถียรนั้น ลำดับแรกต้องคำนวณความเสถียรของแต่ละเอลิเมนต์เชิงฟังก์ชันโดยประยุกต์จากสมการการคำนวณความเสถียรของคลาสเชิงวัตถุโดยวิเคราะห์จากการเปลี่ยนแปลงคุณสมบัติ [4] ดังนั้นจะได้

$$\mathbf{Stab}_{\mathbf{FElement}} = \mathbf{Unchanged}_{\mathbf{FElement}} / \mathbf{Number}_{\mathbf{FElement}} \quad (1)$$

โดยที่ $\mathbf{Stab}_{\mathbf{FElement}}$ คือความเสถียรของเอลิเมนต์เชิงฟังก์ชัน มีค่าอยู่ในช่วง [0,1]

$\mathbf{Unchanged}_{\mathbf{FElement}}$ คือจำนวนเอลิเมนต์เชิงฟังก์ชันที่ไม่มีการเปลี่ยนแปลง

$\mathbf{Number}_{\mathbf{FElement}}$ คือจำนวนเอลิเมนต์เชิงฟังก์ชัน

หลังจากได้ค่าความเสถียรของแต่ละเอลิเมนต์มาแล้ว จะนำค่านั้นมาคำนวณความเสถียรของโอเปอเรชันต่อโดยประยุกต์จากการคำนวณความเสถียรของคลาสเชิงวัตถุโดยวิเคราะห์จากการเปลี่ยนแปลงคุณสมบัติ [4] ทำให้ได้สมการคำนวณความเสถียรของโอเปอเรชันระหว่าง 2 เวอร์ชันเป็น

$$StabOP_{i-j} = \frac{Stab_{Operation}+Stab_{Input}+Stab_{Output}+Stab_{Message}+Stab_{Part}+Stab_{Types}}{FElement_{OPERATION}} \quad (2)$$

โดยที่ $StabOP_{i-j}$ คือความเสถียรของโอเปอเรชันระหว่างเวอร์ชัน i กับ j

$Stab_{Operation}$ คือความเสถียรของเอลิเมนต์ <operation> ที่พิจารณา

$Stab_{Input}$ คือความเสถียรของเอลิเมนต์ <input> ภายใต้โอเปอเรชันที่พิจารณา

$Stab_{Output}$ คือความเสถียรของเอลิเมนต์ <output> ภายใต้โอเปอเรชันที่พิจารณา

$Stab_{Message}$ คือความเสถียรของเอลิเมนต์ <message> ภายใต้โอเปอเรชันที่พิจารณา

$Stab_{Part}$ คือความเสถียรของเอลิเมนต์ <part> ภายใต้โอเปอเรชันที่พิจารณา

$Stab_{Types}$ คือความเสถียรของเอลิเมนต์ <types> ภายใต้โอเปอเรชันที่พิจารณา

$FElement_{OPERATION}$ คือจำนวนประเภทเอลิเมนต์เชิงฟังก์ชันซึ่งเกี่ยวข้องกับโอเปอเรชันที่ใช้คำนวณ (ในสมการนี้คือ 6)

ตัวอย่างที่ 3.1 การคำนวณค่าความเสถียรของโอเปอเรชัน

สมมติให้บางส่วนของวิสเดิลในเวอร์ชันที่ 0 (เวอร์ชันตั้งต้น) เป็นดังภาพที่ 3.3 ซึ่งโอเปอเรชัน “Add” ให้บริการบวกเลข 2 จำนวน จากนั้นมีความต้องการเปลี่ยนแปลงเป็นเวอร์ชันที่ 1 โดยเปลี่ยนแปลงประเภทของข้อมูลจากโฟลต (Float) ไปเป็นดับเบิล (Double) ในประเภทข้อมูล “MathInput” ดังภาพที่ 3.4 ซึ่งสามารถคำนวณหาค่าความเสถียรได้ดังตารางที่ 3.2

ตารางที่ 3.2 การคำนวณหาค่าความเสถียรของโอเปอเรชันระหว่างเวอร์ชัน 0 และ 1

ปัจจัย	จำนวนทั้งหมด	จำนวน Unchanged	$Stab_{FElement}$	ค่า
<operation>	1	1	$Stab_{Operation}$	1/1
<input>	1	1	$Stab_{Input}$	1/1
<output>	1	1	$Stab_{Output}$	1/1
<message>	2	2	$Stab_{Message}$	2/2
<part>	2	2	$Stab_{Part}$	2/2
<types>	5	3	$Stab_{Types}$	3/5

ดังนั้น $StabOP(Add)_{0-1} = (1+1+1+1+1+(3/5))/6 = 0.93$

```
...  <types>
   <xs:schema targetNamespace="http://example.org/math/types/"
   xmlns="http://example.org/math/types/"
   elementFormDefault="unqualified" attributeFormDefault="unqualified">
      <xs:complexType name="MathInput">
        <xs:sequence>
          <xs:element name="x" type="xs:float"/> //type 3
          <xs:element name="y" type="xs:float "/> // type 4
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="MathOutput">
        <xs:sequence>
          <xs:element name="result" type="xs:double"/> // type 5
        </xs:sequence>
      </xs:complexType>
      <xs:element name="Add" type="MathInput"/> // type 1
      <xs:element name="AddResponse" type="MathOutput"/> // type 2
    </xs:schema> </types>
  <message name="AddMessage">// message 1
    <part name="parameters" element="ns:Add"/>  // part 1
  </message>
  <message name="AddResponseMessage"> // message 2
    <part name="parameters" element="ns:AddResponse"/>  //part 2
  </message>
  <portType name="MathInterface">
    <operation name="Add"> // operation 1
      <input message="y:AddMessage"/> // input 1
      <output message="y:AddResponseMessage"/>  // output 1
    </operation>
  </portType>
```

ภาพที่ 3.3 ตัวอย่างบางส่วนวิสเดิลของเซอร์วิซ “MathService” [8] ในเวอร์ชันตั้งต้น

```
... <xs:complexType name="MathInput">
      <xs:sequence>
        <xs:element name="x" type="xs:double"/>
        <xs:element name="y" type="xs:double"/>
      </xs:sequence>
    </xs:complexType>...
```

ภาพที่ 3.4 การเปลี่ยนแปลงประเภทข้อมูลของ “MathInput” ในเวอร์ชัน 1

3.1.2 ความเสถียรของเซอร์วิซเดี่ยว

ความเสถียรของเซอร์วิซเดี่ยวหมายถึงไม่มีการเปลี่ยนแปลงค่าของแอตทริบิวต์ในเอลิเมนต์ต่าง ๆ ทุกเอลิเมนต์ทั้งเชิงฟังก์ชันและไม่ใช่เชิงฟังก์ชันจากเวอร์ชัน i ไปยังเวอร์ชัน j ดังนั้น ในส่วนของเอลิเมนต์ที่ไม่ใช่เชิงฟังก์ชัน สามารถกำหนดปัจจัยที่มีผลต่อความเสถียร เงื่อนไขและการนับเพื่อวัดความเสถียรได้ดังตารางที่ 3.3

ตารางที่ 3.3 ปัจจัยและเงื่อนไขการนับของการวัดความเสถียรของเอลิเมนต์ที่ไม่ใช่เชิงฟังก์ชัน

	ปัจจัย		เงื่อนไข	การนับ
1	<portType>	Unchanged	ถ้าไม่มีการเปลี่ยนแปลงแอตทริบิวต์หรือค่าของแอตทริบิวต์ของ <portType> จากเวอร์ชัน i ไปยัง j	+1
2	<binding>	Unchanged	ถ้าไม่มีการเปลี่ยนแปลงแอตทริบิวต์, ค่าของแอตทริบิวต์, หรือ tag ใด ๆ ภายใต้ <binding> จากเวอร์ชัน i ไปยัง j	+1
3	<service>	Unchanged	ถ้าไม่มีการเปลี่ยนแปลงแอตทริบิวต์ม, ค่าของแอตทริบิวต์, หรือ tag ใด ๆ ภายใต้ <service> จากเวอร์ชัน i ไปยัง j	+1

จากนิยามในหัวข้อ 3.1.2 สามารถสร้างสมการคำนวณค่าความเสถียรของเอลิเมนต์ที่ไม่ใช่เชิงฟังก์ชันได้ดังนี้

$$\mathbf{Stab_{NFElement} = Unchanged_{NFElement} / Number_{NFElement}} \quad \textbf{(3)}$$

โดยที่ $Stab_{NFElement}$ คือความเสถียรของเอลิเมนต์ที่ไม่ใช่เชิงฟังก์ชัน

$Unchanged_{NFElement}$ คือจำนวนเอลิเมนต์ที่ไม่ใช่เชิงฟังก์ชันที่ไม่มีการเปลี่ยนแปลง

$Number_{NFElement}$ คือจำนวนเอลิเมนต์ที่ไม่ใช่เชิงฟังก์ชัน

จากสมการ (1) และ (3) หลังจากได้ค่าความเสถียรของแต่ละเอลิเมนต์แล้ว จะนำค่านั้นมาคำนวณความเสถียรของเซอร์วิซต่อโดยประยุกต์จากการคำนวณความเสถียรของคลาสเชิงวัตถุโดยวิเคราะห์จากการเปลี่ยนแปลงคุณสมบัติ [4] ทำให้ได้สมการคำนวณความเสถียรของเซอร์วิซระหว่าง 2 เวอร์ชันเป็น

$$StabSV_{i-j} = (Stab_{Operation}+Stab_{Input}+Stab_{Output}+Stab_{Message}+Stab_{Part} +Stab_{Types}+Stab_{porttype}+Stab_{Binding}+Stab_{Service}) / Element_{SERVICE} \quad (4)$$

โดยที่ $StabSV_{i-j}$ คือความเสถียรของเซอร์วิซระหว่างเวอร์ชัน i และ j

$Stab_{Operation}$ คือความเสถียรของเอลิเมนต์ <operation> ทั้งหมดในเซอร์วิซ

$Stab_{Input}$ คือความเสถียรของเอลิเมนต์ <input> ทั้งหมดในเซอร์วิซ

$Stab_{Output}$ คือความเสถียรของเอลิเมนต์ <output> ทั้งหมดในเซอร์วิซ

$Stab_{Message}$ คือความเสถียรของเอลิเมนต์ <message> ทั้งหมดในเซอร์วิซ

$Stab_{Part}$ คือความเสถียรของเอลิเมนต์ <part> ทั้งหมดในเซอร์วิซ

$Stab_{Types}$ คือความเสถียรของเอลิเมนต์ <types> ทั้งหมดในเซอร์วิซ

$Stab_{PortType}$ คือความเสถียรของเอลิเมนต์ <portType> ทั้งหมดในเซอร์วิซ

$Stab_{Binding}$ คือความเสถียรของเอลิเมนต์ <Binding> ทั้งหมดในเซอร์วิซ

$Stab_{Service}$ คือความเสถียรของเอลิเมนต์ <Service> ทั้งหมดในเซอร์วิซ

$Elements_{SERVICE}$ คือจำนวนประเภทเอลิเมนต์เชิงฟังก์ชันและไม่ใช่เชิงฟังก์ชันทั้งหมดในเซอร์วิซ

3.1.3 ความเสถียรของระบบเซอร์วิซ

ความเสถียรของระบบเซอร์วิซคือค่าเฉลี่ยความเสถียรของเซอร์วิซเดี่ยวทุกเซอร์วิซในระบบที่พิจารณาจากเวอร์ชัน i ไปยังเวอร์ชัน j

จากนิยามความเสถียรของระบบเซอร์วิซและการคำนวณค่าความเสถียรของเซอร์วิซเดี่ยวสามารถนำมาสร้างเป็นสมการคำนวณค่าความเสถียรของระบบเซอร์วิซได้ดังนี้

$$StabSY_{i-j} = (StabSV(1)_{i-j} + StabSV(2)_{i-j} +...+ StabSV(n)_{i-j})/ n \quad (5)$$

โดยที่ $StabSY_{i-j}$ คือความเสถียรของระบบเซอร์วิซระหว่างเวอร์ชัน i และ j มีค่าอยู่ในช่วง [0,1]

StabSV คือความเสถียรของแต่ละเซอร์วิซเดี่ยวในระบบระหว่างเวอร์ชัน i และ j

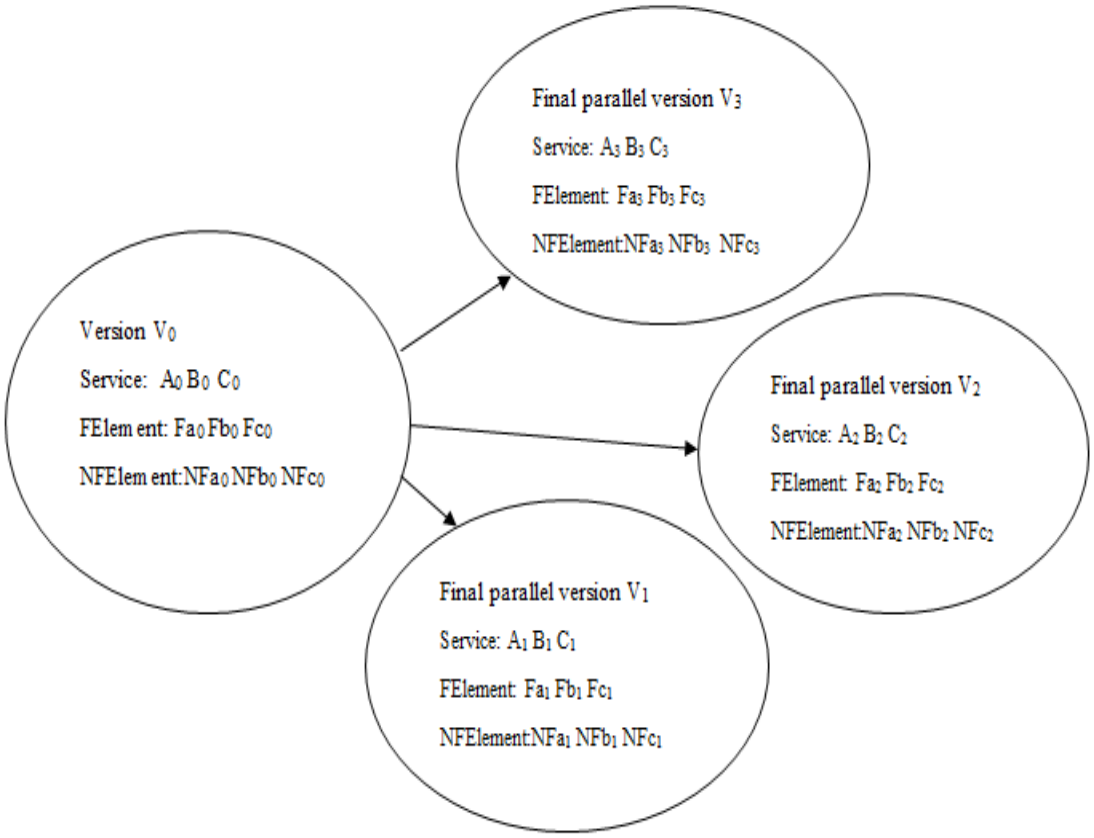
n คือจำนวนเซอร์วิซทั้งหมดในระบบเวอร์ชัน i

3.2 การสร้างแบบจำลองการวัดความเสถียรโดยรวม

เนื่องจากในการพัฒนาซอฟต์แวร์โดยทั่วไปนั้นจะมีการพัฒนาแบบคู่ขนานกล่าวคือ ผู้พัฒนาจะมีการนำซอฟต์แวร์จากเวอร์ชันตั้งต้นไปพัฒนาต่อโดยแตกแขนงเป็นเวอร์ชันคู่ขนานได้มากกว่า 1 เวอร์ชันดังภาพที่ 3.5

สำหรับกรณีที่มีเวอร์ชันคู่ขนาน 2 เวอร์ชันขึ้นไปนั้น จากงานวิจัยของ Kelly [5] สามารถแบ่งการพิจารณาความเสถียรโดยรวมได้ 2 มุมมองคือ 1) พิจารณาระหว่างเวอร์ชันสุดท้ายของเวอร์ชันคู่ขนาน 2) พิจารณาระหว่างเวอร์ชันตั้งต้น (Base Version) กับเวอร์ชันคู่ขนานสุดท้าย (Final Parallel version) โดยแบ่งการวัดค่าความเสถียรโดยรวมดังนี้

- ความเสถียรโดยรวมของโอเปอเรชัน
- ความเสถียรโดยรวมของเซอร์วิซเดี่ยว
- ความเสถียรโดยรวมของระบบเซอร์วิซ



ภาพที่ 3.5 ตัวอย่างเวอร์ชันคู่ขนาน

แต่เนื่องจากงานวิจัยนี้วัดค่าความเสถียรโดยพิจารณาจากการไม่เปลี่ยนแปลงของส่วนต่อประสานของเซอร์วิซซึ่งตรงกันข้ามกับแนวทางของ Kelly ซึ่งพิจารณาการเปลี่ยนแปลง (หรือระยะทาง) ระหว่างซอฟต์แวร์ต่างเวอร์ชันดังที่กล่าวไว้ในหัวข้อที่ 3.2 ดังนั้นวิธีการพิจารณาจึงต้องปรับเปลี่ยนให้สอดคล้องกับเจตนาเดิมคือ จากเดิมที่ต้องหาระยะทางที่มากที่สุดของลักษณะการออกแบบที่วัดได้ เช่น ผลต่างของจำนวนบรรทัดที่มากที่สุดในแต่ละคู่ของเวอร์ชัน ถ้าผลต่างจำนวนบรรทัดที่มากที่สุดนั้นมีค่าน้อย หมายถึงลักษณะการออกแบบนั้นโดยรวมแล้วมีความเสถียร แต่ค่าระยะทางที่งานวิจัยนี้นำเสนอ (ความเสถียร) นั้นคือการไม่เปลี่ยนแปลง (Unchanged) ซึ่งมีค่าตั้งแต่ 0 ถึง 1 โดยที่ 1 หมายถึงไม่เปลี่ยนแปลง ทำให้ต้องใช้ค่าความเสถียรที่น้อยที่สุดในแต่ละคู่ของเวอร์ชันเป็นค่าตัวแทน และค่าน้อยที่สุดนี้ ถ้ามีค่ามากจะหมายถึงมีความเสถียรโดยรวมสูง ดังนั้นคอร์ดที่สนใจคือ

$$\boldsymbol{\delta}_{Stab} = \min(Stab_{i-j}) \tag{6}$$

โดยที่ $Stab_{i-j}$ คือค่าความเสถียรที่วัดระหว่างเวอร์ชันคู่ขนานสุดท้าย i และ j แต่ละคู่, $i,j > 0$

$$\Delta_{Stab} = \min(Stab_{0-j}) \tag{7}$$

โดยที่ $Stab_{0-j}$ คือค่าความเสถียรที่วัดระหว่างเวอร์ชันตั้งต้น (เวอร์ชัน 0) และเวอร์ชันคู่ขนานสุดท้าย j แต่ละเวอร์ชัน, $j > 0$

เนื่องจากความเสถียรเป็นการวัดค่าระหว่างคู่ของเวอร์ชันและเป็นปริมาณที่มีทั้งขนาดและทิศทางกล่าวคือ $\mathbf{Stab_{i-j} \neq Stab_{j-i}}$ ดังนั้นเพื่อหาค่าความเสถียรที่น้อยที่สุดจึงต้องเลือกใช้ค่าที่น้อยที่สุดระหว่าง $\mathbf{Stab_{i-j}}$ และ $\mathbf{Stab_{j-i}}$ เป็นค่าความเสถียรระหว่างเวอร์ชัน i และ j ผู้วิจัยจะใช้หลักการนี้เพื่อคำนวณหาค่า $\boldsymbol{\delta}_{Stab}$

ความเสถียรโดยรวมนั้นจะต้องพิจารณาค่า Δ_{Stab} และ $\boldsymbol{\delta}_{Stab}$ ร่วมกันเพื่อแปลความหมายของความเสถียรโดยรวมในแต่ละมุมมองเช่น

- Δ_{Stab} มีค่ามากแต่ $\boldsymbol{\delta}_{Stab}$ มีค่าน้อยนั้นคือเซอร์วิซในเวอร์ชันคู่ขนานนั้นเปลี่ยนแปลงจากเวอร์ชันตั้งต้นไม่มากแต่มีความแตกต่างระหว่างเวอร์ชันคู่ขนานด้วยกันเองสูง ซึ่งอาจเป็นผลมาจากเซอร์วิซตั้งต้นออกแบบมาไม่ดีหรือเซอร์วิซตั้งต้นมีลักษณะเป็นเซอร์วิซพื้นฐาน (Common) นักออกแบบเซอร์วิซอาจจะต้องตรวจสอบว่าเซอร์วิซเวอร์ชันคู่ขนานนั้นให้บริการในเรื่องเดียวกันอยู่หรือไม่ ถ้าไม่ ก็ควรแยกออกไปตั้งเป็นเซอร์วิซที่ให้บริการเรื่องใหม่ เป็นต้น

- Δ_{Stab} มีค่าน้อยแต่ $\boldsymbol{\delta}_{Stab}$ มีค่ามากนั้นคือเซอร์วิซในเวอร์ชันคู่ขนานนั้นเปลี่ยนแปลงจากเวอร์ชันตั้งต้นมากแต่มีความแตกต่างระหว่างเวอร์ชันคู่ขนานด้วยกันเองน้อย อาจหมายถึงเซอร์วิซเวอร์ชันคู่ขนานมีความเหมือนกันสามารถยุบรวมโอเปอเรชันที่เหมือนกันแล้วนำไปตั้งต้นเป็นเซอร์วิซใหม่ได้
- Δ_{Stab} และ $\boldsymbol{\delta}_{Stab}$ มีค่าน้อยนั้นคือเซอร์วิซในเวอร์ชันคู่ขนานมีความเปลี่ยนแปลงจากเวอร์ชันตั้งต้นมากและมีความแตกต่างระหว่างเซอร์วิซเวอร์ชันคู่ขนานด้วยกันเองมากเช่นกัน อาจหมายถึงเซอร์วิซเวอร์ชันตั้งต้นและแต่ละเซอร์วิซเวอร์ชันคู่ขนานไม่มีความเกี่ยวข้องกันเลยซึ่งไม่ควรเป็นเซอร์วิซเดียวกัน อาจจะเกิดจากการตัดสินใจผิดพลาดในขั้นตอนออกแบบ ซึ่งนักออกแบบเซอร์วิซควรประเมินการออกแบบ หากเซอร์วิซไม่มีความเกี่ยวข้องกัน ควรแยกออกเป็นคนละเซอร์วิซ
- Δ_{Stab} และ $\boldsymbol{\delta}_{Stab}$ มีค่ามากนั้นคือเซอร์วิซในเวอร์ชันคู่ขนานมีความเปลี่ยนแปลงจากเวอร์ชันตั้งต้นเล็กน้อยและมีความแตกต่างระหว่างเซอร์วิซเวอร์ชันคู่ขนานด้วยกันเองเล็กน้อยเช่นกัน หมายถึงระบบเซอร์วิซออกแบบได้ดีตั้งแต่ต้นทำให้เกิดการเปลี่ยนแปลงไม่มากเมื่อมีวิวัฒนาการ

3.2.1 ความเสถียรโดยรวมของโอเปอเรชัน

ความเสถียรโดยรวมของโอเปอเรชัน (Overall_StabOP) เกี่ยวข้องกับพฤติกรรมการเปลี่ยนแปลงของส่วนต่อประสานในโอเปอเรชันที่สนใจโดยจะพิจารณาใน 2 มุมมองคือ 1) พิจารณาค่าความเสถียรระหว่างเวอร์ชันตั้งต้น (Base Version) กับเวอร์ชันคู่ขนานสุดท้าย ในส่วนนี้ต้องคำนวณหาค่า Δ_{Stab} 2) พิจารณาค่าความเสถียรระหว่างเวอร์ชันคู่ขนานสุดท้าย ในส่วนนี้จะต้องคำนวณหาค่า $\boldsymbol{\delta}_{Stab}$ จากภาพที่ 3.5 สามารถคำนวณค่าความเสถียรโดยรวมของโอเปอเรชัน a ของเซอร์วิซ A ซึ่งเกี่ยวข้องกับเอลิเมนต์เชิงฟังก์ชัน Fa ได้ดังนี้

1. กำหนดสมาชิกที่เกี่ยวข้องกับการพิจารณาหาค่า Δ_{Stab}

   จากภาพที่ 3.5 ความเสถียรโดยรวมของโอเปอเรชัน a ของเซอร์วิซมีสมาชิกที่เกี่ยวข้องกับการพิจารณาหาค่า Δ_{Stab} ได้แก่

   - โอเปอเรชัน a_0 ซึ่งเกี่ยวข้องกับเอลิเมนต์เชิงฟังก์ชัน Fa_0 ของเวอร์ชันตั้งต้น V_0
   - โอเปอเรชัน a_1 ซึ่งเกี่ยวข้องกับเอลิเมนต์เชิงฟังก์ชัน Fa_1 ของเวอร์ชันคู่ขนานสุดท้าย V_1
   - โอเปอเรชัน a_2 ซึ่งเกี่ยวข้องกับเอลิเมนต์เชิงฟังก์ชัน Fa_2 ของเวอร์ชันคู่ขนานสุดท้าย V_2

- โอเปอเรชัน a_3 ซึ่งเกี่ยวข้องกับเอลิเมนต์เชิงฟังก์ชัน Fa_3 ของเวอร์ชันคู่ขนานสุดท้าย V_3

2. คำนวณค่าระยะทางระหว่างโอเปอเรชัน a จากเวอร์ชันตั้งต้นไปยังแต่ละเวอร์ชันคู่ขนานสุดท้ายซึ่งหมายถึงค่า Operation Stability (StabOP) ของโอเปอเรชัน a ดังต่อไปนี้
   - StabOP ของโอเปอเรชัน a ระหว่างเวอร์ชันตั้งต้น V_0 และเวอร์ชันคู่ขนานสุดท้าย V_1 คือ $StabOP(a)_{0\text{-}1}$
   - StabOP ของโอเปอเรชัน a ระหว่างเวอร์ชันตั้งต้น V_0 และเวอร์ชันคู่ขนานสุดท้าย V_2 คือ $StabOP(a)_{0\text{-}2}$
   - StabOP ของโอเปอเรชัน a ระหว่างเวอร์ชันตั้งต้น V_0 และเวอร์ชันคู่ขนานสุดท้าย V_3 คือ $StabOP(a)_{0\text{-}3}$

3. เมื่อได้ค่าระยะทางของแต่ละคู่ระหว่างเวอร์ชันตั้งต้นกับเวอร์ชันคู่ขนานสุดท้าย ต่อไปก็นำค่าที่คำนวณได้จากข้อ 2 มาคำนวณหาค่า Δ_{Stab}

$$\Delta_{StabOP(a)} = \min(StabOP(a)_{0\text{-}j}) \text{ , } j > 0 \qquad (8)$$

4. กำหนดสมาชิกที่เกี่ยวข้องกับการพิจารณาหาค่า $\boldsymbol{\delta}_{stab}$

   จากภาพที่ 3.5 ความเสถียรโดยรวมของโอเปอเรชัน a ของเซอร์วิซมีสมาชิกที่เกี่ยวข้องกับการพิจารณาหาค่า $\boldsymbol{\delta}_{Stab}$ ได้แก่
   - โอเปอเรชัน a_1 ซึ่งเกี่ยวข้องกับเอลิเมนต์เชิงฟังก์ชัน Fa_1 ของเวอร์ชันคู่ขนานสุดท้าย V_1
   - โอเปอเรชัน a_2 ซึ่งเกี่ยวข้องกับเอลิเมนต์เชิงฟังก์ชัน Fa_2 ของเวอร์ชันคู่ขนานสุดท้าย V_2
   - โอเปอเรชัน a_3 ซึ่งเกี่ยวข้องกับเอลิเมนต์เชิงฟังก์ชัน Fa_3 ของเวอร์ชันคู่ขนานสุดท้าย V_3

5. คำนวณค่าระยะทางระหว่างโอเปอเรชัน a ของแต่ละเวอร์ชันคู่ขนานสุดท้ายซึ่งหมายถึงค่า Operation Stability (StabOP) ของโอเปอเรชัน a ดังต่อไปนี้

- StabOP ของโอเปอเรชัน a ระหว่างเวอร์ชันคู่ขนานสุดท้าย V_1 และเวอร์ชันคู่ขนานสุดท้าย V_2 คือ ค่าที่น้อยที่สุดระหว่าง $StabOP(a)_{1-2}$ และ $StabOP(a)_{2-1}$
- StabOP ของโอเปอเรชัน a ระหว่างเวอร์ชันคู่ขนานสุดท้าย V_1 และเวอร์ชันคู่ขนานสุดท้าย V_3 คือ ค่าที่น้อยที่สุดระหว่าง $StabOP(a)_{1-3}$ และ $StabOP(a)_{3-1}$
- StabOP ของโอเปอเรชัน a ระหว่างเวอร์ชันคู่ขนานสุดท้าย V_2 และเวอร์ชันคู่ขนานสุดท้าย V_3 คือ ค่าที่น้อยที่สุดระหว่าง $StabOP(a)_{2-3}$ และ $StabOP(a)_{3-2}$

6. เมื่อได้ค่าระยะทางของแต่ละคู่ของเวอร์ชันคู่ขนานสุดท้ายแล้ว ต่อไปก็นำค่าที่คำนวณได้จากข้อ 5 มาคำนวณหาค่า δ_{Stab}

$$\delta_{StabOP(a)} = \min(StabOP(a)_{i-j}) \text{ , } i,j > 0 \quad (9)$$

ตัวอย่างที่ 3.2 การคำนวณความเสถียรโดยรวมของโอเปอเรชัน

จากเวอร์ชันตั้งต้นในภาพที่ 3.3 ถ้าให้ภาพที่ 3.4 เป็นเวอร์ชันคู่ขนานสุดท้าย V_1 และมีการปรับปรุงจากเวอร์ชันตั้งต้นอีกไปเป็นเวอร์ชัน V_2 ดังภาพที่ 3.6 และเวอร์ชัน V_3 ดังภาพที่ 3.7

```
<message name="AddMessage2">
  <part name="parameters" element="ns:Add"/>  </message>
<message name="AddResponseMessage">
  <part name="parameters" element="ns:AddResponse"/>  </message>
<portType name="MathInterface">
  <operation name="Add">
    <input message="y:AddMessage2"/> <output message="y:AddResponseMessage"/>
  </operation>
```

ภาพที่ 3.6 วิสเดิลที่ถูกเปลี่ยนแปลงเป็นเวอร์ชันคู่ขนานสุดท้าย V_2

```
... <portType name="MathInterface">
  <operation name="Add2">
    <input message="y:AddMessage"/> <output message="y:AddResponseMessage"/> ...
```

ภาพที่ 3.7 วิสเดิลที่ถูกเปลี่ยนแปลงเป็นเวอร์ชันคู่ขนานสุดท้าย V_3

จากภาพที่ 3.6 สามารถคำนวณความเสถียรของโอเปอเรชัน “Add” ระหว่างเวอร์ชันตั้งต้นและเวอร์ชันคู่ขนานสุดท้าย 1.2 ได้ดังตารางที่ 3.4 โดยใช้สมการ (1)

ตารางที่ 3.4 ค่าความเสถียรของแต่ละเอลิเมนต์เชิงฟังก์ชันของเวอร์ชันตั้งต้นและเวอร์ชันคู่ขนานสุดท้าย V_2

ปัจจัย	จำนวนทั้งหมด	จำนวน Unchanged	$Stab_{FElement}$	ค่า
<operation>	1	1	$Stab_{Operation}$	1/1
<input>	1	0	$Stab_{Input}$	0/1
<output>	1	1	$Stab_{Output}$	1/1
<message>	2	1	$Stab_{Message}$	1/2
<part>	2	2	$Stab_{Part}$	2/2
<types>	5	5	$Stab_{Types}$	5/5

ดังนั้น $\mathbf{StabOP(Add)_{0\text{-}2}} = (1+0+1+0.5+1+1)/6 = 0.75$

จากภาพที่ 3.7 สามารถคำนวณความเสถียรของโอเปอเรชัน “Add” ระหว่างเวอร์ชันตั้งต้นและเวอร์ชันคู่ขนานสุดท้าย 3 ได้ดังดังตารางที่ 3.5 โดยใช้สมการ (1)

ตารางที่ 3.5 ค่าความเสถียรของแต่ละเอลิเมนต์เชิงฟังก์ชันของเวอร์ชันตั้งต้นและเวอร์ชันคู่ขนานสุดท้าย V_3

ปัจจัย	จำนวนทั้งหมด	จำนวน Unchanged	$Stab_{FElement}$	ค่า
<operation>	1	0	$Stab_{Operation}$	0/1
<input>	1	1	$Stab_{Input}$	1/1
<output>	1	1	$Stab_{Output}$	1/1
<message>	2	2	$Stab_{Message}$	2/2
<part>	2	2	$Stab_{Part}$	2/2
<types>	5	5	$Stab_{Types}$	5/5

ดังนั้น $\mathbf{StabOP(Add)_{0\text{-}1.3}} = (0+1+1+1+1+1)/6 = 0.83$

และจากตัวอย่างที่ 3.1 ถ้าให้ภาพที่ 3.4 เป็นเวอร์ชันคู่ขนานสุดท้าย V_1 จะได้ค่าความเสถียรระหว่างเวอร์ชัน 0 และ 1 คือ $StabOP(Add)_{0\text{-}1}$ ซึ่งมีค่า 0.93 ดังนั้นเมื่อได้ค่าของความเสถียรของโอเปอเรชัน Add ระหว่างเวอร์ชันตั้งต้นไปยังแต่ละเวอร์ชันคู่ขนานสุดท้ายแล้วก็จะสามารถหาความเสถียรโดยรวมของโอเปอเรชัน (Overall_StabOP) ได้โดยใช้สมการ (8) และ (9) ดังนี้

$$\Delta_{StabOP(Add)} = \min(StabOP(Add)_{0\text{-}1.1}, StabOP(Add)_{0\text{-}1.2}, StabOP(Add)_{0\text{-}1.3})$$
$$= \min(0.93, 0.75, 0.83) = 0.75$$

$$\boldsymbol{\delta}_{StabOP(Add)} = \min(\min(StabOP(Add)_{1.1\text{-}1.2}, StabOP(Add)_{1.2\text{-}1.1}),$$
$$\min(StabOP(Add)_{1.1\text{-}1.3}, StabOP(Add)_{1.3\text{-}1.1}),$$
$$\min(StabOP(Add)_{1.2\text{-}1.3}, StabOP(Add)_{1.3\text{-}1.2}))$$
$$= \min(0.68, 0.58, 0.77) = 0.58$$

3.2.2 ความเสถียรโดยรวมของเซอร์วิซเดี่ยว

ความเสถียรโดยรวมของเซอร์วิซเดี่ยว (Overall_StabSV) เกี่ยวข้องกับพฤติกรรมการเปลี่ยนแปลงของส่วนต่อประสานในเซอร์วิซที่สนใจ โดยจะพิจารณาใน 2 มุมมองคือ 1) พิจารณาค่าความเสถียรระหว่างเวอร์ชันตั้งต้น (Base Version) กับเวอร์ชันคู่ขนานสุดท้าย ในส่วนนี้ต้องคำนวณหาค่า Δ_{Stab} 2) พิจารณาค่าความเสถียรระหว่างเวอร์ชันคู่ขนานสุดท้าย ในส่วนนี้จะต้องคำนวณหาค่า $\boldsymbol{\delta}_{Stab}$ จากภาพที่ 3.5 สามารถคำนวณค่าความเสถียรโดยรวมของเซอร์วิซ A ได้ดังนี้

1. กำหนดสมาชิกที่เกี่ยวข้องกับการพิจารณาหาค่า Δ_{Stab}

   จากภาพที่ 3.5 ความเสถียรโดยรวมของเซอร์วิซ A มีสมาชิกที่เกี่ยวข้องกับการพิจารณาหาค่า Δ_{Stab} ได้แก่

   - เซอร์วิซ A_0 ของเวอร์ชันตั้งต้น V_0
   - เซอร์วิซ A_1 ของเวอร์ชันคู่ขนานสุดท้าย V_1
   - เซอร์วิซ A_2 ของเวอร์ชันคู่ขนานสุดท้าย V_2
   - เซอร์วิซ A_3 ของเวอร์ชันคู่ขนานสุดท้าย V_3

2. คำนวณค่าระยะทางระหว่างเซอร์วิซ A จากเวอร์ชันตั้งต้นไปยังแต่ละเวอร์ชันคู่ขนานสุดท้าย ซึ่งหมายถึงค่า Service Stability (StabSV) ของเซอร์วิซ A ดังต่อไปนี้

   - StabSV ของเซอร์วิซ A ระหว่างเวอร์ชันตั้งต้น V_0 และเวอร์ชันคู่ขนานสุดท้าย V_1 คือ $StabSV(A)_{0\text{-}1}$

- StabSV ของเซอร์วิซ A ระหว่างเวอร์ชันตั้งต้น V_0 และเวอร์ชันคู่ขนานสุดท้าย V_2 คือ $StabSV(A)_{0-2}$
- StabSV ของเซอร์วิซ A ระหว่างเวอร์ชันตั้งต้น V_0 และเวอร์ชันคู่ขนานสุดท้าย V_3 คือ $StabSV(A)_{0-3}$

3. เมื่อได้ค่าระยะทางของแต่ละคู่ระหว่างเวอร์ชันตั้งต้นกับเวอร์ชันคู่ขนานสุดท้าย ต่อไปก็นำค่าที่คำนวณได้จากข้อ 2 มาคำนวณหาค่า Δ_{Stab}

$$\Delta_{StabSV(A)} = \min(StabSV(A)_{0-j}) \text{ , } j > 0 \quad (10)$$

4. กำหนดสมาชิกที่เกี่ยวข้องกับการพิจารณาหาค่า δ_{Stab}

จากภาพที่ 3.5 ความเสถียรโดยรวมของเซอร์วิซ A มีสมาชิกที่เกี่ยวข้องกับการพิจารณาหาค่า δ_{Stab} ได้แก่

- เซอร์วิซ A_1 ของเวอร์ชันคู่ขนานสุดท้าย V_1
- เซอร์วิซ A_2 ของเวอร์ชันคู่ขนานสุดท้าย V_2
- เซอร์วิซ A_3 ของเวอร์ชันคู่ขนานสุดท้าย V_3

5. คำนวณค่าระยะทางระหว่างเซอร์วิซ A ของแต่ละเวอร์ชันคู่ขนานสุดท้าย ซึ่งหมายถึง Service Stability (StabSV) ของเซอร์วิซ A ดังต่อไปนี้

- StabSV ของเซอร์วิซ A ระหว่างเวอร์ชันคู่ขนานสุดท้าย V_1 และเวอร์ชันคู่ขนานสุดท้าย V_2 คือค่าที่น้อยที่สุดระหว่าง $StabSV(A)_{1-2}$ และ $StabSV(A)_{2-1}$
- StabSV ของเซอร์วิซ A ระหว่างเวอร์ชันคู่ขนานสุดท้าย V_1 และเวอร์ชันคู่ขนานสุดท้าย V_3 คือค่าที่น้อยที่สุดระหว่าง $StabSV(A)_{1-3}$ และ $StabSV(A)_{3-1}$
- StabSV ของเซอร์วิซ A ระหว่างเวอร์ชันคู่ขนานสุดท้าย V_2 และเวอร์ชันคู่ขนานสุดท้าย V_3 คือค่าที่น้อยที่สุดระหว่าง $StabSV(A)_{2-3}$ และ $StabSV(A)_{3-2}$

6. เมื่อได้ค่าระยะทางของแต่ละคู่ของเวอร์ชันคู่ขนานสุดท้ายแล้ว ต่อไปก็นำค่าที่คำนวณได้จากข้อ 5 มาคำนวณหาค่า δ_{Stab}

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

$$\delta_{StabSV(A)} = \min(StabSV(A)_{i-j}) \text{ , } i,j > 0 \quad (11)$$

3.2.3 ความเสถียรโดยรวมของระบบเซอร์วิซ

ความเสถียรโดยรวมของระบบเซอร์วิซ (Overall_StabSY) เกี่ยวข้องกับพฤติกรรมการเปลี่ยนแปลงของส่วนต่อประสานทั้งระบบ โดยจะพิจารณาใน 2 มุมมองคือ 1) พิจารณาค่าความเสถียรระหว่างเวอร์ชันตั้งต้น (Base Version) กับเวอร์ชันคู่ขนานสุดท้าย ในส่วนนี้ต้องคำนวณหาค่า Δ_{Stab} 2) พิจารณาค่าความเสถียรระหว่างเวอร์ชันคู่ขนานสุดท้าย ในส่วนนี้จะต้องคำนวณหาค่า δ_{Stab} จากภาพที่ 3.5 สามารถคำนวณค่าความเสถียรโดยรวมของระบบ ได้ดังนี้

1. กำหนดสมาชิกที่เกี่ยวข้องกับการพิจารณาหาค่า Δ_{Stab}

   จากภาพที่ 3.5 ความเสถียรโดยรวมของระบบเซอร์วิซมีสมาชิกที่เกี่ยวข้องกับการพิจารณาหาค่า Δ_{Stab} ได้แก่

   - ระบบในเวอร์ชันตั้งต้น V_0
   - ระบบในเวอร์ชันคู่ขนานสุดท้าย V_1
   - ระบบในเวอร์ชันคู่ขนานสุดท้าย V_2
   - ระบบในเวอร์ชันคู่ขนานสุดท้าย V_3

2. คำนวณค่าระยะทางระหว่างระบบจากเวอร์ชันตั้งต้นไปยังแต่ละเวอร์ชันคู่ขนานสุดท้าย ซึ่งหมายถึงค่า System Stability (StabSY) ของระบบดังต่อไปนี้

   - StabSY ของระบบระหว่างเวอร์ชันตั้งต้น V_0 กับเวอร์ชันคู่ขนานสุดท้าย V_1 คือ $StabSY_{0-1}$
   - StabSY ของระบบระหว่างเวอร์ชันตั้งต้น V_0 กับเวอร์ชันคู่ขนานสุดท้าย V_2 คือ $StabSY_{0-2}$
   - StabSY ของระบบระหว่างเวอร์ชันตั้งต้น V_0 กับเวอร์ชันคู่ขนานสุดท้าย V_3 คือ $StabSY_{0-3}$

3. เมื่อได้ค่าระยะทางของแต่ละคู่ระหว่างเวอร์ชันตั้งต้นกับเวอร์ชันคู่ขนานสุดท้าย ต่อไปก็นำค่าที่คำนวณได้จากข้อ 2 มาคำนวณหาค่า Δ_{Stab}

$$\Delta_{StabSY} = \min(StabSY_{0-j}) \text{ , } j > 0 \quad (12)$$

4. กำหนดสมาชิกที่เกี่ยวข้องกับการพิจารณาหาค่า δ_{Stab}

   จากภาพที่ 3.5 ความเสถียรโดยรวมของระบบเซอร์วิซมีสมาชิกที่เกี่ยวข้องกับการพิจารณาหาค่า δ_{Stab} ได้แก่

   - ระบบในเวอร์ชันคู่ขนานสุดท้าย V_1
   - ระบบในเวอร์ชันคู่ขนานสุดท้าย V_2
   - ระบบในเวอร์ชันคู่ขนานสุดท้าย V_3

5. คำนวณค่าระยะทางระหว่างระบบของแต่ละเวอร์ชันคู่ขนานสุดท้าย ซึ่งหมายถึงค่า System Stability (StabSY) ของระบบดังต่อไปนี้

   - StabSY ของระบบระหว่างเวอร์ชันคู่ขนานสุดท้าย V_1 และเวอร์ชันคู่ขนานสุดท้าย V_2 คือค่าที่น้อยที่สุดระหว่าง $StabSY_{1-2}$ และ $StabSY_{2-1}$
   - StabSY ของระบบระหว่างเวอร์ชันคู่ขนานสุดท้าย V_1 และเวอร์ชันคู่ขนานสุดท้าย V_3 คือค่าที่น้อยที่สุดระหว่าง $StabSY_{1-3}$ และ $StabSY_{3-1}$
   - StabSY ของระบบระหว่างเวอร์ชันคู่ขนานสุดท้าย $V_{1.2}$ และเวอร์ชันคู่ขนานสุดท้าย V_3 คือค่าที่น้อยที่สุดระหว่าง $StabSY_{2-3}$ และ $StabSY_{3-2}$

จุฬาลงกรณ์มหาวิทยาลัย

6. เมื่อได้ค่าระยะทางของแต่ละคู่ของเวอร์ชันคู่ขนานสุดท้ายแล้ว ต่อไปก็นำค่าที่คำนวณได้จากข้อ 5 มาคำนวณหาค่า δ_{Stab}

$$\delta_{StabSY} = \min(StabSY_{i-j}) \text{ , } i,j > 0 \qquad (13)$$

บทที่ 4
การทดลองและประเมินแบบจำลอง

ในส่วนของการประเมินแบบจำลองการวัดความเสถียรจะประเมินโดยการทดลองความสอดคล้องกับตัววัดอื่นที่มีความเกี่ยวเนื่องกับความเสถียรโดยมีรายละเอียดดังนี้

4.1 วัตถุประสงค์ของการประเมิน

วัตถุประสงค์ของการประเมินผลแบบจำลองในงานวิจัยนี้คือ เพื่อเปรียบเทียบความสอดคล้องของแบบจำลองการวัดความเสถียรกับตัววัดอื่นที่มีความเกี่ยวเนื่องกับความเสถียรหรือความสามารถในการบำรุงรักษา

4.2 การประเมินแบบจำลองการวัดความเสถียร

การทดลองและประเมินผลแบบจำลองมีขั้นตอนต่าง ๆ ดังนี้

1) รวบรวมวิสเดิลตัวอย่างที่จะนำมาใช้ในการทดลอง
2) วัดความเสถียรของเซอร์วิซของวิสเดิลตัวอย่างเมื่อมีการเปลี่ยนแปลง
3) วัดค่าตัววัดอื่นและเปรียบเทียบความสอดคล้องโดยการหาสหสัมพันธ์ (Correlation) ระหว่างความเสถียรของเซอร์วิซกับตัววัดอื่น ได้แก่

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.1) ขนาดของส่วนต่อประสาน (Interface Granularity: IG) [12] คำนวณได้จากสูตร

$$IG = \frac{\sum_{i=1}^{nip} Wi}{nip} + \frac{\sum_{i=1}^{nop} Wi}{nop}$$

โดยที่ nip คือ จำนวนของอินพุตพารามิเตอร์
nop คือ จำนวนของเอาต์พุตพารามิเตอร์
Wi คือ ค่าน้ำหนักของพารามิเตอร์แต่ละตัวตามประเภทข้อมูลของพารามิเตอร์ โดยที่ void มีค่าน้ำหนักเป็น 0.0, primitive type มีค่าน้ำหนักเป็น 0.25, user-defined type มีค่าน้ำหนักเป็น 0.5, complex type มีค่าน้ำหนักเป็น 1

ขนาดของส่วนต่อประสานมีค่าระหว่าง (0,2] โดยถ้ามีค่าสูงหมายถึงส่วนต่อประสานมีขนาดใหญ่และสมบูรณ์

3.2) การเชื่อมติดกันของข้อมูลในส่วนต่อประสานของเซอร์วิซ (Service Interface Data Cohesion: SIDC) [13] คำนวณได้จากสูตร

$$SIDC(s) = \frac{|Common(Param(so \, \epsilon \, SO(si_s))|}{totalParamTypes}$$

โดยที่ Common(Param(so ϵ SO(si_s)) คือ ฟังก์ชันที่ส่งกลับเซตของโอเปอเรชันของเซอร์วิซที่มีประเภทข้อมูลของพารามิเตอร์เป็นประเภทเดียวกัน

totalParamTypes คือ จำนวนประเภทข้อมูลของพารามิเตอร์ทั้งหมด

การเชื่อมติดกันของข้อมูลในส่วนต่อประสานมีค่าระหว่าง (0,1] โดยถ้ามีค่าสูงหมายถึงมีการเชื่อมติดกันของข้อมูลในส่วนต่อประสานมาก เนื่องจากโอเปอเรชันมีการใช้ประเภทข้อมูลร่วมกันมาก

3.3) ความสามารถในการบำรุงรักษาโดยวัดจากค่าสัดส่วนจำนวนบรรทัดที่แก้ไขในวิสเดิล คำนวณได้จากสูตร

$$LCA = \frac{Number\ of\ changed\ lines}{totalLines}$$

โดยที่ Number of changed lines คือ จำนวนบรรทัดที่แก้ไขในวิสเดิล

totalLines คือ จำนวนบรรทัดทั้งหมดในวิสเดิล

สัดส่วนจำนวนบรรทัดที่แก้ไข มีค่าระหว่าง (0,1] ถ้ามีค่าสูงหมายถึงวิสเดิลมีการแก้ไขมากซึ่งส่งผลให้ความสามารถในการบำรุงรักษาต่ำ

ในส่วนของการรวบรวมวิสเดิลตัวอย่างเพื่อนำมาทดลอง ผู้วิจัยได้ขอความอนุเคราะห์จากบริษัทโลคัส คอมมิวนิเคชัน จำกัด เพื่อขอนำวิสเดิลมาเป็นตัวอย่างในการวิจัยทั้งหมด 6 เซอร์วิซ ซึ่งแสดงอยู่ในภาคผนวก ค หลังจากนั้น ทั้ง 6 เซอร์วิซ จะถูกแก้ไขตามความต้องการ คือประเภทของข้อมูลจะต้องสอดคล้องกับประเภทของข้อมูลที่ประกาศไว้ในระบบฐานข้อมูล ซึ่งหลังจากได้แก้ไขวิสเดิลทั้งหก ให้เป็นไปตามความต้องการแล้ว เพื่อให้สามารถเปรียบเทียบกับตัววัดที่เสนอมาข้างต้นซึ่งเป็นตัววัดที่เกี่ยวข้องกับทั้งวิสเดิลของเซอร์วิซหนึ่ง ๆ ได้อย่างเหมาะสม ผู้วิจัยเลือกที่จะทดลองกับการวัดความเสถียรของเซอร์วิซเดี่ยว ซึ่งได้ผลดังตารางที่ 4.1 และจะนำค่านี้ไปวิเคราะห์หาความสัมพันธ์กับตัววัดอื่นต่อไป

ตารางที่ 4.1 ความเสถียรของเซอร์วิซของวิสเดิลตัวอย่าง

วิสเดิล	ความเสถียรของเซอร์วิซ
WSDL-01	0.996
WSDL-02	0.971
WSDL-03	0.972
WSDL-04	0.974
WSDL-05	0.987
WSDL-06	0.99

4.2.1 การประเมินแบบจำลองการวัดความเสถียรของเซอร์วิซโดยการหาความสัมพันธ์กับขนาดของส่วนต่อประสาน

เนื่องจากมีสมมติฐานว่าเซอร์วิซที่มีขนาดใหญ่หรือมีฟังก์ชันครอบคลุมอยู่แล้วน่าจะไม่เปลี่ยนแปลงมากนักหรือมีความเสถียรเมื่อมีวิวัฒนาการ ผู้วิจัยจึงได้ทำการวัดขนาดของส่วนต่อประสานในวิสเดิลตัวอย่างซึ่งเป็นเวอร์ชันก่อนแก้ไขแล้วนำมาหาความสัมพันธ์กับความเสถียรของเซอร์วิซโดยมีสมมติฐานทางสถิติดังนี้

$H_0 : \rho = 0$ (ขนาดของส่วนต่อประสานไม่มีความสัมพันธ์กับความเสถียรของเซอร์วิซ)

$H_1 : \rho > 0$ (ขนาดของส่วนต่อประสานมีความสัมพันธ์เชิงบวกกับความเสถียรของเซอร์วิซ)

ค่าขนาดของส่วนต่อประสานที่ได้จากการคำนวณจากวิสเดิลตัวอย่างทั้งหก เป็นดังตารางที่ 4.2 ซึ่งสามารถนำผลไปสร้างเป็นแผนภาพกระจัดกระจาย (Scatter Plots) ได้ดังภาพที่ 4.1

ตารางที่ 4.2 ผลการวัดค่าขนาดของส่วนต่อประสานและความเสถียรของเซอร์วิซ

WSDL	Interface Granularity	Service Stability
WSDL-01	2	0.996
WSDL-02	1.0625	0.971
WSDL-03	0.5	0.972
WSDL-04	0.65	0.974

ตารางที่ 4.2 ผลการวัดค่าขนาดของส่วนต่อประสานและความเสถียรของเซอร์วิซ (ต่อ)

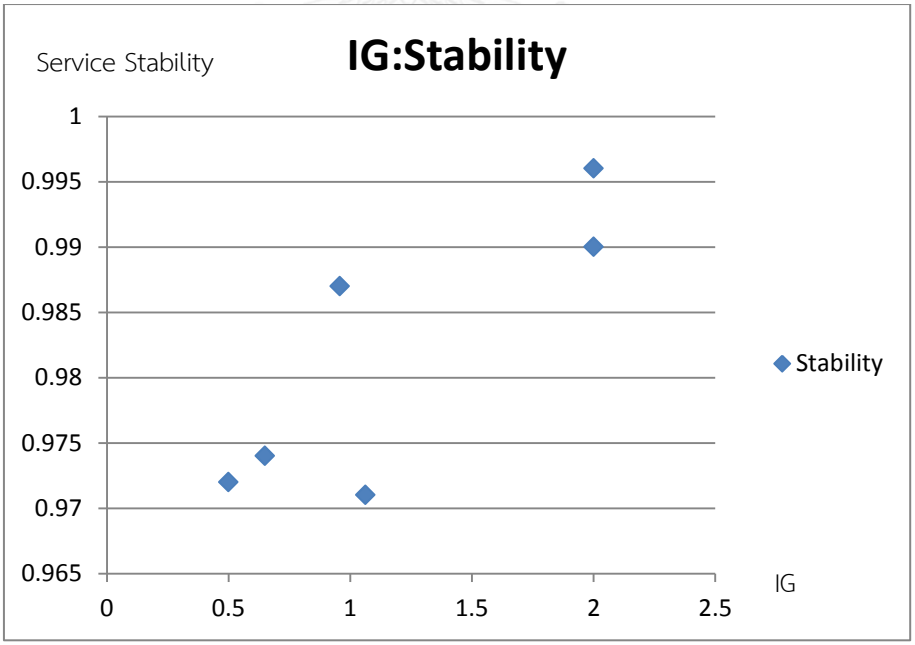
WSDL	Interface Granularity	Service Stability
WSDL-05	0.958	0.987
WSDL-06	2	0.99

กำหนดระดับนัยสำคัญทางสถิติ (α) = 0.05

คำนวณหาค่า r ซึ่งมีค่าเท่ากับ 0.837286 และ $t_{คำนวณ}$ มีค่าเท่ากับ 2.652371 นำค่า $t_{คำนวณ}$ เทียบกับค่า $t_{วิกฤต}$ ที่ได้จากตารางที่ α = 0.05, df = n-2 =4 โดยที่ $t_{0.05,4}$ = 2.132

พบว่า $t_{คำนวณ} > t_{วิกฤต}$ จึงปฏิเสธ H_0 และยอมรับ H_1

จึงสรุปได้ว่า ขนาดของส่วนต่อประสานมีความสัมพันธ์เชิงบวกกับความเสถียรของเซอร์วิซที่ระดับนัยสำคัญ 0.05 แสดงให้เห็นว่าวิสเดิลนั้นถูกออกแบบมาให้มีความสมบูรณ์ของฟังก์ชันสูงเมื่อเกิดวิวัฒนาการของเซอร์วิซขึ้นแล้ววิสเดิลจะมีการเปลี่ยนแปลงน้อยทำให้เซอร์วิซมีความเสถียรสูง



ภาพที่ 4.1 แผนภาพกระจัดกระจายของขนาดของส่วนต่อประสานและความเสถียรของเซอร์วิซ

4.2.2 การประเมินแบบจำลองการวัดความเสถียรของเซอร์วิซโดยการหาความสัมพันธ์กับการเชื่อมติดกันของข้อมูลในส่วนต่อประสานของเซอร์วิซ

เนื่องจากมีสมมติฐานว่าการออกแบบที่ดีควรมีการเชื่อมติดกัน (Cohesion) ของข้อมูลภายในส่วนต่อประสานสูง ถ้าภายในส่วนต่อประสานมีการออกแบบที่ดีอยู่แล้วก็จะส่งผลให้เกิดความเสถียรของเซอร์วิซเมื่อมีวิวัฒนาการ แต่อย่างไรก็ตามหากข้อมูลมีความเชื่อมติดกันสูง เมื่อจำเป็นต้องแก้ไขอาจมีผลกระทบต่อกันในวงกว้างเช่นกัน ผู้วิจัยจึงได้ทำการวัดค่าของการเชื่อมติดกันของส่วนต่อประสานในวิสเดิลตัวอย่างซึ่งเป็นเวอร์ชันก่อนแก้ไขแล้วนำมาหาความสัมพันธ์กับความเสถียรของเซอร์วิซโดยมีสมมติฐานทางสถิติดังนี้

H_0 : $\rho = 0$ (การเชื่อมติดกันของข้อมูลในส่วนต่อประสานของเซอร์วิซไม่มีความสัมพันธ์กับความเสถียรของเซอร์วิซ)

H_1 : $\rho \neq 0$ (การเชื่อมติดกันของข้อมูลในส่วนต่อประสานของเซอร์วิซมีความสัมพันธ์กับความเสถียรของเซอร์วิซ)

ค่าการเชื่อมติดกันของข้อมูลในส่วนต่อประสานของเซอร์วิซที่ได้จากการคำนวณจากวิสเดิลตัวอย่างทั้งหก เป็นดังตารางที่ 4.3 ซึ่งสามารถนำผลไปสร้างเป็นแผนภาพกระจัดกระจาย (Scatter Plots) ได้ดังภาพที่ 4.2

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 4.3 ผลการวัดค่าการเชื่อมติดกันของข้อมูลในส่วนต่อประสานของเซอร์วิซและความเสถียรของเซอร์วิซ

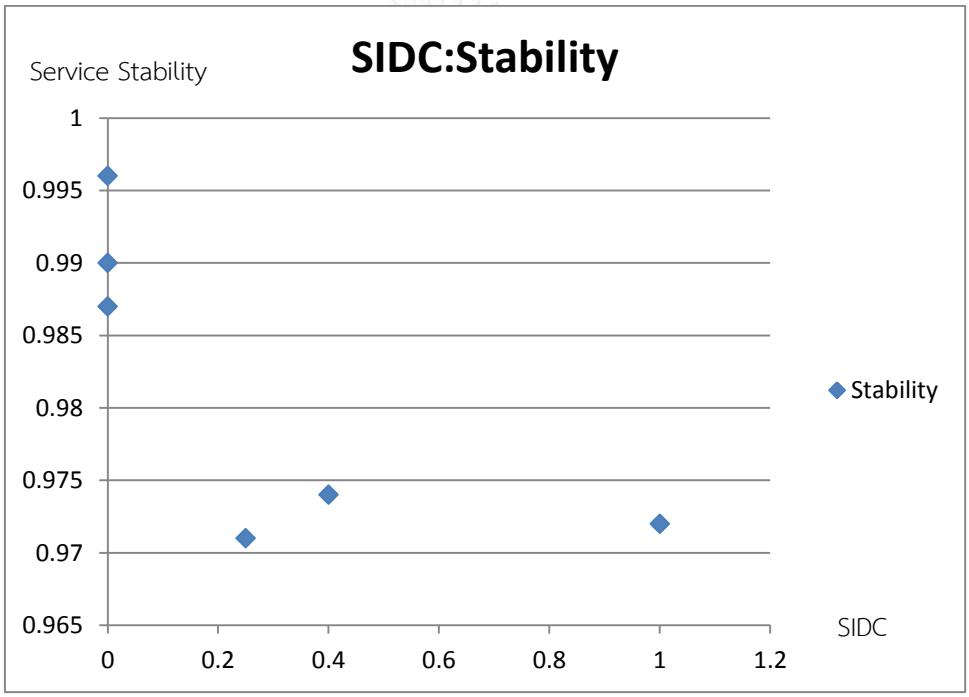
WSDL	Service Interface Data Cohesion	Service Stability
WSDL-01	0	0.996
WSDL-02	0.25	0.971
WSDL-03	1	0.972
WSDL-04	0.4	0.974
WSDL-05	0	0.987
WSDL-06	0	0.99

กำหนดระดับนัยสำคัญทางสถิติ (α) = 0.05

คำนวณหาค่า r ซึ่งมีค่าเท่ากับ -0.73611 และ $t_{คำนวณ}$ มีค่าเท่ากับ -1.88368 นำค่า $t_{คำนวณ}$ เทียบกับค่า $t_{วิกฤต}$ ที่ได้จากตารางที่ α = 0.05, df = n-2 = 4 โดยที่ $t_{0.025,4}$ = 2.776

พบว่า $t_{คำนวณ} < -t_{วิกฤต}$ และ $t_{คำนวณ} > t_{วิกฤต}$ จึงยอมรับ H_0 และปฏิเสธ H_1

จึงสรุปได้ว่า การเชื่อมติดกันของข้อมูลในส่วนต่อประสานของเซอร์วิซไม่มีความสัมพันธ์กับความเสถียรของเซอร์วิซ ที่ระดับนัยสำคัญ 0.05 การออกแบบส่วนต่อประสานให้มีการเชื่อมติดกันของข้อมูลจะไม่ส่งผลต่อความเสถียรของเซอร์วิซ



ภาพที่ 4.2 แผนภาพกระจัดกระจายของการเชื่อมติดกันของข้อมูลในส่วนต่อประสานของเซอร์วิซและความเสถียรของเซอร์วิซ

4.2.3 การประเมินแบบจำลองการวัดความเสถียรของเซอร์วิซโดยการหาความสัมพันธ์กับความสามารถในการบำรุงรักษา

ตามที่ ISO 9126 ได้นิยามให้ความเสถียรเป็นส่วนหนึ่งของความสามารถในการบำรุงรักษา ดังนั้นวิสเดิลที่มีความเสถียรจึงน่าจะมีความสัมพันธ์กับความสามารถในการบำรุงรักษาโดยที่ความสามารถในการบำรุงรักษานี้จะวัดจากแรงงาน (Effort) ที่ใช้ในการบำรุงรักษาซึ่งวัดจากจำนวน

บรรทัดที่แก้ไขในวิสเดิล ผู้วิจัยจึงได้ทำการวัดค่าสัดส่วนจำนวนบรรทัดที่แก้ไขในวิสเดิลตัวอย่างซึ่งเป็นเวอร์ชันก่อนแก้ไขแล้วนำมาหาความสัมพันธ์กับความเสถียรของเซอร์วิซโดยมีสมมติฐานทางสถิติดังนี้

$H_0 : \rho = 0$ (สัดส่วนจำนวนบรรทัดที่แก้ไขในวิสเดิลไม่มีความสัมพันธ์กับความเสถียรของเซอร์วิซ)

$H_1 : \rho < 0$ (สัดส่วนจำนวนบรรทัดที่แก้ไขในวิสเดิลมีความสัมพันธ์เชิงลบกับความเสถียรของเซอร์วิซ)

ค่าสัดส่วนจำนวนบรรทัดที่แก้ไขในวิสเดิลซึ่งได้จากการคำนวณจากวิสเดิลตัวอย่างทั้งหก เป็นดังตารางที่ 4.4 ซึ่งสามารถนำผลไปสร้างเป็นแผนภาพกระจัดกระจาย (Scatter Plots) ได้ดังภาพที่ 4.3

ตารางที่ 4.4 ผลการวัดค่าจำนวนบรรทัดที่แก้ไขเฉลี่ยและความเสถียรของเซอร์วิซ

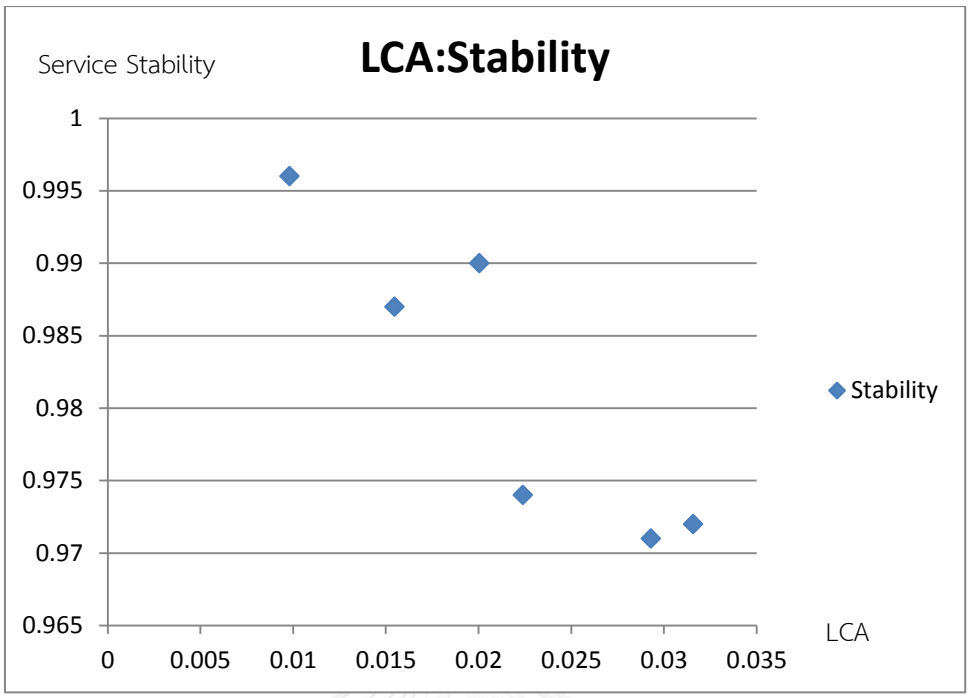
WSDL	Line Change Average	Service Stability
WSDL-01	0.009804	0.996
WSDL-02	0.029304	0.971
WSDL-03	0.031579	0.972
WSDL-04	0.022388	0.974
WSDL-05	0.015464	0.987
WSDL-06	0.02005	0.99

กำหนดระดับนัยสำคัญทางสถิติ (α) = 0.05

คำนวณหาค่า r ซึ่งมีค่าเท่ากับ -0.90968 และ $t_{คำนวณ}$ มีค่าเท่ากับ -3.793982016 นำค่า $t_{คำนวณ}$ เทียบกับค่า $t_{วิกฤต}$ ที่ได้จากตารางที่ α = 0.05, df = n-2 = 4 โดยที่ $t_{0.05,4}$ = 2.132

พบว่า $t_{คำนวณ} < -t_{วิกฤต}$ จึงปฏิเสธ H_0 และยอมรับ H_1

จึงสรุปได้ว่า สัดส่วนจำนวนบรรทัดที่แก้ไขในวิสเดิลมีความสัมพันธ์เชิงลบกับความเสถียรของเซอร์วิซที่ระดับนัยสำคัญ 0.05 แสดงให้เห็นว่าหากสัดส่วนจำนวนบรรทัดที่แก้ไขในวิสเดิลมีค่ามากหมายถึงมีความสามารถในการบำรุงรักษาต่ำ จะส่งผลให้ความเสถียรของเซอร์วิซต่ำเช่นกัน



ภาพที่ 4.3 แผนภาพกระจัดกระจายของจำนวนบรรทัดที่แก้ไขเฉลี่ยของวิสเดิลและความเสถียรของเซอร์วิซ

4.3 สรุปผลการประเมิน

จากการทดลองพบว่าความเสถียรของเซอร์วิซมีความสัมพันธ์เชิงบวกกับขนาดของส่วนต่อประสานที่ระดับนัยสำคัญทางสถิติ 0.05 แสดงให้เห็นว่าวิสเดิลนั้นถูกออกแบบมาให้มีความสมบูรณ์ของฟังก์ชันสูงเมื่อเกิดวิวัฒนาการของเซอร์วิซขึ้นแล้ววิสเดิลจะมีการเปลี่ยนแปลงน้อย ทำให้เซอร์วิซมีความเสถียรสูง ซึ่งเป็นไปตามสมมติฐานที่ตั้งไว้ และความเสถียรของเซอร์วิซมีความสัมพันธ์เชิงลบกับความสามารถในการบำรุงรักษาที่วัดด้วยสัดส่วนจำนวนบรรทัดที่แก้ไขในวิสเดิลที่ระดับนัยสำคัญทางสถิติ 0.05 แสดงให้เห็นว่าหากสัดส่วนจำนวนบรรทัดที่แก้ไขในวิสเดิลมีค่ามาก จะส่งผลให้ความเสถียรของเซอร์วิซต่ำ ซึ่งเป็นไปตามสมมติฐานเช่นกัน ดังนั้นจึงสรุปได้ว่าค่าความเสถียรที่ผู้วิจัยเสนอมีความสอดคล้องกับค่าตัววัดที่มีอยู่และมีความสมเหตุสมผลตามสมมติฐานความสัมพันธ์ที่ตั้งไว้ จึงสามารถนำแบบจำลองการวัดความเสถียรของเซอร์วิซไปใช้งานได้

ส่วนการทดสอบระหว่างความเสถียรของเซอร์วิซกับการเชื่อมติดกันของส่วนต่อประสานของเซอร์วิซนั้น ไม่พบความสัมพันธ์เชิงสถิติ ซึ่งหมายถึงการออกแบบส่วนต่อประสานให้มีการเชื่อมติดกันของข้อมูลสูงหรือต่ำจะไม่ส่งผลต่อความเสถียรของเซอร์วิซ

บทที่ 5
การพัฒนาเครื่องมือสนับสนุน

ในบทนี้จะกล่าวถึงวิธีการพัฒนาเครื่องมือสนับสนุนการวัดค่าความเสถียรในมุมมองต่าง ๆ โดยเริ่มจากการกำหนดความต้องการ ออกแบบและพัฒนาโปรแกรม การออกแบบนั้นจะใช้ภาษายูเอ็มแอลในการอธิบายแผนภาพต่าง ๆ ประกอบด้วย แผนภาพยูสเคสแสดงภาพรวมและการปฏิสัมพันธ์ของผู้ใช้โปรแกรมกับตัวโปรแกรม แผนภาพคลาสแสดงโครงสร้างและความสัมพันธ์ของคลาสต่าง ๆ ภายในโปรแกรม ตัวอย่างการใช้โปรแกรมอีคลิปส์เพื่อพัฒนาเครื่องมือสนับสนุนการวัดค่าความเสถียรแสดงไว้ในภาคผนวก ก และการใช้งานเครื่องมือวัดค่าความเสถียรอย่างละเอียดอธิบายไว้ในภาคผนวก ข ซึ่งในส่วนของการออกแบบและพัฒนาเครื่องมือสนับสนุนการคำนวณค่าความเสถียรมีรายละเอียดดังต่อไปนี้

5.1 ความต้องการที่เป็นฟังก์ชัน

ความต้องการที่เป็นฟังก์ชันหลักของเครื่องมือสนับสนุนการวัดค่าความเสถียร สามารถแบ่งได้เป็น 3 ส่วนดังนี้

5.1.1 ส่วนการนำเข้าข้อมูล

ผู้ใช้สามารถเลือกว่าจะวัดค่าความเสถียรในมุมมองใด จากนั้นระบบจะให้นำเข้าไฟล์วิสเดิลเวอร์ชันต่าง ๆ ซึ่งจะสอดคล้องกับมุมมองความเสถียรที่ผู้ใช้เลือก แต่ละไฟล์ที่นำเข้าคือวิสเดิลที่สมบูรณ์ซึ่งเป็นไฟล์ประเภทเอกซ์เอ็มแอล (XML) โดยจะต้องระบุข้อมูลให้กับวิสเดิลนั้น ๆ ว่าเป็นวิสเดิลเวอร์ชันใดของเซอร์วิซใดในระบบ เพื่อที่ส่วนประมวลผลจะได้นำข้อมูลเหล่านี้ไปประมวลผลได้อย่างถูกต้อง

5.1.2 ส่วนการประมวลผล

โปรแกรมจะประมวลผลจากข้อมูลนำเข้าและข้อมูลที่ระบุในขั้นตอนนำเข้าข้อมูล โดยจะคำนวณตามวิธีการวัดในหัวข้อที่ 3.1 และ 3.2

5.1.3 ส่วนการแสดงผล

โปรแกรมจะแสดงค่าที่ได้จากการประมวลผลเพื่อแสดงวิวัฒนาการของระบบเซอร์วิซและความเสถียรในแบบต่าง ๆ ตามที่ผู้ใช้ได้เลือกในขั้นตอนการนำเข้าข้อมูล ในรูปแบบแผนภูมิแท่ง เช่น แสดงค่าความเสถียรของเซอร์วิซเดี่ยวแบบต่อเนื่องใน 3 เวอร์ชัน เป็นต้น

5.2 ความสามารถของเครื่องมือ

เพื่อให้เครื่องมือสามารถทำงานได้ตามความต้องการที่ระบุไว้ในหัวข้อ 5.1 ผู้วิจัยจึงได้กำหนดให้เครื่องมือสนับสนุนการวัดความเสถียรมีความสามารถได้ดังต่อไปนี้

1) สามารถเลือกมุมมองการวัดความเสถียรได้ดังนี้
   - วัดความเสถียรระหว่างเวอร์ชันของโอเปอร์เรชัน
   - วัดความเสถียรระหว่างเวอร์ชันของเซอร์วิซเดี่ยว
   - วัดความเสถียรระหว่างเวอร์ชันของระบบเซอร์วิซ
   - วัดความเสถียรโดยรวมของโอเปอร์เรชัน
   - วัดความเสถียรโดยรวมของเซอร์วิซเดี่ยว
   - วัดความเสถียรโดยรวมของระบบเซอร์วิซ

2) สามารถนำเข้าไฟล์วิสเดิลได้ตามมุมมองที่เลือกโดยระบุข้อมูล เลขที่เวอร์ชัน, เลขที่เซอร์วิซ และเลขที่เวอร์ชันคู่ขนานสุดท้ายได้

3) สามารถวัดความเสถียรได้ตามมุมมองดังนี้
   - ความเสถียรระหว่างเวอร์ชันของโอเปอร์เรชัน
   - ความเสถียรระหว่างเวอร์ชันของเซอร์วิซเดี่ยว
   - ความเสถียรระหว่างเวอร์ชันของระบบเซอร์วิซ
   - ความเสถียรโดยรวมของโอเปอร์เรชัน
   - ความเสถียรโดยรวมของเซอร์วิซเดี่ยว
   - ความเสถียรโดยรวมของระบบเซอร์วิซ

4) สามารถแสดงผลการวัดความเสถียรโดยแผนภูมิแท่งซึ่งแบ่งกลุ่มได้ 2 กลุ่มคือ ความเสถียรระหว่างเวอร์ชันและความเสถียรโดยรวม

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

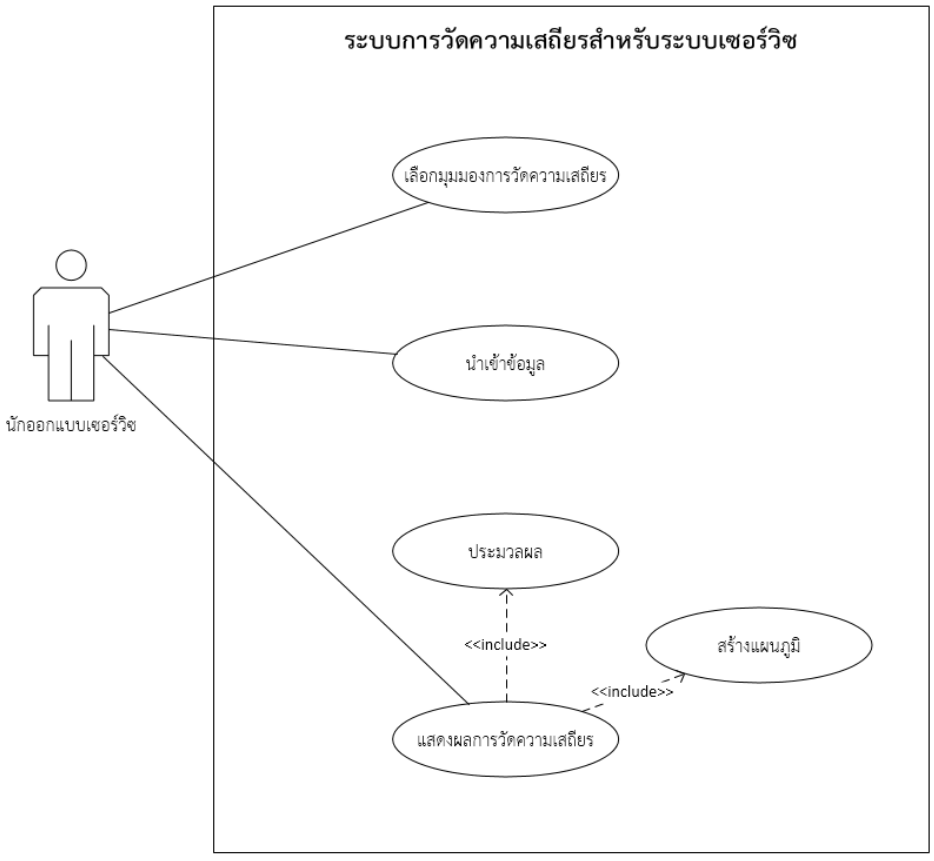
5.3 การออกแบบเครื่องมือ

เครื่องมือที่สร้างใช้เพื่อสนับสนุนการวัดความเสถียรในมุมมองต่าง ๆ และแสดงผลตามมุมมองที่เลือกไว้ ซึ่งสามารถอธิบายรายละเอียดต่าง ๆ โดยแผนภาพยูเอ็มแอลดังต่อไปนี้

5.3.1 แผนภาพยูสเคส

จากการวิเคราะห์การใช้งานระบบของนักออกแบบเซอร์วิซ สามารถอธิบายด้วยแผนภาพยูสเคสดังภาพที่ 5.1 ซึ่งเป็นการแสดงให้เห็นว่านักออกแบบเซอร์วิซจะต้องมีปฏิสัมพันธ์กับโปรแกรมอย่างไร ซึ่งมีรายละเอียดดังนี้

1) นักออกแบบเซอร์วิซสามารถกำหนดมุมมองในการวัดค่าความเสถียรตามที่ต้องการได้
2) นักออกแบบเซอร์วิซเลือกไฟล์วิสเดิลเพื่อทำการวัดค่าความเสถียรตามมุมมองที่ต้องการได้
3) นักออกแบบเซอร์วิซสามารถระบุข้อมูลที่จำเป็นต่อการวัดค่าความเสถียรให้กับไฟล์วิสเดิลที่จะนำมาวัดค่าได้
4) โปรแกรมสามารถแสดงผลการวัดได้ตามมุมมองที่นักออกแบบเซอร์วิซเลือกในขั้นตอนนำเข้าข้อมูล



ภาพที่ 5.1 แผนภาพยูสเคส

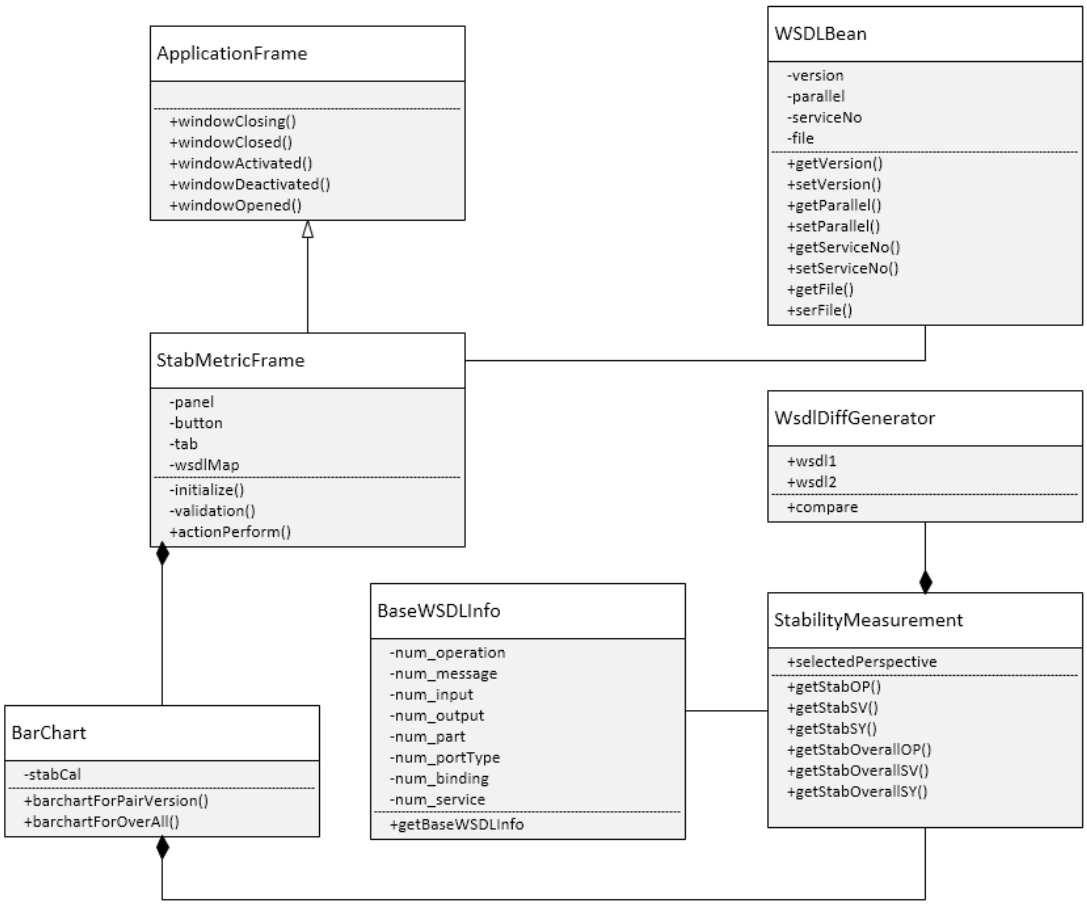
สามารถอธิบายรายละเอียดของยูสเคสได้ดังตารางที่ 5.1

ตารางที่ 5.1 คำอธิบายฟังก์ชันการทำงาน

ลำดับ	ยูสเคส	คำอธิบาย
1	เลือกมุมมองการวัด	ยูสเคสนี้เป็นขั้นตอนที่นักออกแบบเซอร์วิซเลือกประเภทการวัดความเสถียรซึ่งสามารถเลือกได้ 6 แบบ คือ -ความเสถียรของโอเปอเรชัน -ความเสถียรของเซอร์วิซเดี่ยว -ความเสถียรของระบบเซอร์วิซ -ความเสถียรโดยรวมของโอเปอเรชัน -ความเสถียรโดยรวมของเซอร์วิซเดี่ยว -ความเสถียรโดยรวมของระบบเซอร์วิซ
2	นำเข้าข้อมูล	นักออกแบบเซอร์วิซจะต้องทำการนำเข้าไฟล์วิสเดิลและระบุข้อมูลให้กับไฟล์วิซเดิลนั้น ๆ เพื่อให้ระบบวัดค่าความเสถียรได้อย่างถูกต้อง
3	ประมวลผล	ระบบวัดค่าความเสถียรตามมุมมองที่นักออกแบบเซอร์วิซเลือก
4	แสดงผล	ระบบแสดงผลลัพธ์จากการวัดตามมุมมองที่นักออกแบบเซอร์วิซเลือกในขั้นตอนแรก
5	สร้างแผนภูมิ	ระบบสร้างแผนภูมิแสดงผลความเสถียร

5.3.2 แผนภาพคลาส

ในการพัฒนาเครื่องมือผู้วิจัยได้ออกแบบคลาสให้มีลักษณะสอดคล้องกับยูสเคสซึ่งมีด้วยกัน 3 ส่วนคือ ส่วนเก็บข้อมูล ส่วนการวัดและส่วนแสดงผล การออกแบบคลาสและความสัมพันธ์ระหว่างคลาสแสดงดังแผนภาพคลาสเชิงตรรกะ (นำเสนอเฉพาะส่วนที่เป็นตรรกะของโปรแกรม ไม่รวมคลาสที่ใช้สร้างส่วนต่อประสาน) ของโปรแกรมดังภาพที่ 5.2

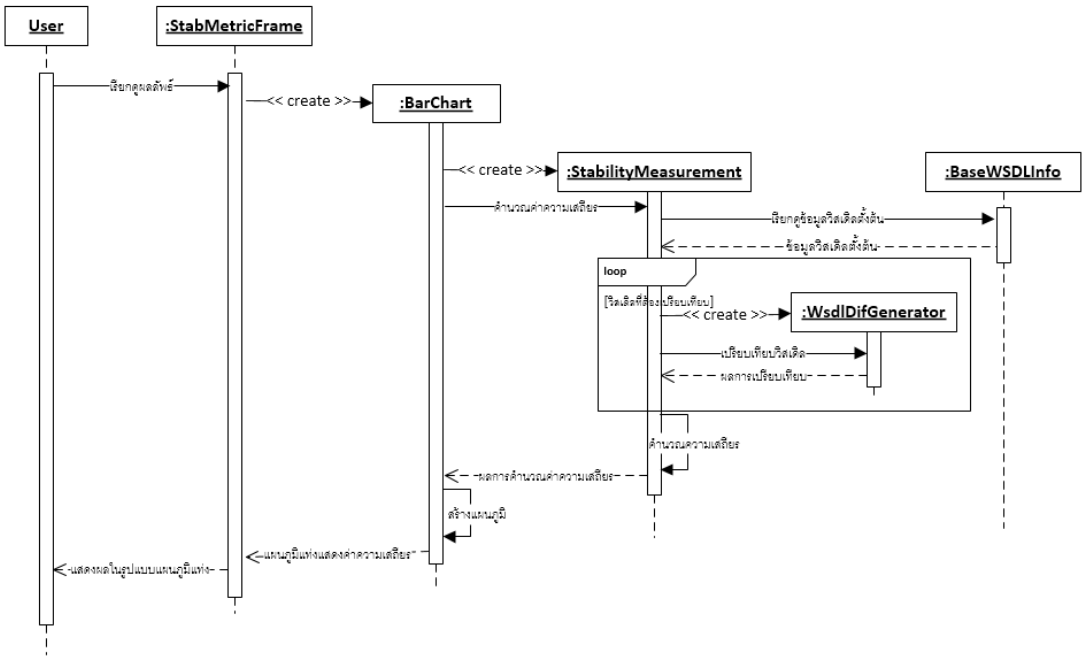


ภาพที่ 5.2 แผนภาพคลาสเชิงตรรกะ

ส่วนข้อมูลที่นำเข้าจะถูกเก็บในรูปแบบของวัตถุซึ่งจะมีรายละเอียดของข้อมูลเช่น เวอร์ชันของวิสเดิล เลขที่เซอร์วิซ เป็นต้น ข้อมูลเหล่านี้จะมีแตกต่างกันไปตามมุมมองที่ผู้ใช้เลือกในขั้นตอนแรก เมื่อมีข้อมูลพร้อมในการวัดความเสถียรแล้ว ระบบจะทำการคำนวณแล้วส่งผลลัพธ์ไปยังส่วนแสดงผลเพื่อแสดงผลลัพธ์จากการวัดให้แก่ผู้ใช้ซึ่งจะมีรูปแบบเป็นไปตามที่ผู้ใช้เลือก

5.3.3 แผนภาพลำดับ

จากแผนภาพคลาสในภาพที่ 5.2 สามารถแสดงตัวอย่างลำดับการทำงานในส่วนของการวัดความเสถียรของเซอร์วิซได้ดังภาพที่ 5.3 โดยอธิบายลำดับการทำงานได้ดังนี้

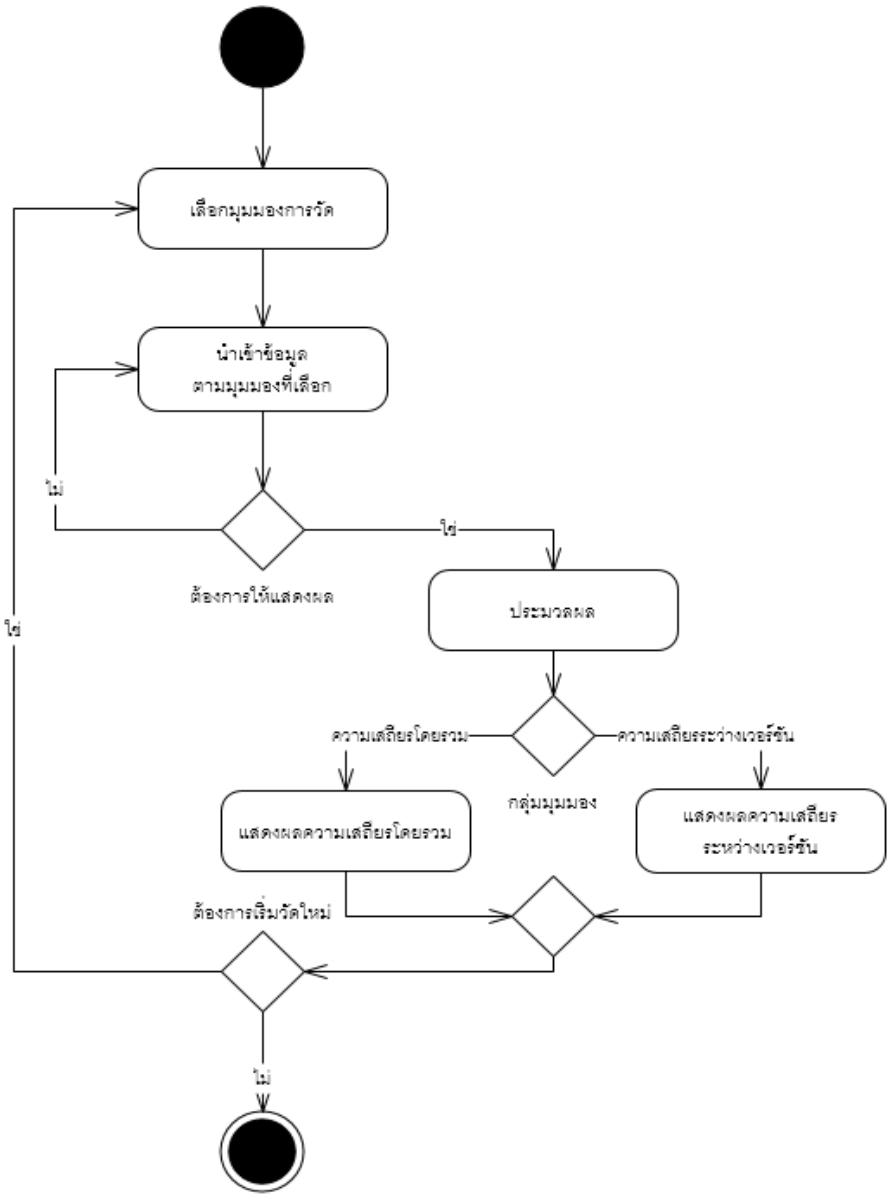


ภาพที่ 5.3 แผนภาพลำดับของการวัดความเสถียรของเซอร์วิซ

หลังจากที่ผู้ใช้ได้นำเข้าข้อมูลตามที่ได้เลือกมุมมองไว้แล้ว ผู้ใช้ต้องการดูผลการวัดความเสถียร จึงเรียกดูค่าความเสถียรจากโปรแกรม ในลำดับนี้คลาส StabMetricFrame จะทำการสร้างวัตถุของคลาส BarChart ขึ้นมา ซึ่งในการสร้างวัตถุจะต้องส่งรายการวิสเดิลเวอร์ชันต่าง ๆ ที่ได้นำเข้าไปด้วย จากนั้นวัตถุ BarChart จะสร้างวัตถุของคลาส StabilityMeasurement ขึ้นมาเพื่อทำหน้าที่คำนวณความเสถียร ลำดับต่อไปวัตถุ BarChart จะเรียกใช้วัตถุ StabilityMeasurement ให้ทำการคำนวณความเสถียรโดยจะเรียกดูข้อมูลจำนวนเอลิเมนต์ต่าง ๆ ในวิสเดิลเวอร์ชันตั้งต้นที่วัตถุ BaseWSDLInfo แล้ววัตถุ StabilityMeasurement จะส่งรายการวิสเดิลเวอร์ชันต่าง ๆ ไปเปรียบเทียบที่วัตถุของคลาส WsdlDiffGenerator โดยเปรียบเทียบทีละคู่ของเวอร์ชันจนกว่าจะหมดแล้วรวบรวมค่าความเสถียรแต่ละคู่ไปคำนวณความเสถียรของเซอร์วิซต่อไป จากนั้นวัตถุ StabilityMeasurement จะส่งผลการคำนวณกลับมาให้กับวัตถุ BarChart ซึ่งวัตถุ BarChart จะนำผลที่ได้นั้นมาสร้างเป็นแผนภูมิ และส่งกลับไปแสดงผลให้แก่ผู้ใช้งานที่หน้าจอ

5.3.4 แผนภาพกิจกรรม

ภาพรวมกิจกรรมที่เกิดขึ้นในการใช้งานเครื่องมือสามารถเขียนเป็นแผนภาพกิจกรรมดังภาพที่ 5.4 ซึ่งแผนภาพจะอธิบายกิจกรรมของผู้ใช้เมื่อใช้งานเครื่องมือตั้งแต่เริ่มต้นจนสิ้นสุดการใช้งาน



ภาพที่ 5.4 แผนภาพกิจกรรมภาพรวมการทำงานของเครื่องมือ

แผนภาพกิจกรรมเริ่มต้นจากผู้ใช้งานเลือกมุมมองการวัดความเสถียรตามที่ต้องการ จากนั้นผู้ใช้ต้องทำการนำเข้าข้อมูลวิสเดิลและระบุข้อมูลให้กับวิสเดิล เมื่อนำเข้าข้อมูลครบตามความต้องการแล้วผู้ใช้จะทำการเรียกดูผลการวัดความเสถียร ซึ่งระบบจะทำการคำนวณและแสดงผลตามกลุ่มมุมมองได้แก่ ความเสถียรระหว่างเวอร์ชันและความเสถียรโดยรวมซึ่งจะเป็นไปตามมุมมองที่ผู้ใช้เลือก

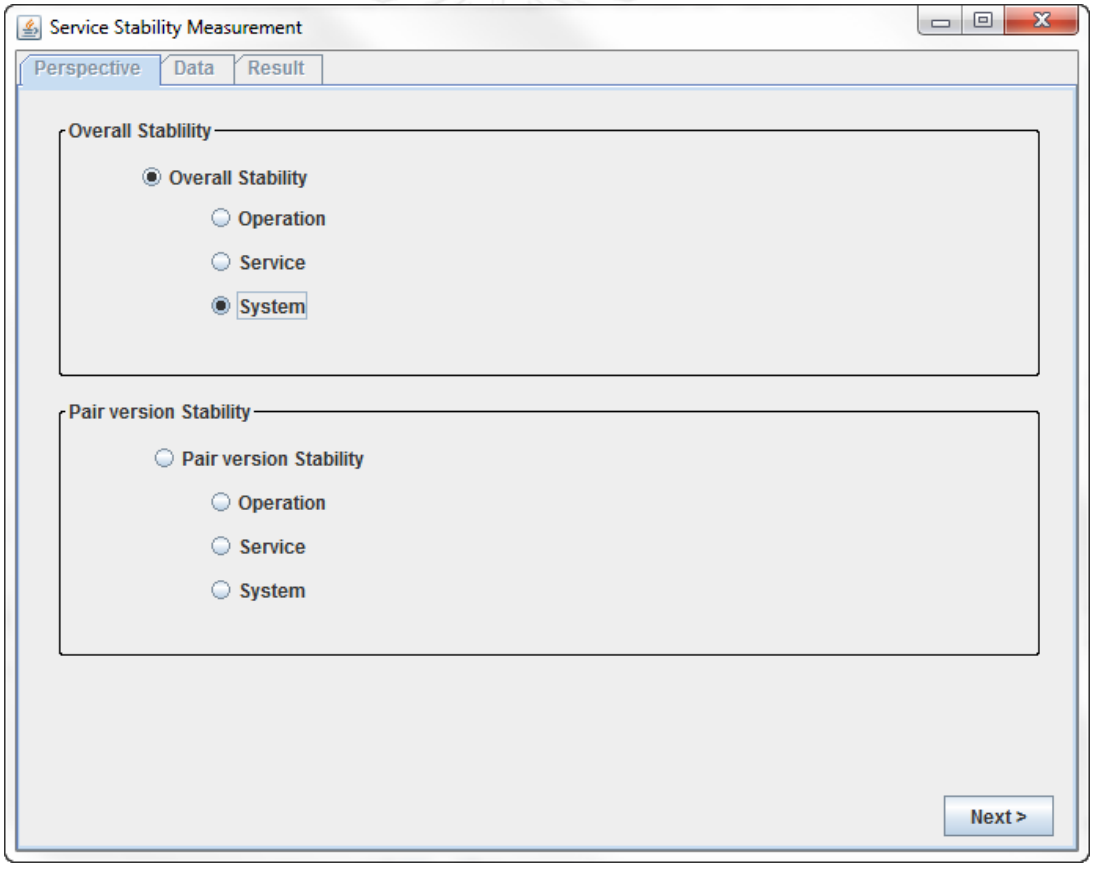
ตั้งแต่ต้น สุดท้ายผู้ใช้สามารถเลือกที่จะสิ้นสุดการใช้งานโดยการปิดโปรแกรมหรือเริ่มต้นการวัดความเสถียรใหม่อีกครั้ง

5.3.5 การออกแบบส่วนต่อประสาน

ผู้วิจัยได้ออกแบบส่วนต่อประสานให้สอดคล้องกับแนวคิดการออกแบบ จึงแบ่งส่วนต่อประสานเป็น 3 หน้าจอ ซึ่งแต่ละหน้าจอมีรายละเอียดดังนี้

1) หน้าจอเลือกมุมมอง

หน้าจอนี้แสดงไว้ดังภาพที่ 5.5 มีเพื่อให้ผู้ใช้งานได้เลือกมุมมองของการวัดค่าความเสถียรผู้ใช้สามารถเลือกได้มุมมองใดมุมมองหนึ่งเท่านั้น รายละเอียดของส่วนต่อประสานแสดงดังตารางที่ 5.2



ภาพที่ 5.5 หน้าจอเลือกมุมมองของการวัดค่าความเสถียร

ตารางที่ 5.2 รายละเอียดหน้าจอเลือกมุมมองการวัดค่าความเสถียร

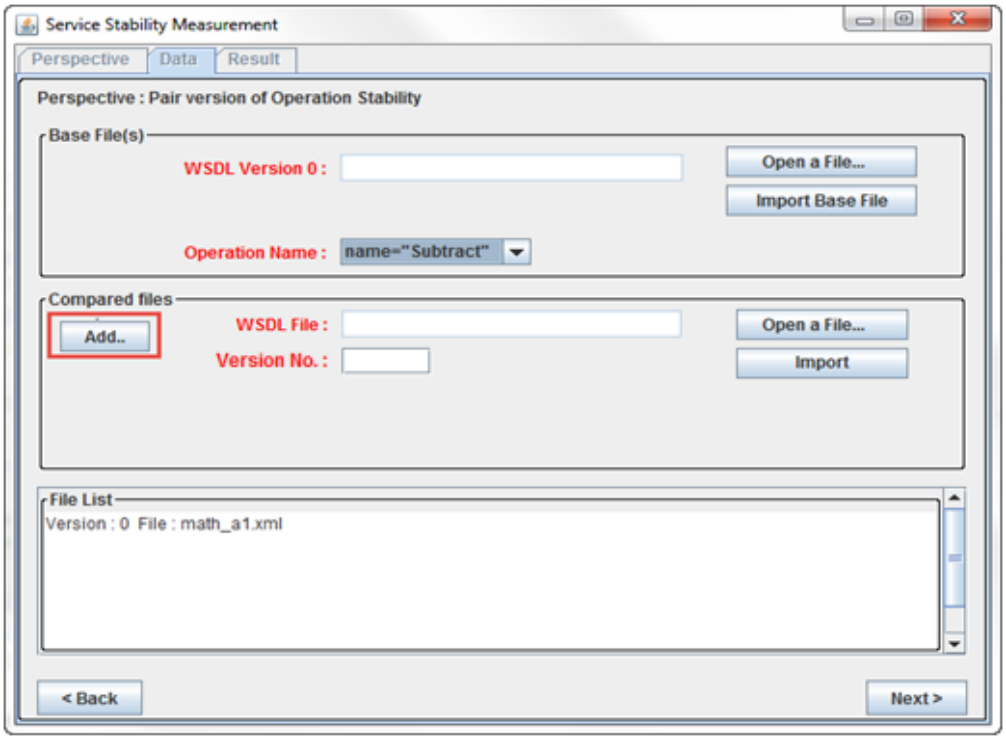
ลำดับ	ชื่อคุณลักษณะของส่วนต่อประสาน	ประเภท	คำอธิบาย	หมายเหตุ
1	Perspective	Radio	มุมมองการวัดค่าความเสถียร	สามารถเลือกได้มุมมองเดียวเท่านั้น
2	Next	Button	ปุ่มเพื่อไปยังหน้าถัดไป	

2) หน้าจอนำเข้าข้อมูล

หน้าจอนำเข้าข้อมูลนี้จุดประสงค์มีไว้เพื่อนำเข้าไฟล์วิสเดิลที่ต้องการทำการวัดค่าความเสถียร โดยหน้าจอจะแสดงส่วนต่อประสานที่สอดคล้องตามมุมมองที่เลือกมาจากหน้าจอเลือกมุมมอง โดยที่หน้าจอนำเข้าข้อมูลมีทั้งหมด 6 รูปแบบซึ่งอธิบายได้โดยภาพและตารางดังนี้

1. หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของโอเปอเรชัน

หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของโอเปอเรชันแสดงดังภาพที่ 5.6 และรายละเอียดส่วนต่าง ๆ ของหน้าจอแสดงดังตารางที่ 5.3



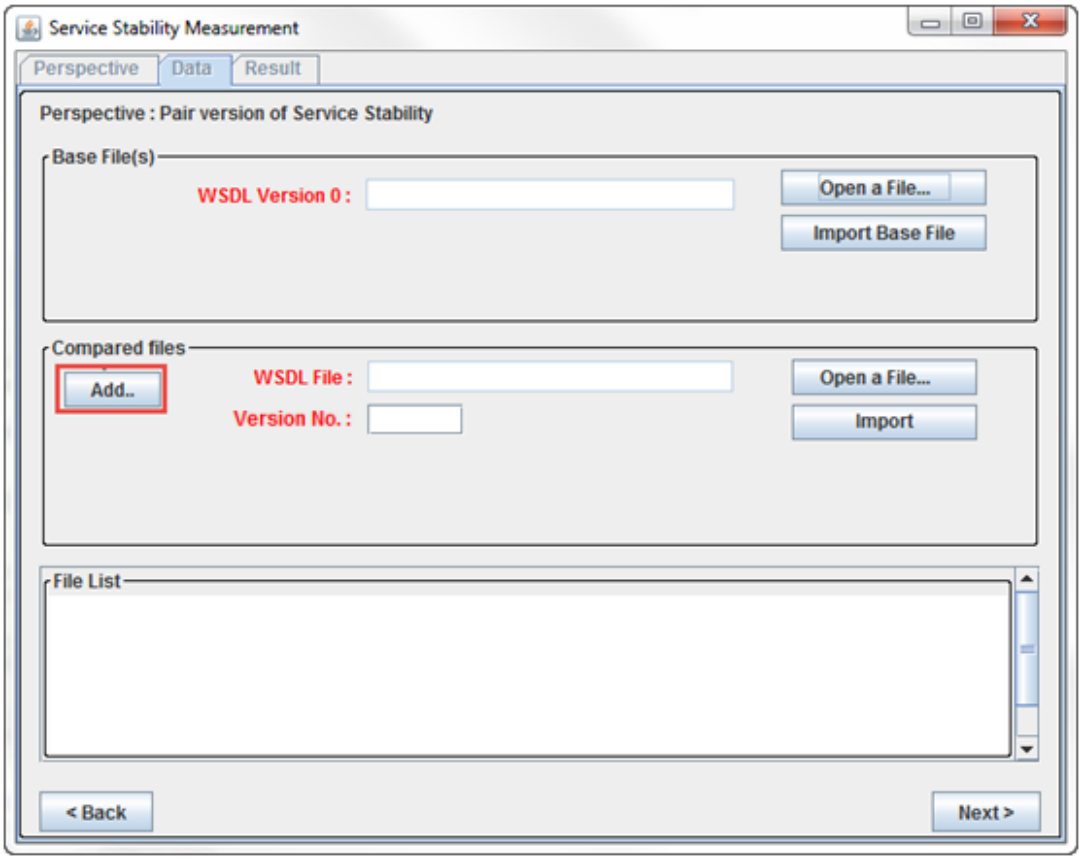
ภาพที่ 5.6 หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของโอเปอเรชัน

ตารางที่ 5.3 รายละเอียดหน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของโอเปอเรชัน

ลำดับ	ชื่อคุณลักษณะของส่วนต่อประสาน	ประเภท	คำอธิบาย	หมายเหตุ
1	WSDL Version 0	Text box	ไฟล์วิสเดิลตั้งต้นของการวัด (เวอร์ชัน 0)	
2	Open a File	Button	เลือกไฟล์ตั้งต้น	
3	Import Base File	Button	นำเข้าไฟล์วิสเดิลตั้งต้น	
4	Operation Name	Combo	โอเปอเรชันที่ต้องการวัดค่าความเสถียร	กรอกได้เฉพาะมุมมองความเสถียรของโอเปอเรชัน
5	Add	Button	ปุ่มเพิ่มไฟล์วิสเดิลที่จะนำมาเปรียบเทียบ	เมื่อกด ปุ่มจะหายไปและจะแสดงเมื่อนำเข้าไฟล์ที่นำมาเปรียบเทียบสำเร็จ
6	Open a File	Button	ปุ่มเลือกไฟล์วิสเดิลที่จะนำมาเปรียบเทียบ	
7	Version No.	Text box	เวอร์ชันของไฟล์วิสเดิลที่จะนำมาเปรียบเทียบ	
8	Import	Button	ปุ่มนำเข้าไฟล์วิสเดิล	
9	File List	Text Area	แสดงชื่อไฟล์วิสเดิลที่นำเข้า	
10	Next	Button	ปุ่มดำเนินการต่อ	เมื่อกดปุ่มนี้ โปรแกรมจะทำการคำนวณค่าความเสถียรและแสดงผลในหน้าต่อไป
11	Back	Button	ปุ่มกลับไปหน้าเลือกมุมมอง	

2. หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของเซอร์วิซ

หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของเซอร์วิซแสดงดังภาพที่ 5.7 และรายละเอียดส่วนต่าง ๆ ของหน้าจอแสดงดังตารางที่ 5.4



ภาพที่ 5.7 หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของเซอร์วิซ

ตารางที่ 5.4 รายละเอียดหน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของเซอร์วิซ

ลำดับ	ชื่อคุณลักษณะของส่วนต่อประสาน	ประเภท	คำอธิบาย	หมายเหตุ
1	WSDL Version 0	Text box	ไฟล์วิสเดิลตั้งต้นของการวัด (เวอร์ชัน 0)	
2	Open a File	Button	เลือกไฟล์ตั้งต้น	
3	Import Base File	Button	นำเข้าไฟล์วิสเดิลตั้งต้น	

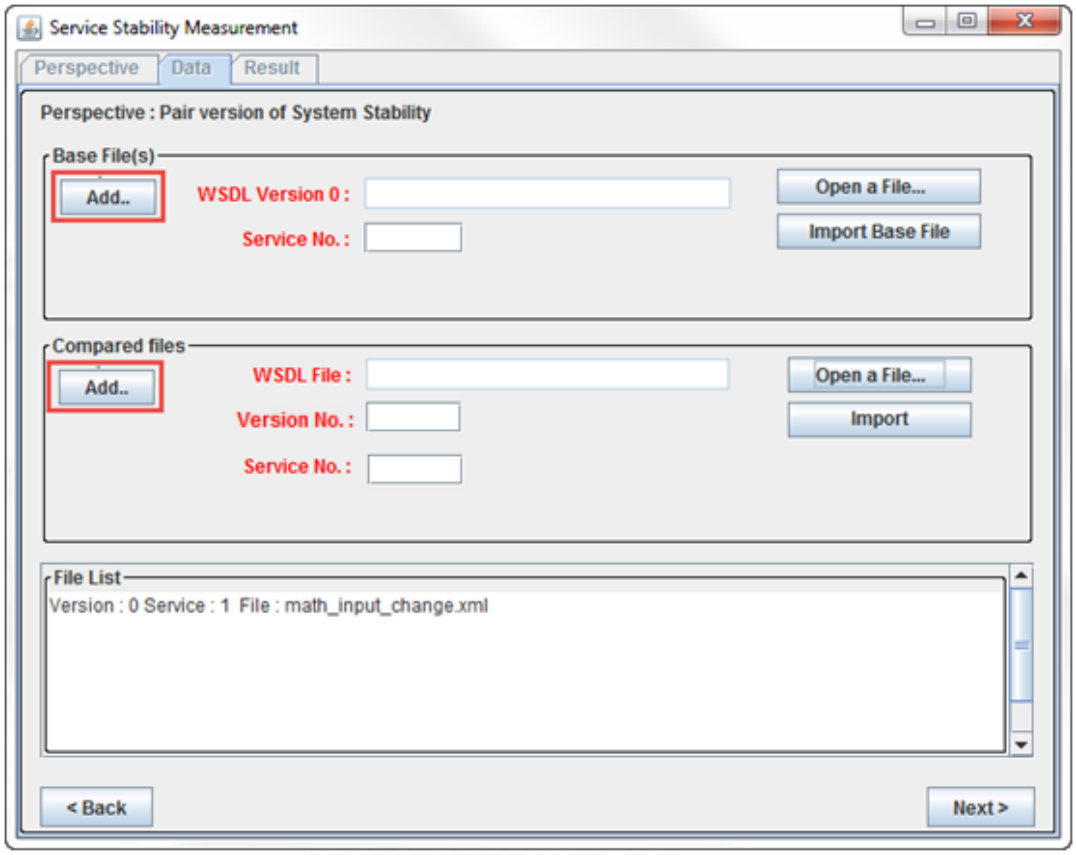
ตารางที่ 5.4 รายละเอียดหน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของเซอร์วิซ (ต่อ)

ลำดับ	ชื่อคุณลักษณะของส่วนต่อประสาน	ประเภท	คำอธิบาย	หมายเหตุ
4	Add	Button	ปุ่มเพิ่มไฟล์วิสเดิลที่จะนำมาเปรียบเทียบ	เมื่อกด ปุ่มจะหายไปและจะแสดงเมื่อนำเข้าไฟล์ที่นำมาเปรียบเทียบสำเร็จ
5	Open a File	Button	ปุ่มเลือกไฟล์วิสเดิลที่จะนำมาเปรียบเทียบ	
6	Version No.	Text box	เวอร์ชันของไฟล์วิสเดิลที่จะนำมาเปรียบเทียบ	
7	Import	Button	ปุ่มนำเข้าไฟล์วิสเดิล	
8	File List	Text Area	แสดงชื่อไฟล์วิสเดิลที่นำเข้า	
9	Next	Button	ปุ่มดำเนินการต่อ	เมื่อกดปุ่มนี้ โปรแกรมจะทำการคำนวณค่าความเสถียรและแสดงผลในหน้าต่อไป
10	Back	Button	ปุ่มกลับไปหน้าเลือกมุมมอง	

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3. หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของระบบเซอร์วิซ

หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของระบบเซอร์วิซ แสดงดังภาพที่ 5.8 และรายละเอียดส่วนต่าง ๆ ของหน้าจอแสดงดังตารางที่ 5.5



ภาพที่ 5.8 หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของระบบเซอร์วิซ

ตารางที่ 5.5 รายละเอียดหน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของระบบเซอร์วิซ

ลำดับ	ชื่อคุณลักษณะของส่วนต่อประสาน	ประเภท	คำอธิบาย	หมายเหตุ
1	Add	Button	ปุ่มเพิ่มไฟล์วิสเดิลตั้งต้น	เมื่อกด ปุ่มจะหายไป และจะแสดงเมื่อนำเข้าไฟล์ที่นำมาเปรียบเทียบสำเร็จ

ตารางที่ 5.5 รายละเอียดหน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของระบบเซอร์วิซ (ต่อ)

ลำดับ	ชื่อคุณลักษณะของส่วนต่อประสาน	ประเภท	คำอธิบาย	หมายเหตุ
2	WSDL Version 0	Text box	ไฟล์วิสเดิลตั้งต้นของการวัด (เวอร์ชัน 0)	
3	Service No.	Text box	เลขที่เซอร์วิซ ของเวอร์ชันตั้งต้น	
4	Open a File	Button	เลือกไฟล์ตั้งต้น	
5	Import Base File	Button	นำเข้าไฟล์วิสเดิลตั้งต้น	
6	Add	Button	ปุ่มเพิ่มไฟล์วิสเดิลที่จะนำมาเปรียบเทียบ	เมื่อกด ปุ่มจะหายไปและจะแสดงเมื่อนำเข้าไฟล์ที่นำมาเปรียบเทียบสำเร็จ
7	Open a File	Button	ปุ่มเลือกไฟล์วิสเดิลที่จะนำมาเปรียบเทียบ	
8	Version No.	Text box	เวอร์ชันของไฟล์วิสเดิลที่จะนำมาเปรียบเทียบ	
9	Service No.	Text box	เลขที่เซอร์วิซ ของแต่ละเวอร์ชัน	
10	Import	Button	ปุ่มนำเข้าไฟล์วิสเดิล	
11	File List	Text Area	แสดงชื่อไฟล์วิสเดิลที่นำเข้า	
12	Next	Button	ปุ่มดำเนินการต่อ	เมื่อกดปุ่มนี้ โปรแกรมจะทำการคำนวณค่าความเสถียรและแสดงผลในหน้าต่อไป

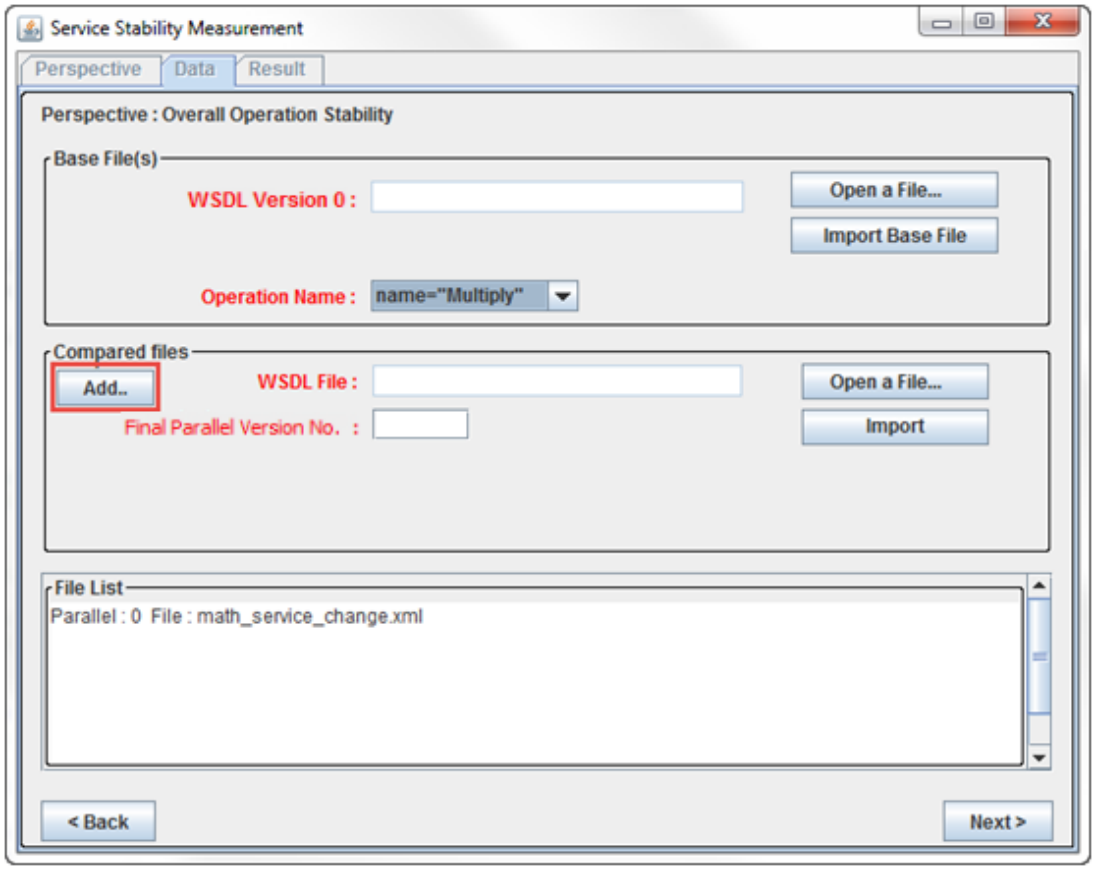
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 5.5 รายละเอียดหน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรระหว่างเวอร์ชันของระบบเซอร์วิซ (ต่อ)

ลำดับ	ชื่อคุณลักษณะของส่วนต่อประสาน	ประเภท	คำอธิบาย	หมายเหตุ
13	Back	Button	ปุ่มกลับไปหน้าเลือกมุมมอง	

4. หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของโอเปอเรชัน

หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของโอเปอเรชันแสดงดังภาพที่ 5.9 และรายละเอียดส่วนต่าง ๆ ของหน้าจอแสดงดังตารางที่ 5.6



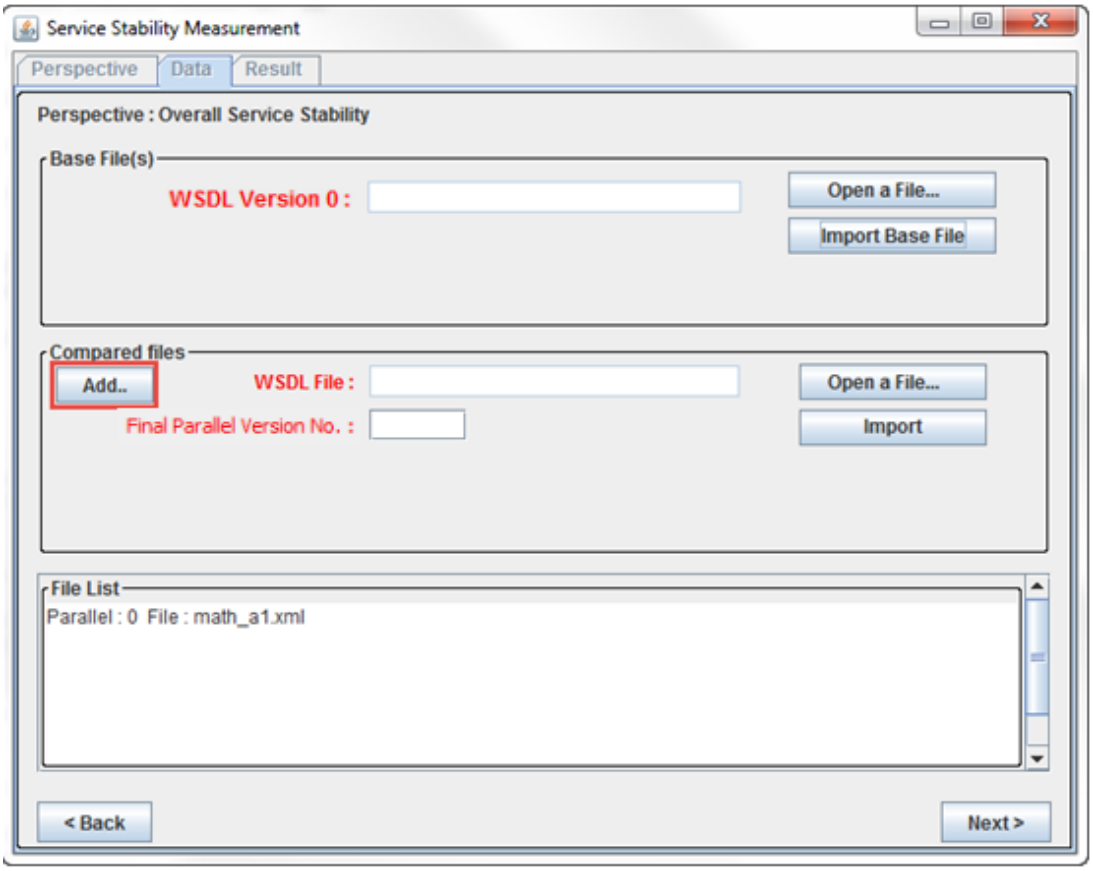
ภาพที่ 5.9 หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของโอเปอเรชัน

ตารางที่ 5.6 รายละเอียดหน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของโอเปอเรชัน

ลำดับ	ชื่อคุณลักษณะของส่วนต่อประสาน	ประเภท	คำอธิบาย	หมายเหตุ
1	WSDL Version 0	Text box	ไฟล์วิสเดิลตั้งต้นของการวัด	
2	Open a File	Button	เลือกไฟล์ตั้งต้น	
3	Import Base File	Button	นำเข้าไฟล์วิสเดิลตั้งต้น	
4	Operation Name	Combo	โอเปอเรชันที่ต้องการวัดค่าความเสถียร	กรอกได้เฉพาะมุมมองความเสถียรของโอเปอเรชัน
5	Add	Button	ปุ่มเพิ่มไฟล์วิสเดิลที่จะนำมาเปรียบเทียบ	เมื่อกด ปุ่มจะหายไปและจะแสดงเมื่อนำเข้าไฟล์ที่นำมาเปรียบเทียบสำเร็จ
6	Open a File	Button	ปุ่มเลือกไฟล์วิสเดิลที่จะนำมาเปรียบเทียบ	
7	Import	Button	ปุ่มนำเข้าไฟล์วิสเดิล	
8	Final Parallel Version No.	Text box	เวอร์ชันคู่ขนานสุดท้ายของไฟล์วิสเดิลที่จะนำมาเปรียบเทียบ	
9	File List	Text Area	แสดงชื่อไฟล์วิสเดิลที่นำเข้า	
10	Next	Button	ปุ่มดำเนินการต่อ	เมื่อกดปุ่มนี้ โปรแกรมจะทำการคำนวณค่าความเสถียรและแสดงผลในหน้าต่อไป
11	Back	Button	ปุ่มกลับไปหน้าเลือกมุมมอง	

5. หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของเซอร์วิซ

หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของเซอร์วิซแสดงดังภาพที่ 5.10 และรายละเอียดส่วนต่าง ๆ ของหน้าจอแสดงดังตารางที่ 5.7



ภาพที่ 5.10 หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของเซอร์วิซ

ตารางที่ 5.7 รายละเอียดหน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของเซอร์วิซ

ลำดับ	ชื่อคุณลักษณะของส่วนต่อประสาน	ประเภท	คำอธิบาย	หมายเหตุ
1	WSDL Version 0	Text box	ไฟล์วิสเดิลตั้งต้นของการวัด	
2	Open a File	Button	เลือกไฟล์ตั้งต้น	
3	Import Base File	Button	นำเข้าไฟล์วิสเดิลตั้งต้น	

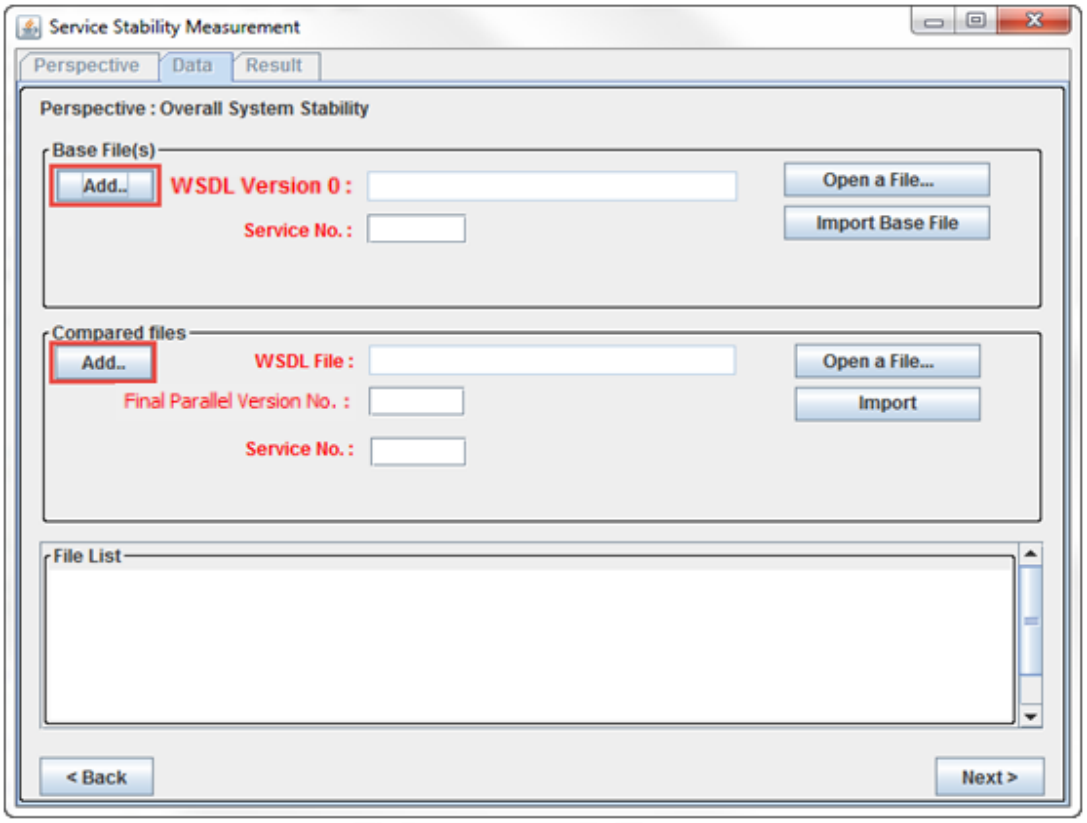
ตารางที่ 5.7 รายละเอียดหน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของเซอร์วิช (ต่อ)

ลำดับ	ชื่อคุณลักษณะของส่วนต่อประสาน	ประเภท	คำอธิบาย	หมายเหตุ
4	Add	Button	ปุ่มเพิ่มไฟล์วิสเดิลที่จะนำมาเปรียบเทียบ	เมื่อกด ปุ่มจะหายไปและจะแสดงเมื่อนำเข้าไฟล์ที่นำมาเปรียบเทียบสำเร็จ
5	Open a File	Button	ปุ่มเลือกไฟล์วิสเดิลที่จะนำมาเปรียบเทียบ	
6	Final Parallel Version No.	Text box	เวอร์ชันคู่ขนานสุดท้ายของไฟล์วิสเดิลที่จะนำมาเปรียบเทียบ	
7	Import	Button	ปุ่มนำเข้าไฟล์วิสเดิล	
8	File List	Text Area	แสดงชื่อไฟล์วิสเดิลที่นำเข้า	
9	Next	Button	ปุ่มดำเนินการต่อ	เมื่อกดปุ่มนี้ โปรแกรมจะทำการคำนวณค่าความเสถียรและแสดงผลในหน้าต่อไป
10	Back	Button	ปุ่มกลับไปหน้าเลือกมุมมอง	

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

6. หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของระบบเซอร์วิซ

หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของระบบเซอร์วิซแสดงดังภาพที่ 5.11 และรายละเอียดส่วนต่าง ๆ ของหน้าจอแสดงดังตารางที่ 5.8



ภาพที่ 5.11 หน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของระบบเซอร์วิซ

ศูนย์วิทยทรัพยากร
CHULALONGKORN UNIVERSITY

ตารางที่ 5.8 รายละเอียดหน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของระบบเซอร์วิซ

ลำดับ	ชื่อคุณลักษณะของส่วนต่อประสาน	ประเภท	คำอธิบาย	หมายเหตุ
1	Add	Button	ปุ่มเพิ่มไฟล์วิสเดิลตั้งต้นของระบบเซอร์วิซ	เมื่อกด ปุ่มจะหายไปและจะแสดงเมื่อนำเข้าไฟล์สำเร็จ
2	WSDL Version 0	Text box	ไฟล์วิสเดิลตั้งต้นของระบบเซอร์วิซ	

ตารางที่ 5.8 รายละเอียดหน้าจอนำเข้าข้อมูลสำหรับมุมมองการวัดความเสถียรโดยรวมของระบบเซอร์วิซ (ต่อ)

ลำดับ	ชื่อคุณลักษณะของส่วนต่อประสาน	ประเภท	คำอธิบาย	หมายเหตุ
3	Open a File	Button	เลือกไฟล์ตั้งต้น	
4	Service No.	Text box	เลขที่เซอร์วิซของระบบเซอร์วิซตั้งต้น	
5	Import Base File	Button	นำเข้าไฟล์วิสเดิลตั้งต้น	
6	Add	Button	ปุ่มเพิ่มไฟล์วิสเดิลที่จะนำมาเปรียบเทียบ	เมื่อกด ปุ่มจะหายไปและจะแสดงเมื่อนำเข้าไฟล์ที่นำมาเปรียบเทียบสำเร็จ
7	Open a File	Button	ปุ่มเลือกไฟล์วิสเดิลที่จะนำมาเปรียบเทียบ	
8	Final Parallel Version No.	Text box	เวอร์ชันคู่ขนานสุดท้ายของไฟล์วิสเดิลที่จะนำมาเปรียบเทียบ	
9	Service No.	Text box	เลขที่เซอร์วิซของระบบเซอร์วิซที่นำมาเปรียบเทียบ	
10	Import	Button	ปุ่มนำเข้าไฟล์วิสเดิล	
11	File List	Text Area	แสดงชื่อไฟล์วิสเดิลที่นำเข้า	
12	Next	Button	ปุ่มดำเนินการต่อ	เมื่อกดปุ่มนี้ โปรแกรมจะทำการคำนวณค่าความเสถียรและแสดงผลในหน้าต่อไป
13	Back	Button	ปุ่มกลับไปหน้าเลือกมุมมอง	

3) หน้าจอแสดงผล

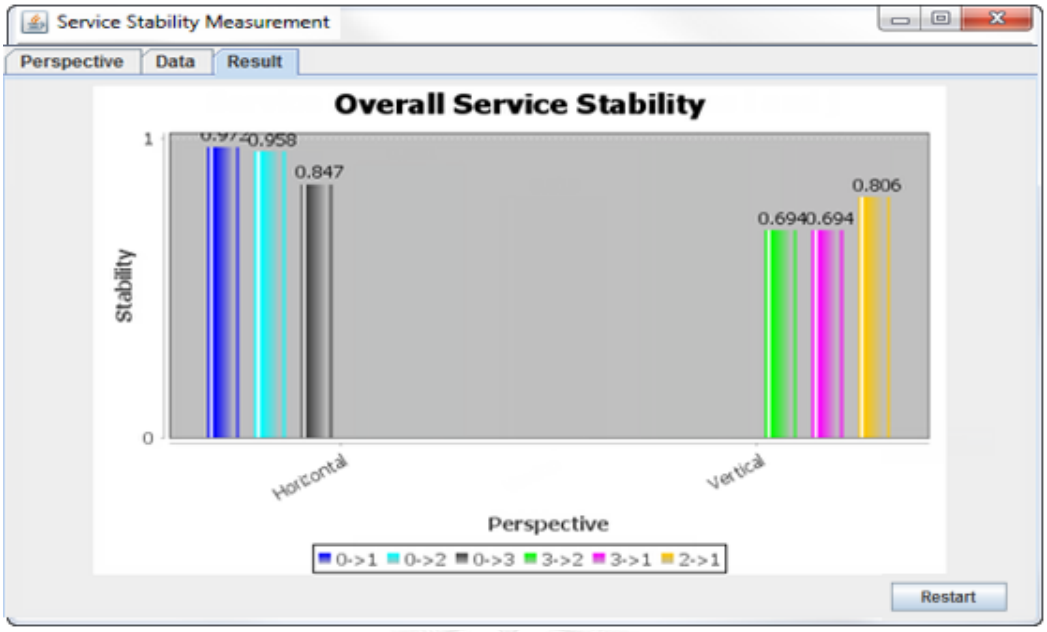
หน้าจอนี้เป็นส่วนแสดงผลของการวัดค่าความเสถียรโดยแยกระหว่างการวัดค่าความเสถียรระหว่างเวอร์ชันและการวัดค่าความเสถียรโดยรวม หน้าจอแสดงผลของการวัดค่าความเสถียรระหว่างเวอร์ชันแสดงดังภาพที่ 5.12 และรายละเอียดดังตารางที่ 5.9 หน้าจอแสดงผลการวัดค่าความเสถียรโดยรวมแสดงดังภาพที่ 5.13 และรายละเอียดดังตารางที่ 5.10

ภาพที่ 5.12 ผลการวัดค่าความเสถียรระหว่างเวอร์ชัน

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 5.9 รายละเอียดหน้าจอแสดงผลการวัดค่าความเสถียรระหว่างเวอร์ชัน

ลำดับ	ชื่อคุณลักษณะของส่วนต่อประสาน	ประเภท	คำอธิบาย	หมายเหตุ
1	Chart Name	Text	ชื่อแผนภูมิ	
2	Y-Axis	Text	แกนของค่าความเสถียร	
3	X-Axis	Text	แกนของเวอร์ชันที่เปรียบเทียบ	
4	Legend	Text	คำนิยามสีของแผนภูมิแท่ง	
5	Restart	Button	ปุ่มเริ่มการวัดใหม่	



ภาพที่ 5.13 ผลการวัดค่าความเสถียรโดยรวม

ตารางที่ 5.10 รายละเอียดหน้าจอแสดงผลการวัดค่าความเสถียรโดยรวม

ลำดับ	ชื่อคุณลักษณะของส่วนต่อประสาน	ประเภท	คำอธิบาย	หมายเหตุ
1	Chart Name	Text	ชื่อแผนภูมิ	
2	Y-Axis	Text	แกนของค่าความเสถียร	
3-1	Horizontal Series	Text	แกนของเวอร์ชันที่เปรียบเทียบระหว่างเวอร์ชันตั้งต้นและเวอร์ชันคู่ขนานสุดท้าย	
3-2	Vertical Series	Text	แกนของเวอร์ชันที่เปรียบเทียบระหว่างเวอร์ชันคู่ขนานสุดท้ายด้วยกัน	
4	Legend	Text	คำนิยามสีของแผนภูมิแท่ง	
5	Restart	Button	ปุ่มเริ่มการวัดใหม่	

5.4 สภาพแวดล้อมในการพัฒนาเครื่องมือ

รายละเอียดของฮาร์ดแวร์และซอฟต์แวร์ที่ใช้พัฒนาเครื่องมือ มีดังนี้

5.4.1 ฮาร์ดแวร์

ฮาร์ดแวร์ที่ใช้ในการพัฒนาเครื่องมือ มีรายละเอียดดังนี้

1) เครื่องคอมพิวเตอร์โน้ตบุ๊ก หน่วยประมวลผล อินเทล คอร์ i5 1.8 กิกะเฮิร์ทซ์ (Intel® Core(TM) i5-3337U CPU @ 1.8GHz)
2) หน่วยความจำ (Memory) 8.0 กิกะไบต์
3) จานบันทึกแบบแข็ง (Hard Disk) ความจุ 1 เทระไบต์

5.4.2 ซอฟต์แวร์

ซอฟต์แวร์ที่ใช้พัฒนาเครื่องมือ มีรายละเอียดดังนี้

1) ระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 7 (Microsoft Windows 7 Professional Service Pack 1)
2) ชุดเครื่องมือพัฒนาโปรแกรมภาษาจาวาเวอร์ชัน 1.7 (Java Development Kit 7 – JDK 1.7)
3) โปรแกรมอีคลิปส์ (Eclipse Java EE IDE) เวอร์ชัน Luna Service Release 2

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

5.5 ทดสอบเครื่องมือ

ผู้วิจัยทำการทดสอบเครื่องมือโดยการคำนวณค่าความเสถียรตามแบบจำลองด้วยตนเองและเปรียบเทียบกับการวัดค่าความเสถียรจากเครื่องมือ โดยใช้ตัวอย่างในการทดสอบ 2 ชุดและครอบคลุมตามมุมมองการวัดค่าความเสถียรทั้ง 6 แบบ ผลลัพธ์ที่ได้จากการเปรียบเทียบค่าความเสถียรแสดงดังตารางที่ 5.11

ตารางที่ 5.11 ผลการเปรียบเทียบค่าความเสถียรจากการคำนวณของผู้วิจัยและจากเครื่องมือ

No.	Perspective		ชุดที่ 1		ชุดที่ 2	
			ค่าจากการคำนวณของผู้วิจัย	ค่าจากการคำนวณของเครื่องมือ	ค่าจากการคำนวณของผู้วิจัย	ค่าจากการคำนวณของเครื่องมือ
1	Operation Stability		0.938	0.938	0.889	0.889
2	Service Stability		0.889	0.889	0.833	0.833
3	System Stability		0.979	0.979	0.888	0.888
4	Overall Operation Stability	Min. Δ	0.938	0.938	0.666	0.666
		Min. δ	0.938	0.938	0.833	0.833
5	Overall Service Stability	Min. Δ	0.958	0.958	0.666	0.666
		Min. δ	0.931	0.931	0.777	0.777
6	Overall System Stability	Min. Δ	0.965	0.965	0.883	0.883
		Min. δ	0.965	0.965	0.722	0.722

ผลการทดลองค่าที่ได้จากการวัดจากเครื่องมือ ตรงกับค่าที่ผู้วิจัยคำนวณตามแบบจำลองด้วยตนเอง สรุปได้ว่าเครื่องมือสามารถวัดค่าได้อย่างถูกต้องตามขอบเขตของงานวิจัย ซึ่งเมื่อนำเครื่องมือไปใช้งาน นักออกแบบเซอร์วิซจะได้รับข้อมูลที่ถูกต้องไปใช้ในการพิจารณาและก่อให้เกิดประโยชน์ในการทำงานต่อไป

บทที่ 6
สรุปผลการวิจัย ข้อจำกัด และข้อเสนอแนะ

6.1 สรุปผลการวิจัย

งานวิจัยนี้ได้เสนอแบบจำลองการวัดค่าความเสถียรของเซอร์วิซ โดยจะพิจารณาการเปลี่ยนแปลงของวิสเดิลเมื่อเกิดวิวัฒนาการจากการบำรุงรักษาซอฟต์แวร์แบบเซอร์วิซ ซึ่งสามารถพิจารณาได้ 3 มุมมองคือ ความเสถียรของโอเปอเรชัน ความเสถียรของเซอร์วิซเดี่ยว และความเสถียรของระบบเซอร์วิซ และเนื่องจากการพัฒนาระบบเซอร์วิซอาจจะมีการพัฒนาแบบคู่ขนานโดยนำเซอร์วิซตั้งต้นไปปรับปรุงแก้ไขเพื่อให้สอดคล้องตามความต้องการของแต่ละจุดประสงค์การใช้งาน ดังนั้นผู้วิจัยจึงสร้างแบบจำลองให้สามารถวัดความเสถียรโดยรวมของเวอร์ชันคู่ขนานได้ และครอบคลุมทั้ง 3 มุมมอง

ผู้วิจัยได้ทำการประเมินแบบจำลองวัดความเสถียรโดยเปรียบเทียบกับตัววัดอื่นที่ใช้ตัวคุณลักษณะของซอฟต์แวร์เชิงเซอร์วิซที่เกี่ยวเนื่องกับความสามารถในการบำรุงรักษาซอฟต์แวร์ พบว่าค่าความเสถียรของเซอร์วิซมีความสัมพันธ์เชิงบวกกับขนาดของส่วนต่อประสาน (Interface Granularity) ที่ระดับนัยสำคัญทางสถิติ 0.05 และค่าความเสถียรของเซอร์วิซมีความสัมพันธ์เชิงลบกับความสามารถในการบำรุงรักษาที่วัดด้วยสัดส่วนจำนวนบรรทัดที่แก้ไขในวิสเดิลที่ระดับนัยสำคัญทางสถิติ 0.05 ส่วนการทดสอบระหว่างค่าความเสถียรของเซอร์วิซกับการเชื่อมติดกันของข้อมูลในส่วนต่อประสานของเซอร์วิซนั้น ไม่พบความสัมพันธ์ทางสถิติ

ผู้วิจัยได้สร้างเครื่องมือสนับสนุนการวัดค่าความเสถียรโดยสามารถวัดได้ตามมุมมองแบบจำลองที่ได้นิยามไว้ ค่าความเสถียรที่วัดได้นั้นนำไปใช้ประโยชน์ได้ในอุตสาหกรรมซอฟต์แวร์ที่มีการพัฒนาระบบเซอร์วิซได้หลายด้าน ผู้ใช้สามารถนำข้อมูลที่วัดได้ไปพิจารณาในขั้นตอนออกแบบเพื่อช่วยในการตัดสินใจเลือกการปรับปรุงที่มีความเสถียรมากที่สุด หรือผู้บริหารนำข้อมูลค่าความเสถียรไปใช้ประเมินแรงงานหรือผลกระทบของการแก้ไขปรับปรุงระบบเซอร์วิซเป็นต้น

6.2 อุปสรรคและข้อจำกัด

ในการทำวิจัย ผู้วิจัยได้พบอุปสรรคและข้อจำกัดดังนี้

1) การประเมินผลแบบจำลองการวัดความเสถียรโดยการเปรียบเทียบค่าจากการวัด ยังไม่มีตัววัดความเสถียรของเซอร์วิซที่ใกล้เคียงที่จะนำมาเปรียบเทียบกับค่าความเสถียรที่ผู้วิจัยเสนอ

ได้โดยตรง ผู้วิจัยจึงประเมินแบบจำลองโดยการเปรียบเทียบความสอดคล้องกับตัววัดอื่นที่วัดคุณลักษณะที่มีผลต่อความสามารถในการบำรุงรักษา

2) ในการทดลองประเมินแบบจำลองนั้น เพื่อให้การทดลองมีความน่าเชื่อถือ ผู้วิจัยได้ขอความอนุเคราะห์ข้อมูลวิสเดิลจาก บริษัท โลคัส เทเลคอมมิวนิเคชัน อิงค์ ซึ่งมีข้อจำกัดด้านจำนวนของวิสเดิลตัวอย่างที่มีไม่มากนัก

3) เนื่องจากวิสเดิลมีความยืดหยุ่นสูง สามารถเขียนได้หลายแนวทางทำให้ยากในการนำมาเปรียบเทียบ ดังนั้นในการสร้างเครื่องมือสนับสนุนการวัดค่าความเสถียรจึงต้องกำหนดให้ 1 วิสเดิลคือ 1 เซอร์วิซ เพื่อเป็นมาตรฐานในการคำนวณ

4) เครื่องมือสนับสนุนการวัดค่าความเสถียร สามารถวัดค่าความเสถียรได้ตามมุมมองต่าง ๆ เท่านั้น ไม่สามารถแนะนำการปรับปรุงแก้ไขเพื่อให้ค่าความเสถียรเพิ่มขึ้นหรือลดลงได้

6.3 ข้อเสนอแนะ

แบบจำลองการวัดความเสถียรของเซอร์วิซ และเครื่องมือสนับสนุนการวัดค่าความเสถียรของเซอร์วิซ สามารถนำไปต่อยอดพัฒนาให้ครอบคลุมและสะดวกแก่การใช้งานได้ดียิ่งขึ้น เช่น

1) ในปัจจุบันการพัฒนาซอฟต์แวร์เชิงเซอร์วิซ มีการพัฒนาและเปลี่ยนแปลงอยู่ตลอดและสามารถใช้โปรโตคอลอื่นนอกจาก โซป (SOAP) เช่น เรสท์ (REST) เป็นต้น ซึ่งการอธิบายส่วนต่อประสานจะไม่ใช่วิสเดิล ดังนั้นแบบจำลองควรพัฒนาให้สามารถวัดค่าความเสถียรของเซอร์วิซในโปรโตคอลอื่นได้

2) พัฒนาเครื่องมือให้สามารถระบุเอลิเมนต์ที่แตกต่างของวิสเดิลที่ทำการเปรียบเทียบกันได้

3) พัฒนาเครื่องมือให้สามารถนำข้อมูลการวัดออกจากระบบสู่ภายนอกในรูปแบบไฟล์เอ็กเซล (.xls) หรือ ไฟล์ที่คั่นด้วยเครื่องหมายจุลภาค (CSV) เป็นต้น

Chulalongkorn University

รายการอ้างอิง

[1] S. S. Yau and J. S. Collofello, "Some Stability Measures for Software Maintenance," *Software Engineering, IEEE Transactions on,* vol. SE-6, pp. 545-552, 1980.

[2] S. S. Yau and J. S. Collofello, "Design Stability Measures for Software Maintenance," *Software Engineering, IEEE Transactions on,* vol. SE-11, pp. 849-856, 1985.

[3] M. O. Elish and D. Rine, "Investigation of metrics for object-oriented design logical stability," in *Software Maintenance and Reengineering, 2003. Proceedings. Seventh European Conference on*, 2003, pp. 193-200.

[4] M. Alshayeb, M. Naji, M. O. Elish, and J. Al-Ghamdi, "Towards measuring object-oriented class stability," *Software, IET,* vol. 5, pp. 415-424, 2011.

[5] D. Kelly, "A study of design characteristics in evolving software using stability as a criterion," *Software Engineering, IEEE Transactions on,* vol. 32, pp. 315-329, 2006.

[6] W3C. (2004). *Web Services Architecture*. Available: http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

[7] W3C. (2007). *Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language*. Available: http://www.w3.org/TR/2007/REC-wsdl20-20070626

[8] A. Skonnardservice. (2003). *Understanding WSDL*. Available: https://msdn.microsoft.com/en-us/library/ms996486.aspx

[9] P. Leitner, A. Michlmayr, F. Rosenberg, and S. Dustdar, "End-to-End Versioning Support for Web Services," in *Services Computing, 2008. SCC '08. IEEE International Conference on*, 2008, pp. 59-66.

[10] Z. Ma and K. Ben, "Research on maintainability evaluation of service-oriented software," in *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, 2010, pp. 510-512.

[11] M. Perepletchikov and C. Ryan, "A Controlled Experiment for Evaluating the Impact of Coupling on the Maintainability of Service-Oriented Software,"

Software Engineering, IEEE Transactions on, vol. 37, pp. 449-465, 2011.

[12] T. Karthikeyan and J. Geetha, "A Quantitative Measurement and Validation of Granularity in Service Oriented Architecture," *International Journal of Computer Science Issues on,* vol. 9, p. 377, 2012.

[13] M. Perepletchikov, C. Ryan, and K. Frampton, "Cohesion Metrics for Predicting Maintainability of Service-Oriented Software," in *Quality Software, 2007. QSIC '07. Seventh International Conference on*, 2007, pp. 328-335.

ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

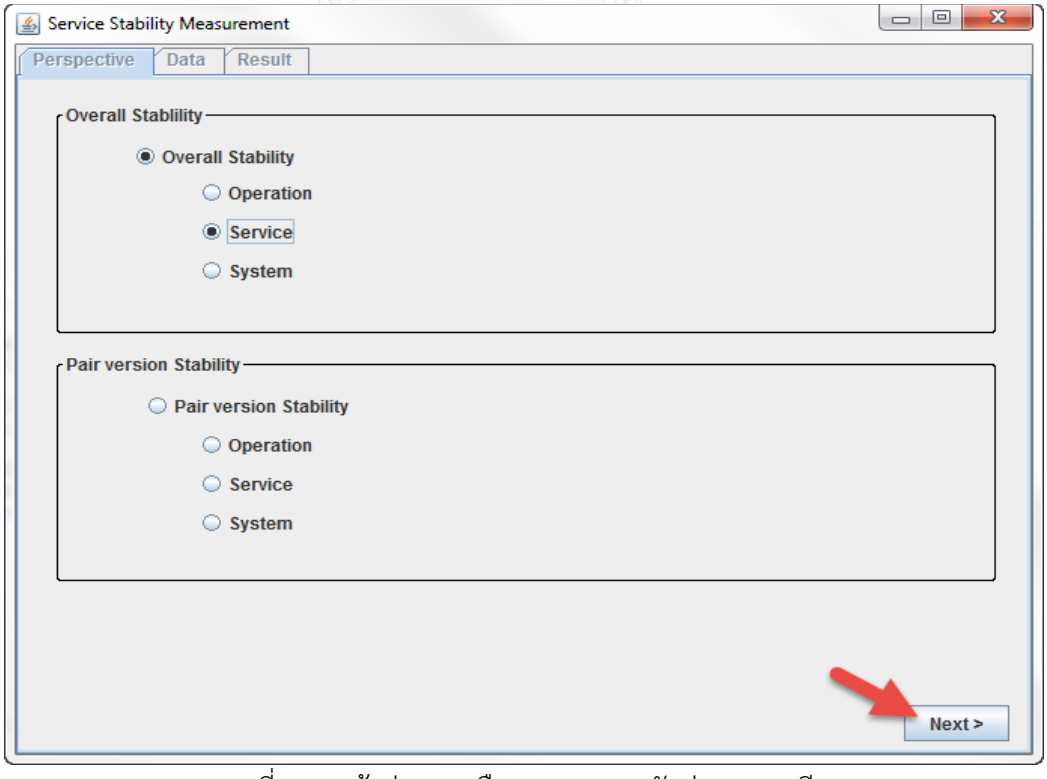
ภาคผนวก ก
การใช้งานเครื่องมือสนับสนุนการวัดค่าความเสถียร

การสร้างเครื่องมือสนับสนุนการวัดค่าความเสถียรเป็นส่วนหนึ่งของการวิจัย เพื่อพิสูจน์ว่าสามารถนำแบบจำลองไปใช้งานได้จริง เพื่อให้เกิดความเข้าใจและใช้งานได้อย่างถูกต้อง ผู้วิจัยจึงได้จัดทำรายละเอียดการใช้งานดังต่อไปนี้

1) หน้าเริ่มต้นดังภาพที่ ก.1 ผู้ใช้งานจะต้องเลือกมุมมองของการวัดค่าความเสถียรซึ่งมีด้วยกัน 6 มุมมองคือ

- ความเสถียรโดยรวมของโอเปอเรชัน
- ความเสถียรโดยรวมของเซอร์วิซ
- ความเสถียรโดยรวมของระบบเซอร์วิซ
- ความเสถียรระหว่างเวอร์ชันของโอเปอเรชัน
- ความเสถียรระหว่างเวอร์ชันของเซอร์วิซ
- ความเสถียรระหว่างเวอร์ชันของระบบเซอร์วิซ

หลังจากเลือกมุมมองแล้วให้กดปุ่ม “Next”



ภาพที่ ก.1 หน้าต่างการเลือกมุมมองการวัดค่าความเสถียร

2) หลังจากเลือกมุมมองการวัดค่าความเสถียรแล้ว ผู้ใช้งานจะต้องทำการนำเข้าข้อมูลและระบุข้อมูลให้แก่วิสเดิลดังภาพที่ ก.2 สำหรับกลุ่มการวัดความเสถียรระหว่างเวอร์ชัน และภาพที่ ก.3 สำหรับกลุ่มการวัดความเสถียรโดยรวม ดังนี้

- กลุ่มการวัดความเสถียรระหว่างเวอร์ชัน

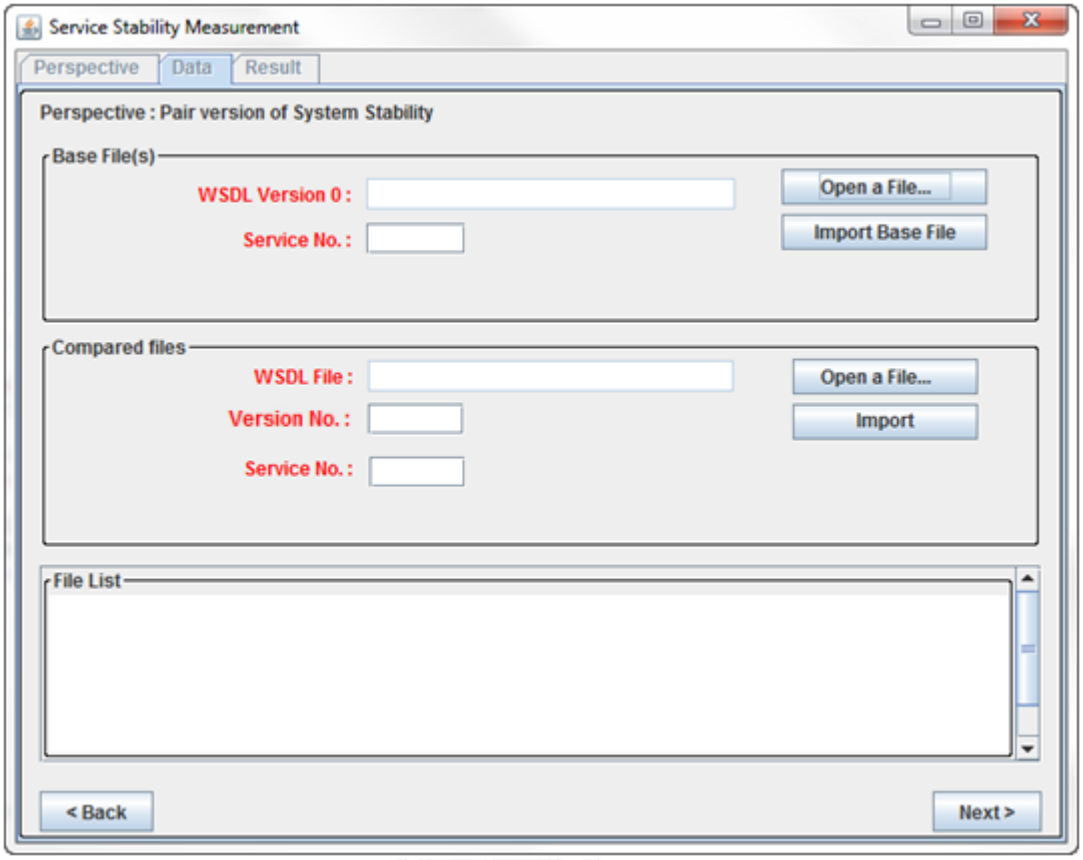
Base File

- WSDL Version 0 คือ วิสเดิลเวอร์ชันตั้งต้นที่จะใช้ในการคำนวณ
- Service No. คือ เลขที่เซอร์วิซของวิสเดิลตั้งต้น มีไว้เพื่อให้ทำการเปรียบเทียบได้ถูกเซอร์วิซในกรณีที่วัดค่าความเสถียรในมุมมองของระบบเซอร์วิซ
- Operation Name คือ ชื่อโอเปอเรชันที่เลือกมาวัดค่าความเสถียรในกรณีที่เลือกวัดค่าความเสถียรของโอเปอเรชัน

Compared files

- WSDL File คือ วิสเดิลที่จะนำมาเปรียบเทียบกับวิสเดิลตั้งต้น
- Version No. คือ เวอร์ชันของวิสเดิลที่นำมาเปรียบเทียบกับวิสเดิลตั้งต้น
- Service No. คือ เลขที่เซอร์วิซของวิซเดิลที่จะนำมาเปรียบเทียบในกรณีที่เลือกวัดค่าความเสถียรของระบบเซอร์วิซ
- File List คือ ไฟล์ที่ได้ทำการนำเข้าระบบ

หลังจากนำเข้าและระบุข้อมูลเรียบร้อยแล้ว กดปุ่ม “Next” เพื่อทำการวัดและแสดงผล



ภาพที่ ก.2 หน้าจอการนำเข้าและระบุข้อมูลให้แก่วิสเดิลในกลุ่มการวัดความเสถียรระหว่างเวอร์ชัน

- กลุ่มการวัดความเสถียรโดยรวม

Base File

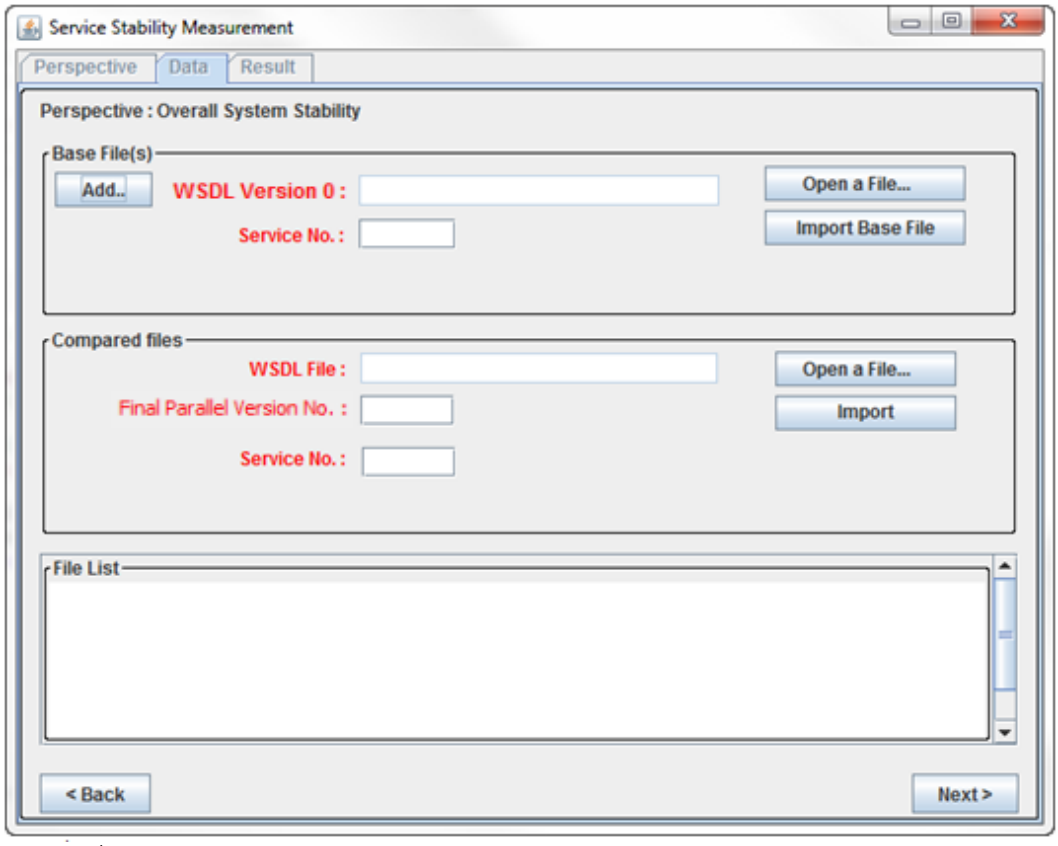
- WSDL Version 0 คือ วิสเดิลตั้งต้นที่จะใช้ในการคำนวณ
- Service No. คือ เลขที่เซอร์วิซของวิสเดิลตั้งต้น มีไว้เพื่อให้ทำการเปรียบเทียบได้ถูกเซอร์วิซในกรณีที่วัดค่าความเสถียรในมุมมองของระบบเซอร์วิซ
- Operation Name คือ ชื่อโอเปอเรชันที่เลือกมาวัดค่าความเสถียรในกรณีที่เลือกวัดค่าความเสถียรของโอเปอเรชัน

Compared file

- WSDL File คือ วิสเดิลที่จะนำมาเปรียบเทียบกับวิสเดิลตั้งต้น

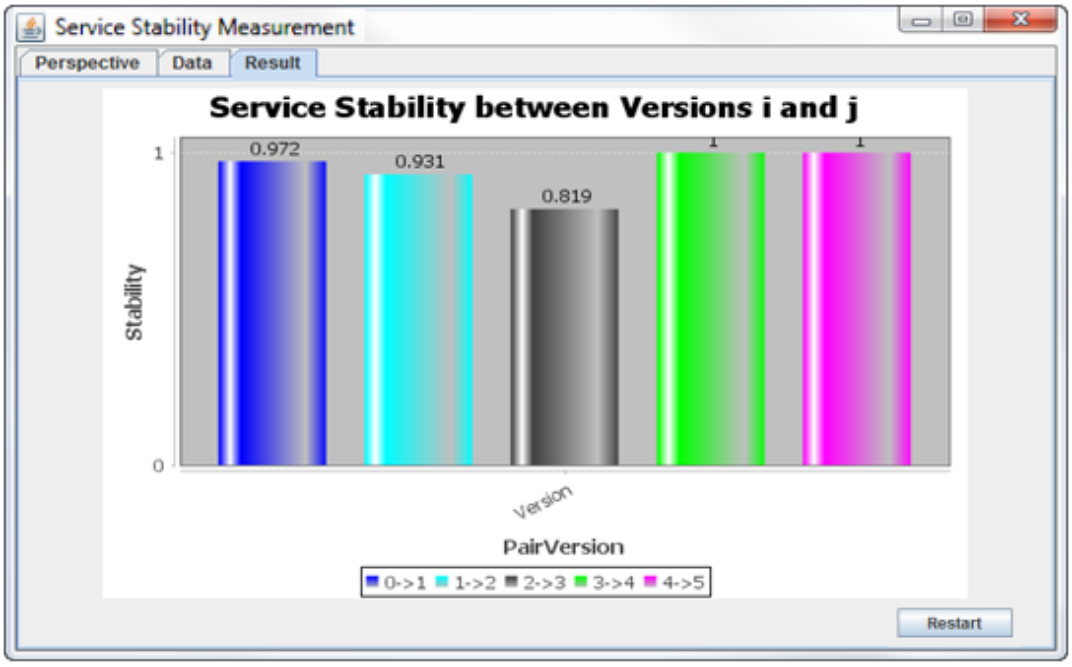
- Final Parallel Version No. คือ เลขที่เวอร์ชันคู่ขนานสุดท้าย ที่จะนำมาวัดค่าความเสถียรโดยรวม
- Service No. คือ เลขที่เซอร์วิซของวิซเดิลที่จะนำมาเปรียบเทียบในกรณีที่เลือกวัดค่าความเสถียรของระบบเซอร์วิซ
- File List คือ ไฟล์ที่ได้ทำการนำเข้าระบบ

หลังจากนำเข้าและระบุข้อมูลเรียบร้อยแล้ว กดปุ่ม “Next” เพื่อทำการวัดและแสดงผล



ภาพที่ ก.3 หน้าจอการนำเข้าและระบุข้อมูลให้แก่วิสเดิลในกลุ่มการวัดความเสถียรโดยรวม

3) หลังจากการคำนวณ โปรแกรมจะแสดงผลดังภาพที่ ก.4 ซึ่งจะแสดงเป็นแผนภูมิแท่งเปรียบเทียบค่าความเสถียรตามมุมมองที่ได้เลือกมา



ภาพที่ ก.4 หน้าจอแสดงผลการวัดค่าความเสถียร

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ข

วิสเดิลที่ใช้ในการประเมินแบบจำลอง

วิสเดิลที่ใช้ในการประเมินแบบจำลองการวัดความเสถียรของระบบเซอร์วิซมีด้วยกันทั้งหมด 6 วิสเดิลดังนี้

วิสเดิลที่ 1

รายการแก้ไข	
เดิม	<element name="createdDate" nillable="true" type="xsd:string"/> <element name="dateOfLoss" nillable="true" type="xsd:string"/>
แก้ไข	<element name="createdDate" nillable="true" type="xsd:date"/> <element name="dateOfLoss" nillable="true" type="xsd:date"/>

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://tempuri.org/TeleproWebService"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://tempuri.org/TeleproWebService"
xmlns:intf="http://tempuri.org/TeleproWebService"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!--WSDL created by Apache Axis version: 1.4
Built on Apr 22, 2006 (06:55:48 PDT)-->
 <wsdl:types>
  <schema elementFormDefault="qualified"
targetNamespace="http://tempuri.org/TeleproWebService"
xmlns="http://www.w3.org/2001/XMLSchema">
   <element name="getToken">
    <complexType>
     <sequence>
```

```
      <element name="getTokenRequest" type="impl:GetTokenRequest"/>
     </sequence>
    </complexType>
   </element>
   <complexType name="GetTokenRequest">
    <sequence>
     <element name="password" nillable="true" type="xsd:string"/>
     <element name="serviceName" nillable="true" type="xsd:string"/>
     <element name="userName" nillable="true" type="xsd:string"/>
    </sequence>
   </complexType>
   <element name="getTokenResponse">
    <complexType>
     <sequence>
      <element name="getTokenReturn" type="impl:GetTokenResponse"/>
     </sequence>
    </complexType>
   </element>
   <complexType name="GetTokenResponse">
    <sequence>
     <element name="errorCode" nillable="true" type="xsd:string"/>
     <element name="errorMessage" nillable="true" type="xsd:string"/>
     <element name="tokenCode" nillable="true" type="xsd:string"/>
    </sequence>
   </complexType>
   <element name="getClaimNotiNo">
    <complexType>
     <sequence>
      <element name="getClaimNotiNoRequest"
type="impl:GetClaimNotiNoRequest"/>
     </sequence>
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
    </complexType>
   </element>
   <complexType name="GetClaimNotiNoRequest">
    <sequence>
     <element name="accidentDescription" nillable="true" type="xsd:string"/>
     <element name="chassisNo" nillable="true" type="xsd:string"/>
     <element name="dateOfLoss" nillable="true" type="xsd:string"/>
     <element name="district" nillable="true" type="xsd:string"/>
     <element name="driverLicenseNumber" nillable="true" type="xsd:string"/>
     <element name="driverMobile" nillable="true" type="xsd:string"/>
     <element name="driverName" nillable="true" type="xsd:string"/>
     <element name="informerMobile" nillable="true" type="xsd:string"/>
     <element name="informerName" nillable="true" type="xsd:string"/>
     <element name="informerTel" nillable="true" type="xsd:string"/>
     <element name="insurerBrId" nillable="true" type="xsd:string"/>
     <element name="policyNo" nillable="true" type="xsd:string"/>
     <element name="province" nillable="true" type="xsd:string"/>
     <element name="regNo" nillable="true" type="xsd:string"/>
     <element name="registrationProvinceCode" nillable="true" type="xsd:string"/>
     <element name="relationshipWithInsured" nillable="true" type="xsd:string"/>
     <element name="repairerBrId" nillable="true" type="xsd:string"/>
     <element name="repairerId" nillable="true" type="xsd:string"/>
     <element name="timeOfLoss" nillable="true" type="xsd:string"/>
     <element name="tokenCode" nillable="true" type="xsd:string"/>
    </sequence>
   </complexType>
   <element name="getClaimNotiNoResponse">
    <complexType>
     <sequence>
      <element name="getClaimNotiNoReturn"
type="impl:GetClaimNotiNoResponse"/>
```

```
    </sequence>
   </complexType>
  </element>
  <complexType name="GetClaimNotiNoResponse">
   <sequence>
    <element name="claimNotiNo" nillable="true" type="xsd:string"/>
    <element name="errorCode" nillable="true" type="xsd:string"/>
    <element name="errorMessage" nillable="true" type="xsd:string"/>
    <element name="policyNo" nillable="true" type="xsd:string"/>
   </sequence>
  </complexType>
  <element name="getGeneralInfo">
   <complexType>
    <sequence>
     <element name="getGeneralInfoRequest"
type="impl:GetGeneralInfoRequest"/>
    </sequence>
   </complexType>
  </element>
  <complexType name="GetGeneralInfoRequest">
   <sequence>
    <element name="claimNo" nillable="true" type="xsd:string"/>
    <element name="tokenCode" nillable="true" type="xsd:string"/>
   </sequence>
  </complexType>
  <element name="getGeneralInfoResponse">
   <complexType>
    <sequence>
     <element name="getGeneralInfoReturn"
type="impl:GetGeneralInfoResponse"/>
    </sequence>
```

```
   </complexType>
  </element>
  <complexType name="GetGeneralInfoResponse">
   <sequence>
    <element name="cPolicyNo" nillable="true" type="xsd:string"/>
    <element name="chassisNo" nillable="true" type="xsd:string"/>
    <element name="claimNo" nillable="true" type="xsd:string"/>
    <element name="claimNotiNo" nillable="true" type="xsd:string"/>
    <element name="cosi" nillable="true" type="xsd:string"/>
    <element name="createdDate" nillable="true" type="xsd:string"/>
    <element name="createdTime" nillable="true" type="xsd:string"/>
    <element name="dateOfLoss" nillable="true" type="xsd:string"/>
    <element name="errorCode" nillable="true" type="xsd:string"/>
    <element name="errorMessage" nillable="true" type="xsd:string"/>
    <element name="informerName" nillable="true" type="xsd:string"/>
    <element name="insuranceType" nillable="true" type="xsd:string"/>
    <element name="insuredName" nillable="true" type="xsd:string"/>
    <element name="itemNo" nillable="true" type="xsd:string"/>
    <element name="policyNo" nillable="true" type="xsd:string"/>
    <element name="policyType" nillable="true" type="xsd:string"/>
    <element name="registrationNo" nillable="true" type="xsd:string"/>
    <element name="registrationProvinceCode" nillable="true" type="xsd:string"/>
    <element name="timeOfLoss" nillable="true" type="xsd:string"/>
   </sequence>
  </complexType>
 </schema>
</wsdl:types>
  <wsdl:message name="getClaimNotiNoRequest">
     <wsdl:part element="impl:getClaimNotiNo" name="parameters">
     </wsdl:part>
  </wsdl:message>
```

```
  <wsdl:message name="getTokenRequest">
    <wsdl:part element="impl:getToken" name="parameters">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="getGeneralInfoRequest">
    <wsdl:part element="impl:getGeneralInfo" name="parameters">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="getClaimNotiNoResponse">
    <wsdl:part element="impl:getClaimNotiNoResponse" name="parameters">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="getGeneralInfoResponse">
    <wsdl:part element="impl:getGeneralInfoResponse" name="parameters">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="getTokenResponse">
    <wsdl:part element="impl:getTokenResponse" name="parameters">
    </wsdl:part>
  </wsdl:message>
  <wsdl:portType name="TeleproWebService">
    <wsdl:operation name="getToken">
      <wsdl:input message="impl:getTokenRequest" name="getTokenRequest">
     </wsdl:input>
      <wsdl:output message="impl:getTokenResponse"
name="getTokenResponse">
     </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="getClaimNotiNo">
      <wsdl:input message="impl:getClaimNotiNoRequest"
name="getClaimNotiNoRequest">
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
      </wsdl:input>
        <wsdl:output message="impl:getClaimNotiNoResponse"
name="getClaimNotiNoResponse">
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="getGeneralInfo">
      <wsdl:input message="impl:getGeneralInfoRequest"
name="getGeneralInfoRequest">
      </wsdl:input>
        <wsdl:output message="impl:getGeneralInfoResponse"
name="getGeneralInfoResponse">
      </wsdl:output>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="TeleproWebServiceSoapBinding"
type="impl:TeleproWebService">
    <wsdlsoap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="getToken">
      <wsdlsoap:operation soapAction=""/>
      <wsdl:input name="getTokenRequest">
        <wsdlsoap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="getTokenResponse">
        <wsdlsoap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="getClaimNotiNo">
      <wsdlsoap:operation soapAction=""/>

      <wsdl:input name="getClaimNotiNoRequest">
```

```
            <wsdlsoap:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="getClaimNotiNoResponse">
            <wsdlsoap:body use="literal"/>
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="getGeneralInfo">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="getGeneralInfoRequest">
            <wsdlsoap:body use="literal"/>
         </wsdl:input>
         <wsdl:output name="getGeneralInfoResponse">
            <wsdlsoap:body use="literal"/>
         </wsdl:output>
      </wsdl:operation>
   </wsdl:binding>
   <wsdl:service name="TeleproWebServiceService">
      <wsdl:port binding="impl:TeleproWebServiceSoapBinding"
name="TeleproWebService">
         <wsdlsoap:address
location="http://10.35.0.36:8080/MCSIWebService/services/TeleproWebService"/>
      </wsdl:port>
   </wsdl:service>
</wsdl:definitions>
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

วิสเดิลที่ 2

รายการแก้ไข	
เดิม	<element name="dueDate" nillable="true" type="xsd:string"/> <element name="totalAmount" nillable="true" type="xsd:string"/> <element name="outDueDate" nillable="true" type="xsd:string"/> <element name="accidentDate" nillable="true" type="xsd:string"/> <element name="okDate" nillable="true" type="xsd:string"/> <element name="totalAmount" nillable="true" type="xsd:string"/> <wsdl:part name="reg_time" type="xsd:string"> <wsdl:part name="datetime" type="xsd:string">
แก้ไข	<element name="dueDate" nillable="true" type="xsd:date"/> <element name="totalAmount" nillable="true" type="xsd:float"/> <element name="outDueDate" nillable="true" type="xsd:date"/> <element name="accidentDate" nillable="true" type="xsd:date"/> <element name="okDate" nillable="true" type="xsd:date"/> <element name="totalAmount" nillable="true" type="xsd:float"/> <wsdl:part name="reg_time" type="xsd:datetime"> <wsdl:part name="datetime" type="xsd:datetime">

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
targetNamespace="http://192.168.100.74:8080/mkt/webservice/IVR.jws"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://192.168.100.74:8080/mkt/webservice/IVR.jws"
xmlns:intf="http://192.168.100.74:8080/mkt/webservice/IVR.jws"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
```

```
xmlns:tns1="http://dto.ivr.locus.com"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!--WSDL created by Apache Axis version: 1.4
Built on Apr 22, 2006 (06:55:48 PDT)-->
 <wsdl:types>
  <schema targetNamespace="http://dto.ivr.locus.com"
xmlns="http://www.w3.org/2001/XMLSchema">
   <import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
   <complexType name="VerifyAgentCodeDTO">
    <sequence>
     <element name="agentCode" nillable="true" type="xsd:string"/>
     <element name="error" nillable="true" type="xsd:string"/>
     <element name="pinCode" nillable="true" type="xsd:string"/>
     <element name="strID" nillable="true" type="xsd:string"/>
     <element name="activeYN" nillable="true" type="xsd:string"/>
    </sequence>
   </complexType>
   <complexType name="PinCodeRegDTO">
    <sequence>
     <element name="error" nillable="true" type="xsd:string"/>
    </sequence>
   </complexType>
   <complexType name="PinCodeServDTO">
    <sequence>
     <element name="agentCode" nillable="true" type="xsd:string"/>
     <element name="citizenID" nillable="true" type="xsd:string"/>
     <element name="error" nillable="true" type="xsd:string"/>
     <element name="licenNo" nillable="true" type="xsd:string"/>
     <element name="mobileNo" nillable="true" type="xsd:string"/>
```

```
  <element name="strid" nillable="true" type="xsd:string"/>
 </sequence>
</complexType>
<complexType name="PolicyStatusDTO">
 <sequence>
  <element name="dueDate" nillable="true" type="xsd:string"/>
  <element name="error" nillable="true" type="xsd:string"/>
  <element name="statusCode" nillable="true" type="xsd:string"/>
  <element name="totalAmount" nillable="true" type="xsd:string"/>
 </sequence>
</complexType>
<complexType name="PolicyOutDTO">
 <sequence>
  <element name="error" nillable="true" type="xsd:string"/>
  <element name="outCashValue" nillable="true" type="xsd:string"/>
  <element name="outDueDate" nillable="true" type="xsd:string"/>
  <element name="outFlag" nillable="true" type="xsd:string"/>
  <element name="outPolicyNo" nillable="true" type="xsd:string"/>
 </sequence>
</complexType>
<complexType name="PolicyClaimDTO">
 <sequence>
  <element name="accidentDate" nillable="true" type="xsd:string"/>
  <element name="error" nillable="true" type="xsd:string"/>
  <element name="inFlag" nillable="true" type="xsd:string"/>
  <element name="okDate" nillable="true" type="xsd:string"/>
  <element name="policyNo" nillable="true" type="xsd:string"/>
  <element name="remarkNumber" nillable="true" type="xsd:string"/>
  <element name="totalAmount" nillable="true" type="xsd:string"/>
 </sequence>
</complexType>
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
 </schema>
</wsdl:types>
 <wsdl:message name="pincodeRegResponse">
    <wsdl:part name="pincodeRegReturn" type="tns1:PinCodeRegDTO">
    </wsdl:part>
 </wsdl:message>
 <wsdl:message name="pincodeRegRequest">
    <wsdl:part name="agentCode" type="xsd:string">
    </wsdl:part>
    <wsdl:part name="pinCode" type="xsd:string">
    </wsdl:part>
 </wsdl:message>
 <wsdl:message name="insertCallbackRequest">
    <wsdl:part name="account_no" type="xsd:string">
    </wsdl:part>
    <wsdl:part name="voice_mail_flag" type="xsd:string">
    </wsdl:part>
    <wsdl:part name="file_path" type="xsd:string">
    </wsdl:part>
    <wsdl:part name="reg_time" type="xsd:string">
    </wsdl:part>
    <wsdl:part name="phone_no" type="xsd:string">
    </wsdl:part>
 </wsdl:message>
 <wsdl:message name="checkPolicyStatusResponse">
    <wsdl:part name="checkPolicyStatusReturn" type="tns1:PolicyStatusDTO">
    </wsdl:part>
 </wsdl:message>
 <wsdl:message name="verifyAgentCodeResponse">
    <wsdl:part name="verifyAgentCodeReturn" type="tns1:VerifyAgentCodeDTO">
    </wsdl:part>
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
</wsdl:message>
<wsdl:message name="faxOnDemandInsertResponse">
   <wsdl:part name="faxOnDemandInsertReturn" type="xsd:string">
   </wsdl:part>
</wsdl:message>
<wsdl:message name="faxOnDemandInsertRequest">
   <wsdl:part name="fax_code" type="xsd:string">
   </wsdl:part>
   <wsdl:part name="tel_no" type="xsd:string">
   </wsdl:part>
   <wsdl:part name="datetime" type="xsd:string">
   </wsdl:part>
</wsdl:message>
<wsdl:message name="pinCodeServResponse">
   <wsdl:part name="pinCodeServReturn" type="tns1:PinCodeServDTO">
   </wsdl:part>
</wsdl:message>
<wsdl:message name="checkPolicyStatusRequest">
   <wsdl:part name="policyNo" type="xsd:string">
   </wsdl:part>
</wsdl:message>
<wsdl:message name="checkPolicyClaimResponse">
   <wsdl:part name="checkPolicyClaimReturn" type="tns1:PolicyClaimDTO">
   </wsdl:part>
</wsdl:message>
<wsdl:message name="verifyAgentCodeRequest">
   <wsdl:part name="agentCode" type="xsd:string">
   </wsdl:part>
</wsdl:message>
<wsdl:message name="checkPolicyMatureRefundDividendResponse">
   <wsdl:part name="checkPolicyMatureRefundDividendReturn"
```

Chulalongkorn University

```
type="tns1:PolicyOutDTO">
      </wsdl:part>
   </wsdl:message>
   <wsdl:message name="checkPolicyClaimRequest">
      <wsdl:part name="policy_no" type="xsd:string">
      </wsdl:part>
      <wsdl:part name="flag" type="xsd:string">
      </wsdl:part>
   </wsdl:message>
   <wsdl:message name="pinCodeServRequest">
      <wsdl:part name="agentCode" type="xsd:string">
      </wsdl:part>
   </wsdl:message>
   <wsdl:message name="checkPolicyMatureRefundDividendRequest">
      <wsdl:part name="policyNo" type="xsd:string">
      </wsdl:part>
      <wsdl:part name="MRDFlag" type="xsd:string">
      </wsdl:part>
   </wsdl:message>
   <wsdl:message name="insertCallbackResponse">
      <wsdl:part name="insertCallbackReturn" type="xsd:string">
      </wsdl:part>
   </wsdl:message>
   <wsdl:portType name="IVR">
      <wsdl:operation name="verifyAgentCode" parameterOrder="agentCode">
         <wsdl:input message="impl:verifyAgentCodeRequest"
name="verifyAgentCodeRequest">
       </wsdl:input>
         <wsdl:output message="impl:verifyAgentCodeResponse"
name="verifyAgentCodeResponse">
       </wsdl:output>
```

```
    </wsdl:operation>
    <wsdl:operation name="pincodeReg" parameterOrder="agentCode pinCode">
      <wsdl:input message="impl:pincodeRegRequest"
name="pincodeRegRequest">
     </wsdl:input>
      <wsdl:output message="impl:pincodeRegResponse"
name="pincodeRegResponse">
     </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="pinCodeServ" parameterOrder="agentCode">
      <wsdl:input message="impl:pinCodeServRequest"
name="pinCodeServRequest">
     </wsdl:input>
      <wsdl:output message="impl:pinCodeServResponse"
name="pinCodeServResponse">
     </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="insertCallback" parameterOrder="account_no
voice_mail_flag file_path reg_time phone_no">
      <wsdl:input message="impl:insertCallbackRequest"
name="insertCallbackRequest">
     </wsdl:input>
      <wsdl:output message="impl:insertCallbackResponse"
name="insertCallbackResponse">
     </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="checkPolicyStatus" parameterOrder="policyNo">
      <wsdl:input message="impl:checkPolicyStatusRequest"
name="checkPolicyStatusRequest">
     </wsdl:input>
      <wsdl:output message="impl:checkPolicyStatusResponse"
```

```
name="checkPolicyStatusResponse">
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="checkPolicyMatureRefundDividend"
parameterOrder="policyNo MRDFlag">
      <wsdl:input message="impl:checkPolicyMatureRefundDividendRequest"
name="checkPolicyMatureRefundDividendRequest">
      </wsdl:input>
      <wsdl:output message="impl:checkPolicyMatureRefundDividendResponse"
name="checkPolicyMatureRefundDividendResponse">
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="checkPolicyClaim" parameterOrder="policy_no flag">
      <wsdl:input message="impl:checkPolicyClaimRequest"
name="checkPolicyClaimRequest">
      </wsdl:input>
      <wsdl:output message="impl:checkPolicyClaimResponse"
name="checkPolicyClaimResponse">
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="faxOnDemandInsert" parameterOrder="fax_code
tel_no datetime">
      <wsdl:input message="impl:faxOnDemandInsertRequest"
name="faxOnDemandInsertRequest">
      </wsdl:input>
      <wsdl:output message="impl:faxOnDemandInsertResponse"
name="faxOnDemandInsertResponse">
      </wsdl:output>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="IVRSoapBinding" type="impl:IVR">
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
      <wsdlsoap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
      <wsdl:operation name="verifyAgentCode">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="verifyAgentCodeRequest">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="verifyAgentCodeResponse">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.168.100.74:8080/mkt/webservice/IVR.jws" use="encoded"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="pincodeReg">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="pincodeRegRequest">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="pincodeRegResponse">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.168.100.74:8080/mkt/webservice/IVR.jws" use="encoded"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="pinCodeServ">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="pinCodeServRequest">
```

```
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="pinCodeServResponse">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.168.100.74:8080/mkt/webservice/IVR.jws" use="encoded"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="insertCallback">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="insertCallbackRequest">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="insertCallbackResponse">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.168.100.74:8080/mkt/webservice/IVR.jws" use="encoded"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="checkPolicyStatus">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="checkPolicyStatusRequest">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="checkPolicyStatusResponse">
```

```
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.168.100.74:8080/mkt/webservice/IVR.jws" use="encoded"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="checkPolicyMatureRefundDividend">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="checkPolicyMatureRefundDividendRequest">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="checkPolicyMatureRefundDividendResponse">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.168.100.74:8080/mkt/webservice/IVR.jws" use="encoded"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="checkPolicyClaim">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="checkPolicyClaimRequest">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="checkPolicyClaimResponse">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.168.100.74:8080/mkt/webservice/IVR.jws" use="encoded"/>
        </wsdl:output>
      </wsdl:operation>
```

```
      <wsdl:operation name="faxOnDemandInsert">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="faxOnDemandInsertRequest">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
         </wsdl:input>
         <wsdl:output name="faxOnDemandInsertResponse">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.168.100.74:8080/mkt/webservice/IVR.jws" use="encoded"/>
         </wsdl:output>
      </wsdl:operation>
   </wsdl:binding>
   <wsdl:service name="IVRService">
      <wsdl:port binding="impl:IVRSoapBinding" name="IVR">
         <wsdlsoap:address
location="http://192.168.100.74:8080/mkt/webservice/IVR.jws"/>
      </wsdl:port>
   </wsdl:service>
</wsdl:definitions>
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

วิสเดิลที่ 3

รายการแก้ไข	
เดิม	<wsdl:part name="QueueID" type="xsd:string"> <wsdl:part name="QueueID" type="xsd:string"> <wsdl:part name="QueueID" type="xsd:string">
แก้ไข	<wsdl:part name="QueueID" type="xsd:int"> <wsdl:part name="QueueID" type="xsd:int"> <wsdl:part name="QueueID" type="xsd:int">

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
targetNamespace="http://192.168.100.74:8080/mkt/webservice/multimedia.jws"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://192.168.100.74:8080/mkt/webservice/multimedia.jws"
xmlns:intf="http://192.168.100.74:8080/mkt/webservice/multimedia.jws"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!--WSDL created by Apache Axis version: 1.4
Built on Apr 22, 2006 (06:55:48 PDT)-->
   <wsdl:message name="acknowledgeRemoveWorkItemRequest">
      <wsdl:part name="WorkItemID" type="xsd:string">
      </wsdl:part>
     <wsdl:part name="QueueID" type="xsd:string">
      </wsdl:part>
      <wsdl:part name="RemoveReason" type="xsd:string">
      </wsdl:part>
   </wsdl:message>
```

```
  <wsdl:message name="acknowledgeRemoveWorkItemResponse">
    <wsdl:part name="acknowledgeRemoveWorkItemReturn" type="xsd:string">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="acknowledgeWorkItemStatusResponse">
    <wsdl:part name="acknowledgeWorkItemStatusReturn" type="xsd:string">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="acknowledgeTimeoutResponse">
    <wsdl:part name="acknowledgeTimeoutReturn" type="xsd:string">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="acknowledgeTimeoutRequest">
    <wsdl:part name="WorkItemID" type="xsd:string">
    </wsdl:part>
    <wsdl:part name="QueueID" type="xsd:string">
    </wsdl:part>
    <wsdl:part name="StillQueued" type="xsd:boolean">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="acknowledgeWorkItemStatusRequest">
    <wsdl:part name="WorkItemID" type="xsd:string">
    </wsdl:part>
    <wsdl:part name="State" type="xsd:string">
    </wsdl:part>
    <wsdl:part name="QueueID" type="xsd:string">
    </wsdl:part>
  </wsdl:message>
  <wsdl:portType name="multimedia">
    <wsdl:operation name="acknowledgeTimeout" parameterOrder="WorkItemID
QueueID StillQueued">
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
        <wsdl:input message="impl:acknowledgeTimeoutRequest"
name="acknowledgeTimeoutRequest">
       </wsdl:input>
        <wsdl:output message="impl:acknowledgeTimeoutResponse"
name="acknowledgeTimeoutResponse">
       </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="acknowledgeRemoveWorkItem"
parameterOrder="WorkItemID QueueID RemoveReason">
        <wsdl:input message="impl:acknowledgeRemoveWorkItemRequest"
name="acknowledgeRemoveWorkItemRequest">
       </wsdl:input>
        <wsdl:output message="impl:acknowledgeRemoveWorkItemResponse"
name="acknowledgeRemoveWorkItemResponse">
       </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="acknowledgeWorkItemStatus"
parameterOrder="WorkItemID State QueueID">
        <wsdl:input message="impl:acknowledgeWorkItemStatusRequest"
name="acknowledgeWorkItemStatusRequest">
       </wsdl:input>
        <wsdl:output message="impl:acknowledgeWorkItemStatusResponse"
name="acknowledgeWorkItemStatusResponse">
       </wsdl:output>
      </wsdl:operation>
   </wsdl:portType>
   <wsdl:binding name="multimediaSoapBinding" type="impl:multimedia">
      <wsdlsoap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
      <wsdl:operation name="acknowledgeTimeout">
        <wsdlsoap:operation soapAction=""/>
```

```
        <wsdl:input name="acknowledgeTimeoutRequest">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="acknowledgeTimeoutResponse">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.168.100.74:8080/mkt/webservice/multimedia.jws"
use="encoded"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="acknowledgeRemoveWorkItem">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="acknowledgeRemoveWorkItemRequest">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
        </wsdl:input>

        <wsdl:output name="acknowledgeRemoveWorkItemResponse">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.168.100.74:8080/mkt/webservice/multimedia.jws"
use="encoded"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="acknowledgeWorkItemStatus">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="acknowledgeWorkItemStatusRequest">
          <wsdlsoap:body
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="acknowledgeWorkItemStatusResponse">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.168.100.74:8080/mkt/webservice/multimedia.jws"
use="encoded"/>
        </wsdl:output>
      </wsdl:operation>
   </wsdl:binding>
   <wsdl:service name="multimediaService">
      <wsdl:port binding="impl:multimediaSoapBinding" name="multimedia">
        <wsdlsoap:address
location="http://192.168.100.74:8080/mkt/webservice/multimedia.jws"/>
      </wsdl:port>
   </wsdl:service>
</wsdl:definitions>
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

วิสเดิลที่ 4

รายการแก้ไข	
เดิม	<element name="decPoint" nillable="true" type="xsd:string"/> <wsdl:part name="callInDate" type="xsd:string"/> <wsdl:part name="birthDate" type="xsd:string"/>
แก้ไข	<element name="decPoint" nillable="true" type="xsd:float"/> <wsdl:part name="callInDate" type="xsd:date"/> <wsdl:part name="birthDate" type="xsd:date"/>

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://192.6.6.71/common/service/IVR.jws"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://192.6.6.71/common/service/IVR.jws"
xmlns:intf="http://192.6.6.71/common/service/IVR.jws"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns1="http://dto.ws.com" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!--WSDL created by Apache Axis version: 1.4
Built on Apr 22, 2006 (06:55:48 PDT)-->
 <wsdl:types>
  <schema targetNamespace="http://dto.ws.com"
xmlns="http://www.w3.org/2001/XMLSchema">
   <import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
   <complexType name="CardGetSumDTO">
    <sequence>
     <element name="day" nillable="true" type="xsd:string"/>
     <element name="decPoint" nillable="true" type="xsd:string"/>
     <element name="error" nillable="true" type="xsd:string"/>
```

```
    <element name="hasTLCard" nillable="true" type="xsd:string"/>
    <element name="month" nillable="true" type="xsd:string"/>
    <element name="numPoint" nillable="true" type="xsd:string"/>
    <element name="year" nillable="true" type="xsd:string"/>
   </sequence>
  </complexType>
 </schema>
</wsdl:types>
  <wsdl:message name="VerifyInfoResponse">
     <wsdl:part name="VerifyInfoReturn" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="InsertCallbackRequest">
     <wsdl:part name="strPolicyNo" type="xsd:string"/>
     <wsdl:part name="callInNumber" type="xsd:string"/>
     <wsdl:part name="callBackNumber" type="xsd:string"/>
     <wsdl:part name="callInDate" type="xsd:string"/>
     <wsdl:part name="callInTime" type="xsd:string"/>
     <wsdl:part name="body" type="xsd:string"/>
     <wsdl:part name="attachment" type="xsd:string"/>
     <wsdl:part name="sourceContact" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="CheckPinCodeResponse">
     <wsdl:part name="CheckPinCodeReturn" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="CheckPolicyResponse">
     <wsdl:part name="CheckPolicyReturn" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="CheckPinCodeRequest">
     <wsdl:part name="strPolicyNo" type="xsd:string"/>
     <wsdl:part name="strPinCode" type="xsd:string"/>
  </wsdl:message>
```

```
  <wsdl:message name="InsertCallbackResponse">
    <wsdl:part name="InsertCallbackReturn" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="VerifyInfoRequest">
    <wsdl:part name="strPolicyNo" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="cardGetSumPointResponse">
    <wsdl:part name="cardGetSumPointReturn" type="tns1:CardGetSumDTO"/>
  </wsdl:message>
  <wsdl:message name="CheckPolicyRequest">
    <wsdl:part name="strPolicyNo" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="cardGetSumPointRequest">
    <wsdl:part name="barCode" type="xsd:string"/>
    <wsdl:part name="birthDate" type="xsd:string"/>
  </wsdl:message>
  <wsdl:portType name="IVR">
    <wsdl:operation name="CheckPolicy" parameterOrder="strPolicyNo">
      <wsdl:input message="impl:CheckPolicyRequest"
name="CheckPolicyRequest"/>
      <wsdl:output message="impl:CheckPolicyResponse"
name="CheckPolicyResponse"/>
    </wsdl:operation>
    <wsdl:operation name="cardGetSumPoint" parameterOrder="barCode
birthDate">
      <wsdl:input message="impl:cardGetSumPointRequest"
name="cardGetSumPointRequest"/>
      <wsdl:output message="impl:cardGetSumPointResponse"
name="cardGetSumPointResponse"/>
    </wsdl:operation>
    <wsdl:operation name="InsertCallback" parameterOrder="strPolicyNo
```

```
callInNumber callBackNumber callInDate callInTime body attachment
sourceContact">
        <wsdl:input message="impl:InsertCallbackRequest"
name="InsertCallbackRequest"/>
        <wsdl:output message="impl:InsertCallbackResponse"
name="InsertCallbackResponse"/>
      </wsdl:operation>
      <wsdl:operation name="CheckPinCode" parameterOrder="strPolicyNo
strPinCode">
        <wsdl:input message="impl:CheckPinCodeRequest"
name="CheckPinCodeRequest"/>
        <wsdl:output message="impl:CheckPinCodeResponse"
name="CheckPinCodeResponse"/>
      </wsdl:operation>
      <wsdl:operation name="VerifyInfo" parameterOrder="strPolicyNo">
        <wsdl:input message="impl:VerifyInfoRequest" name="VerifyInfoRequest"/>
        <wsdl:output message="impl:VerifyInfoResponse"
name="VerifyInfoResponse"/>
      </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="IVRSoapBinding" type="impl:IVR">
      <wsdlsoap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
      <wsdl:operation name="CheckPolicy">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="CheckPolicyRequest">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="CheckPolicyResponse">
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.6.6.71/common/service/IVR.jws" use="encoded"/>
          </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="cardGetSumPoint">
          <wsdlsoap:operation soapAction=""/>
          <wsdl:input name="cardGetSumPointRequest">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
          </wsdl:input>
          <wsdl:output name="cardGetSumPointResponse">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.6.6.71/common/service/IVR.jws" use="encoded"/>
          </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="InsertCallback">
          <wsdlsoap:operation soapAction=""/>
          <wsdl:input name="InsertCallbackRequest">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
          </wsdl:input>
          <wsdl:output name="InsertCallbackResponse">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.6.6.71/common/service/IVR.jws" use="encoded"/>
          </wsdl:output>
        </wsdl:operation>
```

```
      <wsdl:operation name="CheckPinCode">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="CheckPinCodeRequest">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="CheckPinCodeResponse">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.6.6.71/common/service/IVR.jws" use="encoded"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="VerifyInfo">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="VerifyInfoRequest">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="VerifyInfoResponse">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.6.6.71/common/service/IVR.jws" use="encoded"/>
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="IVRService">
      <wsdl:port binding="impl:IVRSoapBinding" name="IVR">
        <wsdlsoap:address location="http://192.6.6.71/common/service/IVR.jws"/>
      </wsdl:port>
```

Chulalongkorn University

```
    </wsdl:service>
</wsdl:definitions>
```

วิสเดิลที่ 5

รายการแก้ไข	
เดิม	<element name="birthDate" nillable="true" type="xsd:string"/> <element name="lastComplaintDay" nillable="true" type="xsd:string"/> <wsdl:part name="birthDate" type="xsd:string"/>
แก้ไข	<element name="birthDate" nillable="true" type="xsd:date"/> <element name="lastComplaintDay" nillable="true" type="xsd:date"/> <wsdl:part name="birthDate" type="xsd:date"/>

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
targetNamespace="http://192.6.6.71/common/service/TLICCOnlineWS.jws"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://192.6.6.71/common/service/TLICCOnlineWS.jws"
xmlns:intf="http://192.6.6.71/common/service/TLICCOnlineWS.jws"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns1="http://dto.ws.com" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!--WSDL created by Apache Axis version: 1.4
Built on Apr 22, 2006 (06:55:48 PDT)-->
 <wsdl:types>
  <schema targetNamespace="http://dto.ws.com"
xmlns="http://www.w3.org/2001/XMLSchema">
   <import namespace="http://192.6.6.71/common/service/TLICCOnlineWS.jws"/>
   <import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
   <complexType name="TLICCOnlineCustomerDto">
```

```
 <sequence>
  <element name="birthDate" nillable="true" type="xsd:string"/>
  <element name="cardId" nillable="true" type="xsd:string"/>
  <element name="email" nillable="true" type="xsd:string"/>
  <element name="firstName" nillable="true" type="xsd:string"/>
  <element name="lastName" nillable="true" type="xsd:string"/>
  <element name="personId" nillable="true" type="xsd:string"/>
  <element name="phoneNo" nillable="true" type="xsd:string"/>
  <element name="policyNo" nillable="true" type="xsd:string"/>
  <element name="title" nillable="true" type="xsd:string"/>
  <element name="address" nillable="true" type="xsd:string"/>
  <element name="custId" nillable="true" type="xsd:string"/>
  <element name="custCode" nillable="true" type="xsd:string"/>
  <element name="lastComplaintDay" nillable="true" type="xsd:string"/>
 </sequence>
</complexType>
<complexType name="VocCategoryCodeBookDto">
 <sequence>
  <element name="codeId" nillable="true" type="xsd:string"/>
  <element name="codeName" nillable="true" type="xsd:string"/>
  <element name="parentCodeId" nillable="true" type="xsd:string"/>
  <element name="etc1" nillable="true" type="xsd:string"/>
 </sequence>
</complexType>
<complexType name="TLICCOnlineComplaintDto">
 <sequence>
  <element name="srNumber" nillable="true" type="xsd:string"/>
  <element name="statusCode" nillable="true" type="xsd:string"/>
  <element name="statusName" nillable="true" type="xsd:string"/>
  <element name="subStatusCode" nillable="true" type="xsd:string"/>
  <element name="subStatusName" nillable="true" type="xsd:string"/>
```

```
    </sequence>
   </complexType>
  </schema>
  <schema
targetNamespace="http://192.6.6.71/common/service/TLICCOnlineWS.jws"
xmlns="http://www.w3.org/2001/XMLSchema">
   <import namespace="http://dto.ws.com"/>
   <import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
   <complexType name="ArrayOf_tns1_VocCategoryCodeBookDto">
    <complexContent>
     <restriction base="soapenc:Array">
      <attribute ref="soapenc:arrayType"
wsdl:arrayType="tns1:VocCategoryCodeBookDto[]"/>
     </restriction>
    </complexContent>
   </complexType>
  </schema>
 </wsdl:types>
   <wsdl:message name="saveNewComplainRequest">
      <wsdl:part name="policyCode" type="xsd:string"/>
      <wsdl:part name="custId" type="xsd:string"/>
      <wsdl:part name="prename" type="xsd:string"/>
      <wsdl:part name="firstName" type="xsd:string"/>
      <wsdl:part name="lastname" type="xsd:string"/>
      <wsdl:part name="birthDate" type="xsd:string"/>
      <wsdl:part name="address" type="xsd:string"/>
      <wsdl:part name="telNo" type="xsd:string"/>
      <wsdl:part name="email" type="xsd:string"/>
      <wsdl:part name="requestFullName" type="xsd:string"/>
      <wsdl:part name="typeCode" type="xsd:string"/>
      <wsdl:part name="categoryCode" type="xsd:string"/>
```

```
      <wsdl:part name="complainAgentFullName" type="xsd:string"/>
      <wsdl:part name="branchOrWorkPath" type="xsd:string"/>
      <wsdl:part name="description" type="xsd:string"/>
      <wsdl:part name="contactRelationCode" type="xsd:string"/>
   </wsdl:message>
   <wsdl:message name="getCustomerInfoByPolicyNoResponse">
      <wsdl:part name="getCustomerInfoByPolicyNoReturn"
type="tns1:TLICCOnlineCustomerDto"/>
   </wsdl:message>
   <wsdl:message name="getListRelationResponse">
      <wsdl:part name="getListRelationReturn"
type="impl:ArrayOf_tns1_VocCategoryCodeBookDto"/>
   </wsdl:message>
   <wsdl:message name="getListRelationRequest">
   </wsdl:message>
   <wsdl:message name="getListVocCategoryRequest">
      <wsdl:part name="vocSubTypeName" type="xsd:string"/>
   </wsdl:message>
   <wsdl:message name="countOpenedClaimin30DayByCusCodeRequest">
      <wsdl:part name="custId" type="xsd:string"/>
   </wsdl:message>
   <wsdl:message name="getCustomerInfoByPolicyNoRequest">
      <wsdl:part name="policyNo" type="xsd:string"/>
      <wsdl:part name="cardId" type="xsd:string"/>
   </wsdl:message>
   <wsdl:message name="getListVocCategoryResponse">
      <wsdl:part name="getListVocCategoryReturn"
type="impl:ArrayOf_tns1_VocCategoryCodeBookDto"/>
   </wsdl:message>
   <wsdl:message name="countOpenedClaimin30DayByCusCodeResponse">
      <wsdl:part name="countOpenedClaimin30DayByCusCodeReturn"
```

```
type="xsd:int"/>
   </wsdl:message>
   <wsdl:message name="saveNewComplainResponse">
      <wsdl:part name="saveNewComplainReturn" type="xsd:string"/>
   </wsdl:message>
   <wsdl:message name="getComplaintStatusRequest">
      <wsdl:part name="complaintNo" type="xsd:string"/>
   </wsdl:message>
   <wsdl:message name="getComplaintStatusResponse">
      <wsdl:part name="getComplaintStatusReturn"
type="tns1:TLICCOnlineComplaintDto"/>
   </wsdl:message>
   <wsdl:portType name="TLICCOnlineWS">
      <wsdl:operation name="getCustomerInfoByPolicyNo"
parameterOrder="policyNo cardId">
         <wsdl:input message="impl:getCustomerInfoByPolicyNoRequest"
name="getCustomerInfoByPolicyNoRequest"/>
         <wsdl:output message="impl:getCustomerInfoByPolicyNoResponse"
name="getCustomerInfoByPolicyNoResponse"/>
      </wsdl:operation>
      <wsdl:operation name="getListVocCategory"
parameterOrder="vocSubTypeName">
         <wsdl:input message="impl:getListVocCategoryRequest"
name="getListVocCategoryRequest"/>
         <wsdl:output message="impl:getListVocCategoryResponse"
name="getListVocCategoryResponse"/>
      </wsdl:operation>
      <wsdl:operation name="getListRelation">
         <wsdl:input message="impl:getListRelationRequest"
name="getListRelationRequest"/>
         <wsdl:output message="impl:getListRelationResponse"
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
name="getListRelationResponse"/>
      </wsdl:operation>
      <wsdl:operation name="getComplaintStatus" parameterOrder="complaintNo">
        <wsdl:input message="impl:getComplaintStatusRequest"
name="getComplaintStatusRequest"/>
        <wsdl:output message="impl:getComplaintStatusResponse"
name="getComplaintStatusResponse"/>
      </wsdl:operation>
      <wsdl:operation name="saveNewComplain" parameterOrder="policyCode
custId prename firstName lastname birthDate address telNo email
requestFullName typeCode categoryCode complainAgentFullName
branchOrWorkPath description contactRelationCode">
        <wsdl:input message="impl:saveNewComplainRequest"
name="saveNewComplainRequest"/>
        <wsdl:output message="impl:saveNewComplainResponse"
name="saveNewComplainResponse"/>
      </wsdl:operation>
      <wsdl:operation name="countOpenedClaimin30DayByCusCode"
parameterOrder="custId">
        <wsdl:input message="impl:countOpenedClaimin30DayByCusCodeRequest"
name="countOpenedClaimin30DayByCusCodeRequest"/>
        <wsdl:output
message="impl:countOpenedClaimin30DayByCusCodeResponse"
name="countOpenedClaimin30DayByCusCodeResponse"/>
      </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="TLICCOnlineWSSoapBinding" type="impl:TLICCOnlineWS">
      <wsdlsoap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
      <wsdl:operation name="getCustomerInfoByPolicyNo">
        <wsdlsoap:operation soapAction=""/>
```

```
        <wsdl:input name="getCustomerInfoByPolicyNoRequest">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="getCustomerInfoByPolicyNoResponse">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.6.6.71/common/service/TLICCOnlineWS.jws"
use="encoded"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="getListVocCategory">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="getListVocCategoryRequest">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="getListVocCategoryResponse">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.6.6.71/common/service/TLICCOnlineWS.jws"
use="encoded"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="getListRelation">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="getListRelationRequest">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
```

```
namespace="http://DefaultNamespace" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="getListRelationResponse">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.6.6.71/common/service/TLICCOnlineWS.jws"
use="encoded"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="getComplaintStatus">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="getComplaintStatusRequest">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="getComplaintStatusResponse">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.6.6.71/common/service/TLICCOnlineWS.jws"
use="encoded"/>
        </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="saveNewComplain">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="saveNewComplainRequest">
          <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
        </wsdl:input>
        <wsdl:output name="saveNewComplainResponse">
```

```
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.6.6.71/common/service/TLICCOnlineWS.jws"
use="encoded"/>
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="countOpenedClaimin30DayByCusCode">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="countOpenedClaimin30DayByCusCodeRequest">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
         </wsdl:input>
         <wsdl:output name="countOpenedClaimin30DayByCusCodeResponse">
            <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://192.6.6.71/common/service/TLICCOnlineWS.jws"
use="encoded"/>
         </wsdl:output>
      </wsdl:operation>
   </wsdl:binding>
   <wsdl:service name="TLICCOnlineWSService">
      <wsdl:port binding="impl:TLICCOnlineWSSoapBinding"
name="TLICCOnlineWS">
         <wsdlsoap:address
location="http://192.6.6.71/common/service/TLICCOnlineWS.jws"/>
      </wsdl:port>
   </wsdl:service>
</wsdl:definitions>
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

วิสเดิลที่ 6.1

รายการแก้ไข	
เดิม	<xs:element name="statusDate" type="xs:string" minOccurs="0"></xs:element> <xs:element name="documentSendDate" type="xs:string" minOccurs="0"></xs:element> <xs:element name="birthdate" type="xs:string" minOccurs="0"></xs:element> <xs:element name="contactConvenientDate" type="xs:string" minOccurs="0"></xs:element> <xs:element name="age" type="xs:string" minOccurs="0"></xs:element> <xs:element name="admitDate" type="xs:string" minOccurs="0"></xs:element> <xs:element name="statusDate" type="xs:string" minOccurs="0"></xs:element> <xs:element name="documentSendDate" type="xs:string" minOccurs="0"></xs:element>
แก้ไข	<xs:element name="statusDate" type="xs:date" minOccurs="0"></xs:element> <xs:element name="documentSendDate" type="xs:date" minOccurs="0"></xs:element> <xs:element name="birthdate" type="xs:date" minOccurs="0"></xs:element> <xs:element name="contactConvenientDate" type="xs:date" minOccurs="0"></xs:element> <xs:element name="age" type="xs:int" minOccurs="0"></xs:element> <xs:element name="admitDate" type="xs:date" minOccurs="0"></xs:element>

	<xs:element name="statusDate" type="xs:date" minOccurs="0"></xs:element> <xs:element name="documentSendDate" type="xs:date" minOccurs="0"></xs:element>

```
<?xml version="1.0" encoding="UTF-8"?><!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2.8 svn-revision#13980. --><xs:schema xmlns:tns="http://mca.locus.services" xmlns:xs="http://www.w3.org/2001/XMLSchema" version="1.0" targetNamespace="http://mca.locus.services">

<xs:element name="rejectApplication" type="tns:rejectApplication"></xs:element>

<xs:element name="rejectApplicationResponse" type="tns:rejectApplicationResponse"></xs:element>

<xs:element name="removeProspectCustomer" type="tns:removeProspectCustomer"></xs:element>

<xs:element name="removeProspectCustomerResponse" type="tns:removeProspectCustomerResponse"></xs:element>

<xs:element name="sendHospitalAdmitData" type="tns:sendHospitalAdmitData"></xs:element>

<xs:element name="sendHospitalAdmitDataResponse" type="tns:sendHospitalAdmitDataResponse"></xs:element>

<xs:element name="sendProspectCustomer" type="tns:sendProspectCustomer"></xs:element>
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
<xs:element name="sendProspectCustomerResponse"
type="tns:sendProspectCustomerResponse"></xs:element>

<xs:element name="updateHospitalAdmitData"
type="tns:updateHospitalAdmitData"></xs:element>

<xs:element name="updateHospitalAdmitDataResponse"
type="tns:updateHospitalAdmitDataResponse"></xs:element>

<xs:complexType name="updateHospitalAdmitData">
<xs:sequence>
<xs:element name="request" type="tns:updateHospitalAdmitRequest"
minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="updateHospitalAdmitRequest">
<xs:complexContent>
<xs:extension base="tns:commonRequest">
<xs:sequence>
<xs:element name="cardNo" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="policyNo" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="claimNo" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="salePersonId" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="status" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="statusDate" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="rejectReason" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="documentSendDate" type="xs:string"
minOccurs="0"></xs:element>
</xs:sequence>
```

```
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="commonRequest">
<xs:sequence></xs:sequence>
</xs:complexType>

<xs:complexType name="updateHospitalAdmitDataResponse">
<xs:sequence>
<xs:element name="response" type="tns:UpdateHospitalAdmitResponseType"
minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="UpdateHospitalAdmitResponseType">
<xs:complexContent>
<xs:extension base="tns:commonResponse">
<xs:sequence></xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="commonResponse">
<xs:sequence>
<xs:element name="status" type="tns:status" minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="status">
<xs:sequence>
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
<xs:element name="code" type="xs:int"></xs:element>
<xs:element name="message" type="xs:string" minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="sendProspectCustomer">
<xs:sequence>
<xs:element name="request" type="tns:sendProspectCustomerRequest"
minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="sendProspectCustomerRequest">
<xs:complexContent>
<xs:extension base="tns:commonRequest">
<xs:sequence>
<xs:element name="prospects">
<xs:complexType>
<xs:sequence>
<xs:element name="prospect" type="tns:prospectCustomer" minOccurs="0"
maxOccurs="unbounded"></xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="prospectCustomer">
<xs:sequence>
```

```
<xs:element name="prospectId" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="agentId" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="remainDays" type="xs:int"></xs:element>
<xs:element name="title" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="firstname" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="lastname" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="birthdate" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="gender" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="age" type="xs:int" minOccurs="0"></xs:element>
<xs:element name="zipcode" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="phoneNo" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="mobileNo" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="address" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="businessType" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="parentId" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="noRiderFlag" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="contact" type="tns:contact" minOccurs="0"></xs:element>
<xs:element name="insuranceDetail" type="tns:insuranceDetail"
minOccurs="0"></xs:element>
<xs:element name="riders">
<xs:complexType>
<xs:sequence>
<xs:element name="rider" type="tns:rider" minOccurs="0"
maxOccurs="unbounded"></xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="contact">
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
<xs:sequence>
<xs:element name="contactTitle" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="contactFirstname" type="xs:string"
minOccurs="0"></xs:element>
<xs:element name="contactLastname" type="xs:string"
minOccurs="0"></xs:element>
<xs:element name="contactPhoneNo" type="xs:string"
minOccurs="0"></xs:element>
<xs:element name="contactMobileNo" type="xs:string"
minOccurs="0"></xs:element>
<xs:element name="contactEmail" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="contactConvenientDate" type="xs:string"
minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="insuranceDetail">
<xs:sequence>
<xs:element name="reason" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="insureType" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="sumInsured" type="xs:decimal" minOccurs="0"></xs:element>
<xs:element name="taxDeduction" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="paymentType" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="totalPremium" type="xs:decimal"
minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="rider">
<xs:sequence>
<xs:element name="riderType" type="xs:string" minOccurs="0"></xs:element>
```

```
<xs:element name="riderPremium" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="age" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="gender" type="xs:string" minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="sendProspectCustomerResponse">
<xs:sequence>
<xs:element name="response" type="tns:SendProspectCustomerResponseType"
minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="SendProspectCustomerResponseType">
<xs:complexContent>
<xs:extension base="tns:commonResponse">
<xs:sequence></xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="removeProspectCustomer">
<xs:sequence>
<xs:element name="request" type="tns:removeProspectCustomerRequest"
minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="removeProspectCustomerRequest">
<xs:complexContent>
<xs:extension base="tns:commonRequest">
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
<xs:sequence>
<xs:element name="prospectIdList">
<xs:complexType>
<xs:sequence>
<xs:element name="prospectId" type="xs:string" minOccurs="0"
maxOccurs="unbounded"></xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="removeProspectCustomerResponse">
<xs:sequence>
<xs:element name="response" type="tns:RemoveProspectCustomerResponseType"
minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="RemoveProspectCustomerResponseType">
<xs:complexContent>
<xs:extension base="tns:commonResponse">
<xs:sequence></xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="sendHospitalAdmitData">
<xs:sequence>
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
<xs:element name="request" type="tns:sendHospitalAdmitRequest"
minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="sendHospitalAdmitRequest">
<xs:complexContent>
<xs:extension base="tns:commonRequest">
<xs:sequence>
<xs:element name="cardNo" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="policyNo" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="claimNo" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="admitDate" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="customerFirstname" type="xs:string"
minOccurs="0"></xs:element>
<xs:element name="customerLastname" type="xs:string"
minOccurs="0"></xs:element>
<xs:element name="customerId" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="salePersonId" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="hospitalName" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="status" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="statusDate" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="rejectReason" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="documentSendDate" type="xs:string"
minOccurs="0"></xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="sendHospitalAdmitDataResponse">
```

```
<xs:sequence>
<xs:element name="response" type="tns:SendHospitalAdmitResponseType"
minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="SendHospitalAdmitResponseType">
<xs:complexContent>
<xs:extension base="tns:commonResponse">
<xs:sequence></xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="rejectApplication">
<xs:sequence>
<xs:element name="request" type="tns:rejectApplicationRequest"
minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="rejectApplicationRequest">
<xs:complexContent>
<xs:extension base="tns:commonRequest">
<xs:sequence>
<xs:element name="receiptNo" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="salePersonId" type="xs:string" minOccurs="0"></xs:element>
<xs:element name="rejectReason" type="xs:string" minOccurs="0"></xs:element>
</xs:sequence>
</xs:extension>
</xs:complexContent>
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
</xs:complexType>

<xs:complexType name="rejectApplicationResponse">
<xs:sequence>
<xs:element name="response" type="tns:RejectApplicationResponseType"
minOccurs="0"></xs:element>
</xs:sequence>
</xs:complexType>

<xs:complexType name="RejectApplicationResponseType">
<xs:complexContent>
<xs:extension base="tns:commonResponse">
<xs:sequence></xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:schema>
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

วิสเดิลที่ 6.2


```
<?xml version="1.0" encoding="UTF-8"?><!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2.8 svn-revision#13980. --><!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2.8 svn-revision#13980. --><definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://webservice.tli.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://webservice.tli.com/" name="MCAServiceImplService">
<import namespace="http://mca.locus.services"
location="http://113.53.239.243:8080/MCAService?wsdl=1"></import>
<binding xmlns:ns1="http://mca.locus.services" name="MCAServiceImplPortBinding"
type="ns1:MCAService">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http"
style="document"></soap:binding>
<operation name="sendProspectCustomer">
<soap:operation soapAction=""></soap:operation>
<input>
<soap:body use="literal"></soap:body>
</input>
<output>
<soap:body use="literal"></soap:body>
</output>
</operation>
<operation name="removeProspectCustomer">
<soap:operation soapAction=""></soap:operation>
```

```
<input>
<soap:body use="literal"></soap:body>
</input>
<output>
<soap:body use="literal"></soap:body>
</output>
</operation>
<operation name="sendHospitalAdmitData">
<soap:operation soapAction=""></soap:operation>
<input>
<soap:body use="literal"></soap:body>
</input>
<output>
<soap:body use="literal"></soap:body>
</output>
</operation>
<operation name="updateHospitalAdmitData">
<soap:operation soapAction=""></soap:operation>
<input>
<soap:body use="literal"></soap:body>
</input>
<output>
<soap:body use="literal"></soap:body>
</output>
</operation>
<operation name="rejectApplication">
<soap:operation soapAction=""></soap:operation>
<input>
<soap:body use="literal"></soap:body>
</input>
<output>
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
<soap:body use="literal"></soap:body>
</output>
</operation>
</binding>
<service name="MCAServiceImplService">
<port name="MCAServiceImplPort" binding="tns:MCAServiceImplPortBinding">
<soap:address location="http://113.53.239.243:8080/MCAService"></soap:address>
</port>
</service>
</definitions>
```


จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

วิสเดิลที่ 6.3


```
<?xml version="1.0" encoding="UTF-8"?><!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2.8 svn-revision#13980. --><!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2.8 svn-revision#13980. --><definitions xmlns:tns="http://mca.locus.services"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://mca.locus.services">
<types>
<xsd:schema>
<xsd:import namespace="http://mca.locus.services"
schemaLocation="http://113.53.239.243:8080/MCAService?xsd=1"></xsd:import>
</xsd:schema>
</types>
<message name="sendProspectCustomer">
<part name="parameters" element="tns:sendProspectCustomer"></part>
</message>
<message name="sendProspectCustomerResponse">
<part name="parameters" element="tns:sendProspectCustomerResponse"></part>
</message>
<message name="removeProspectCustomer">
<part name="parameters" element="tns:removeProspectCustomer"></part>
</message>
<message name="removeProspectCustomerResponse">
<part name="parameters" element="tns:removeProspectCustomerResponse"></part>
</message>
<message name="sendHospitalAdmitData">
<part name="parameters" element="tns:sendHospitalAdmitData"></part>
</message>
<message name="sendHospitalAdmitDataResponse">
<part name="parameters" element="tns:sendHospitalAdmitDataResponse"></part>
```


Chulalongkorn University

```
</message>
<message name="updateHospitalAdmitData">
<part name="parameters" element="tns:updateHospitalAdmitData"></part>
</message>
<message name="updateHospitalAdmitDataResponse">
<part name="parameters" element="tns:updateHospitalAdmitDataResponse"></part>
</message>
<message name="rejectApplication">
<part name="parameters" element="tns:rejectApplication"></part>
</message>
<message name="rejectApplicationResponse">
<part name="parameters" element="tns:rejectApplicationResponse"></part>
</message>
<portType name="MCAService">
<operation name="sendProspectCustomer">
<input xmlns:ns1="http://www.w3.org/2007/05/addressing/metadata"
ns1:Action="http://mca.locus.services/MCAService/sendProspectCustomerRequest"
message="tns:sendProspectCustomer"></input>
<output xmlns:ns2="http://www.w3.org/2007/05/addressing/metadata"
ns2:Action="http://mca.locus.services/MCAService/sendProspectCustomerResponse"
message="tns:sendProspectCustomerResponse"></output>
</operation>
<operation name="removeProspectCustomer">
<input xmlns:ns3="http://www.w3.org/2007/05/addressing/metadata"
ns3:Action="http://mca.locus.services/MCAService/removeProspectCustomerRequest"
message="tns:removeProspectCustomer"></input>
<output xmlns:ns4="http://www.w3.org/2007/05/addressing/metadata"
ns4:Action="http://mca.locus.services/MCAService/removeProspectCustomerRespons
e" message="tns:removeProspectCustomerResponse"></output>
</operation>
<operation name="sendHospitalAdmitData">
```

```
<input xmlns:ns5="http://www.w3.org/2007/05/addressing/metadata"
ns5:Action="http://mca.locus.services/MCAService/sendHospitalAdmitDataRequest"
message="tns:sendHospitalAdmitData"></input>
<output xmlns:ns6="http://www.w3.org/2007/05/addressing/metadata"
ns6:Action="http://mca.locus.services/MCAService/sendHospitalAdmitDataResponse"
message="tns:sendHospitalAdmitDataResponse"></output>
</operation>
<operation name="updateHospitalAdmitData">
<input xmlns:ns7="http://www.w3.org/2007/05/addressing/metadata"
ns7:Action="http://mca.locus.services/MCAService/updateHospitalAdmitDataRequest"
message="tns:updateHospitalAdmitData"></input>
<output xmlns:ns8="http://www.w3.org/2007/05/addressing/metadata"
ns8:Action="http://mca.locus.services/MCAService/updateHospitalAdmitDataResponse
" message="tns:updateHospitalAdmitDataResponse"></output>
</operation>
<operation name="rejectApplication">
<input xmlns:ns9="http://www.w3.org/2007/05/addressing/metadata"
ns9:Action="http://mca.locus.services/MCAService/rejectApplicationRequest"
message="tns:rejectApplication"></input>
<output xmlns:ns10="http://www.w3.org/2007/05/addressing/metadata"
ns10:Action="http://mca.locus.services/MCAService/rejectApplicationResponse"
message="tns:rejectApplicationResponse"></output>
</operation>
</portType>
</definitions>
```

ประวัติผู้เขียนวิทยานิพนธ์

นายอัคคนัฐ จอมจุมพล เกิดเมื่อวันที่ 12 สิงหาคม พ.ศ. 2526 ณ จังหวัดมุกดาหาร วุฒิการศึกษาระดับปริญญาตรี หลักสูตรวิศวกรรมศาสตรบัณฑิต (วิศวกรรมคอมพิวเตอร์) ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น สำเร็จการศึกษาในปี พ.ศ. 2549 และเข้าศึกษาในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปี พ.ศ. 2556

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University