# การปรับปรุงการประเมินความคล้ายคลึงเชิงโครงสร้างและเชิงความหมายเพื่อการค้นคืนเว็บเซอร์วิซ

นายนิติพันธ์ ป้อมพันธ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2556
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย
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 Graduate School.

# IMPROVING STRUCTURAL AND SEMANTIC SIMILARITY EVALUATION FOR WEB SERVICE RETRIEVAL

Mr. Nitipan Pompan

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2013

Copyright of Chulalongkorn University

| | |
|---|---|
| หัวข้อวิทยานิพนธ์ | การปรับปรุงการประเมินความคล้ายคลึงเชิงโครงสร้างและเชิงความหมายเพื่อการค้นคืนเว็บเซอร์วิซ |
| โดย | นายนิติพันธ์ ป้อมพันธ์ |
| สาขาวิชา | วิทยาศาสตร์คอมพิวเตอร์ |
| อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก | รองศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา |

---

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.บัณฑิต เอื้ออาภรณ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(อาจารย์ ดร.ยรรยง เต็งอำนวย)

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

..... กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.กฤดาภัทร สีหารี)

นิติพันธ์ ป้อมพันธ์ : การปรับปรุงการประเมินความคล้ายคลึงเชิงโครงสร้างและเชิงความหมายเพื่อการค้นคืนเว็บเซอร์วิซ. (IMPROVING STRUCTURAL AND SEMANTIC SIMILARITY EVALUATION FOR WEB SERVICE RETRIEVAL) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : รศ. ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา, 67 หน้า

เว็บเซอร์วิซได้ถูกใช้งานอย่างแพร่หลายในแอพพลิเคชันซอฟต์แวร์ที่ทันสมัยเนื่องจากการเป็นหน่วยของซอฟต์แวร์บนเครือข่ายที่สามารถให้บริการฟังก์ชันงานบางอย่างและถูกเรียกใช้ในการสร้างแอพพลิเคชันซอฟต์แวร์ได้อย่างยืดหยุ่น เช่นเดียวกับซอฟต์แวร์อื่น ๆ เว็บเซอร์วิซอาจได้รับการเปลี่ยนแปลงหรือเกิดความล้มเหลวในการทำงาน จนทำให้ไม่สามารถเข้าถึงได้โดยแอพพลิเคชันของผู้ใช้บริการ ในกรณีนี้แอพพลิเคชันของผู้ใช้บริการจึงจำเป็นต้องหาบริการทางเลือกอื่น ๆ หนึ่งในวิธีที่มีประสิทธิผลคือการประเมินความคล้ายคลึงกันทั้งในเชิงโครงสร้างและเชิงความหมายระหว่างคำอธิบายการให้บริการของเว็บเซอร์วิซที่ใช้อยู่กับของผู้ให้บริการเว็บเซอร์วิซอื่น ๆ เพื่อที่จะค้นหาบริการทางเลือก วิทยานิพนธ์นี้ใช้แนวทางตามวิธีการที่เรียกว่า ยูอาร์บีอี เพื่อพิจารณาความคล้ายคลึงกันทั้งในเชิงโครงสร้างและเชิงความหมายของเว็บเซอร์วิซ โดยเพิ่มการประเมินความคล้ายคลึงกันของชนิดข้อมูลโดยพิจารณาในกลุ่มของชนิดข้อมูลและหลักการโคแวเรียนซ์/คอนทราแวเรียนซ์ร่วมด้วย อีกทั้งยังพิจารณาความคล้ายคลึงกันของชื่อโดยพิจารณาความคล้ายคลึงกันของรูปคำ อัลกอริทึมที่เพิ่มความสามารถในการประเมินความคล้ายคลึงนี้เรียกว่า เอ็ม-ยูอาร์บีอี ผลจากการทดลองแสดงให้เห็นว่าเมื่อเปรียบเทียบกับยูอาร์บีอีแล้ว เอ็ม-ยูอาร์บีอีสามารถปรับปรุงประสิทธิภาพของการค้นคืนเว็บเซอร์วิซได้ นอกจากนี้ยังได้มีการพัฒนาระบบค้นคืนเว็บเซอร์วิซซึ่งรองรับอัลกอริทึมเอ็ม-ยูอาร์บีอี และสามารถวิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิซที่ใช้งานอยู่กับเว็บเซอร์วิซที่สามารถทำงานทดแทนได้ ทั้งนี้เพื่อให้ผู้ใช้บริการสามารถเตรียมการรับมือกับความแตกต่างก่อนที่จะใช้งานเว็บเซอร์วิซตัวแทน

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่อ อ. ที่ปรึกษาวิทยานิพนธ์หลัก .....

ปีการศึกษา 2556

# # 5470958521 : MAJOR COMPUTER SCIENCE
KEYWORD : WEB SERVICES / RETRIEVAL / WSDL / ONTOLOGY / SUBSTITUTABILITY

NITIPAN POMPAN : IMPROVING STRUCTURAL AND SEMANTIC SIMILARITY EVALUATION FOR WEB SERVICE RETRIEVAL. ADVISOR : ASSOC. PROF. TWITTIE SENIVONGSE, Ph.D., 67 pp.

Web services have been used widely in modern software applications since they, as networked software units, provide certain functionality that can be incorporated into building software applications in a flexible manner. Like other software, Web services may experience changes and failures which make them inaccessible to service consuming applications. In this case, it is then necessary for those applications to find other alternative services. One of the effective approaches is to evaluate both structural similarity and semantic similarity between the description of the service in use and those of other candidate services in order to identify an alternative. This thesis follows an approach called URBE to determine structural and semantic similarity between Web services. In particular, we enhance the evaluation on data type similarity, by also considering family of data types and covariance/contravariance principle, and on name similarity, by also considering text similarity. The enhanced algorithm is called M-URBE. An experiment shows that, in comparison with URBE, M-URBE can improve the performance of Web service retrieval. In addition, a web service retrieval system is developed. It supports the M-URBE algorithm and analyzes the difference between the service is use and potential substitute services so that the service consumer can prepare for the difference before using any substitute service.

Department: Computer Engineering Student's Signature .....
Field of Study: Computer Science Advisor's Signature.....
Academic Year: 2013

# กิตติกรรมประกาศ

ขอขอบพระคุณรองศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่สละเวลาคอยให้คำปรึกษา คำแนะนำ ข้อคิด และความช่วยเหลือต่าง ๆ ด้วยความเมตตา อันมีค่าอย่างยิ่ง ตลอดระยะเวลาการศึกษาและการวิจัย ทำให้วิทยานิพนธ์นี้สำเร็จลุล่วงได้ด้วยดี

ขอขอบพระคุณอาจารย์ ดร.ยรรยง เต็งอำนวย ประธานคณะกรมการสอบวิทยานิพนธ์ และผู้ช่วยศาสตราจารย์ ดร.กฤดาภัทร สีหารี กรรมการสอบวิทยานิพนธ์ที่ให้ข้อชี้แนะในการปรับปรุงงานวิทยานิพนธ์ให้มีคุณภาพยิ่งขึ้น

ขอขอบพระคุณคณาจารย์ทุกท่านในภาควิชาวิศวกรรมคอมพิวเตอร์ที่ได้อบรม สั่งสอน ให้ความรู้ต่าง ๆ ซึ่งเป็นประโยชน์ต่อการทำวิจัยและการทำงานในอนาคต

ท้ายที่สุดนี้ขอขอบพระคุณ คุณพ่อ คุณแม่ และครอบครัวที่เป็นกำลังใจและแรงสนับสนุนสำคัญในด้านการศึกษาจนทำให้ประสบความสำเร็จมาได้ถึงทุกวันนี้

# สารบัญ

หน้า

บทคัดย่อภาษาไทย.....ง

บทคัดย่อภาษาอังกฤษ.....จ

กิตติกรรมประกาศ.....ฉ

สารบัญ .....ช

สารบัญตาราง.....ฎ

สารบัญภาพ .....ฎ

บทที่

1 บทนำ ..... 1

1.1 ความเป็นมาและความสำคัญของปัญหา .....1

1.2 วัตถุประสงค์ของการวิจัย .....2

1.3 ขอบเขตของการวิจัย .....2

1.4 ประโยชน์ที่จะได้รับ.....3

1.5 ขั้นตอนและวิธีการดำเนินงานวิจัย.....3

1.6 ผลงานตีพิมพ์ .....4

2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง ..... 5

2.1 วิสเดิล .....5

2.2 ออนโทโลจี .....5

2.3 เอสเอวิสเดิล.....6

2.4 ความเข้ากันได้แบบโคแวเรียนซ์/คอนทราแวเรียนซ์.....7

2.5 งานวิจัยที่เกี่ยวข้อง.....8

บทที่ หน้า

2.5.1 URBE: Web Service Retrieval Based on Similarity Evaluation ....8
2.5.2 Structural and Semantic Matching for Assessing Web-Service Similarity....8
2.5.3 Toward High-Precision Service Retrieval....9
2.5.4 A Specialized Search Engine for Web Service Discovery....9
2.5.5 Clustering WSDL Documents to Bootstrap the Discovery of Web Services .... 10
2.5.6 Measuring Similarity of Web Services Based on WSDL .... 10
2.5.7 On The Evolution of Services .... 10
2.5.8 End-to-End Versioning Support for Web Services.... 11
2.5.9 การเปรียบเทียบกับงานวิจัยที่เกี่ยวข้อง .... 11
3 การเปรียบเทียบความคล้ายคลึงกันของเว็บเซอร์วิซ .... 13
3.1 การเปรียบเทียบความคล้ายคลึงกันของเว็บเซอร์วิซในเชิงโครงสร้าง.... 17
3.1.1 ฟังก์ชันการให้คะแนนที่สูงที่สุด (maxSim).... 17
3.1.2 ฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันของชื่อ (nameSim) .... 18
3.1.3 ฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันของชนิดข้อมูล (datatypeSim) .... 19
3.1.4 ฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันของพารามิเตอร์ (parSim).... 22
3.1.5 ฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันของโอเปอเรชัน (opSim).... 22

บทที่ หน้า

3.1.6 ฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันของเว็บเซอร์วิซ (fSim) ..... 22
3.2 การเปรียบเทียบความคล้ายคลึงกันของเว็บเซอร์วิซในเชิงความหมาย ..... 23
3.2.1 ฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันระหว่างคลาส - คลาส และ คุณสมบัติ - คุณสมบัติ (pathSim) ..... 23
3.2.2 ฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันระหว่างคลาส – คุณสมบัติ (classPropSim) ..... 25
3.2.3 ฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันระหว่าง คุณสมบัติ – คลาส (propClassSim) ..... 25
3.3 สรุปการปรับปรุงโดยงานของผู้วิจัย ..... 26
4 ระบบค้นคืนและวิเคราะห์ความแตกต่างสำหรับเว็บเซอร์วิซที่มีความคล้ายคลึงกัน ..... 27
4.1 ภาพรวมของระบบ ..... 27
4.2 เว็บเซอร์วิซเบราว์เซอร์ ..... 28
4.3 เอพีไอด้านความคล้ายคลึงกันของเว็บเซอร์วิซ ..... 31
4.3.1 เอพีไอสำหรับการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกัน ..... 32
4.3.2 เอพีไอสำหรับเรียกดูผลการวิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิซ ..... 33
4.4 เครื่องประมวลความคล้ายคลึง ..... 35
4.4.1 Abstract Layer Parser ..... 37
4.4.2 WSDL Similarity ..... 38
4.4.3 WSDL Difference Analysis ..... 40
4.5 เครื่องมือที่ใช้ในการพัฒนา ..... 42

บทที่ หน้า

5 การทดสอบความคล้ายคลึงและวิเคราะห์ความแตกต่างของเว็บเซอร์วิซ ..... 43
5.1 การวัดประสิทธิภาพการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกัน ..... 43
5.1.1 การวัดประสิทธิภาพการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกัน เชิงโครงสร้าง ..... 44
5.1.2 การวัดประสิทธิภาพการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกัน เชิงความหมาย ..... 45
5.1.3 สรุปผลการทดสอบ ..... 46
5.2 การทดสอบการค้นคืนเว็บเซอร์วิซโดยกรณีต่าง ๆ ..... 47
5.2.1 การทดสอบการค้นคืนเว็บเซอร์วิซโดยพิจารณาเชิงโครงสร้าง ..... 47
5.2.2 การทดสอบการค้นคืนเว็บเซอร์วิซโดยพิจารณาเชิงความหมาย ..... 53
5.2.3 สรุปผลการทดสอบ ..... 59
6 บทสรุป ..... 60
6.1 สรุปผลการวิจัย ..... 60
6.2 ปัญหาและข้อจำกัดที่พบจากการวิจัย ..... 61
6.3 ข้อเสนอแนะ ..... 62
รายการอ้างอิง ..... 63
ประวัติผู้เขียนวิทยานิพนธ์ ..... 67

# สารบัญตาราง

ตารางที่ หน้า

2.1 การเปรียบเทียบระบบการค้นคืนเว็บเซอร์วิซ ..... 12

3.1 กฎการแยกคำตามรูปแบบการตั้งชื่อของเอลิเมนต์ [8] ..... 18

3.2 การให้คะแนนการเข้าคู่ชนิดข้อมูลแบบ Simple Type ตามกลุ่มชนิดข้อมูล (ปรับจาก [8]) ..... 20

3.3 การให้คะแนนแบบ GNA สำหรับกลุ่มข้อมูลที่เป็นชนิด Integer [24] ..... 21

3.4 การให้คะแนนแบบ GNA สำหรับกลุ่มข้อมูลที่เป็นชนิด Real [24] ..... 21

3.5 สรุปการปรับปรุงโดยงานวิจัยของผู้วิจัย ..... 26

4.1 ผลการวิเคราะห์ในแต่ละส่วนประกอบของ WSDL Difference Analysis ..... 41

# สารบัญภาพ

ภาพที่ หน้า

2.1 ความสัมพันธ์ของวัตถุภายในออนโทโลจี Vehicle ..... 6

2.2 การอธิบายเพิ่มเติมเชิงความหมายโดยอ้างอิงไปยังแบบจำลองออนโทโลจี [16] ..... 7

2.3 แนวทางการค้นคืนเว็บเซอร์วิซซึ่งให้ความแม่นยำและจำนวนที่ค้นคืนได้ที่แตกต่างกันในแต่ละเทคนิค [17] ..... 9

3.1 การให้คะแนนความคล้ายคลึงกันของเว็บเซอร์วิซ [8] ..... 13

3.2 ตัวอย่างเว็บเซอร์วิซ ..... 16

3.3 ตัวอย่างเว็บเซอร์วิซที่ถูกแปลงให้อยู่ในรูปแบบเชิงนามธรรม ..... 16

3.4 การเข้าคู่ในทุก ๆ คู่ที่เป็นไปได้ระหว่างเอลิเมนต์ต้นทางไปยังเอลิเมนต์ปลายทาง ปรับจาก [8] ..... 17

3.5 ลำดับชั้นของการสืบทอดชนิดตัวแปรในกลุ่มของ Integer และ Real ..... 20

3.6 อัลกอริทึมฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันของชนิดข้อมูล (*datatypeSim*) ..... 21

4.1 ระบบค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันและรายงานผลการแทนที่กับผู้ใช้งาน (ปรับจาก [8]) ..... 28

4.2 เว็บเซอร์วิซเบราว์เซอร์ ..... 29

4.3 รายละเอียดของเว็บเซอร์วิซปลายทาง ..... 30

4.4 โครงสร้างของเอพีไอด้านความคล้ายคลึงกันของเว็บเซอร์วิซ ..... 31

4.5 วิธีการเรียกใช้งานเอพีไอสำหรับการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกัน ..... 32

4.6 วิธีการเรียกใช้งานเอพีไอสำหรับเรียกดูผลการวิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิซ ..... 33

4.7 กลุ่มของเครื่องประมวลความคล้ายคลึง Engine 1 – 4 ..... 35

4.8 ส่วนประกอบต่าง ๆ ภายในเครื่องประมวลความคล้ายคลึง ..... 35

ภาพที่ หน้า

4.9 กระบวนการการให้คะแนนความคล้ายคลึงกันของเว็บเซอร์วิซและการรายงานความแตกต่างระหว่างเว็บเซอร์วิซ ..... 36

4.10 Abstract Layer Parser ..... 38

4.11 WSDL Similarity ..... 38

4.12 WSDL Difference Analysis ..... 40

5.1 เว็บเซอร์วิซที่เกี่ยวข้องในโดเมนเดียวกัน ..... 44

5.2 ผลการวัดประสิทธิภาพการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันเชิงโครงสร้าง ..... 45

5.3 ผลการวัดประสิทธิภาพการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันเชิงความหมาย ..... 46

5.4 ตัวอย่างเว็บเซอร์วิซต้นทางที่ใช้ในการการทดสอบในเชิงโครงสร้าง ..... 48

5.5 ตัวอย่างเว็บเซอร์วิซปลายทาง 1 ..... 49

5.6 ตัวอย่างเว็บเซอร์วิซปลายทาง 2 ..... 50

5.7 ผลการค้นคืนเว็บเซอร์วิซด้วยความคล้ายคลึงเชิงโครงสร้างโดยใช้อัลกอริทึมยูอาร์บีอี ..... 51

5.8 ผลการค้นคืนเว็บเซอร์วิซด้วยความคล้ายคลึงเชิงโครงสร้างโดยใช้อัลกอริทึมเอ็ม-ยูอาร์บีอี ..... 51

5.9 รายละเอียดของเว็บเซอร์วิซ PortType ContractInquiryService2 จากการค้นคืนด้วยการพิจารณาเชิงโครงสร้าง ..... 52

5.10 ตัวอย่างออนโทโลจีสำหรับการทดสอบการค้นคืนเว็บเซอร์วิซโดยพิจารณาเชิงความหมาย ..... 53

5.11 ตัวอย่างเว็บเซอร์วิซต้นทางที่ทำการกำกับความหมายด้วยเทอมในออนโทโลจีตัวอย่าง ..... 54

5.12 ตัวอย่างเว็บเซอร์วิซปลายทางที่กำกับความหมายด้วยเทอมในออนโทโลจี 1 ..... 55

ภาพที่ หน้า

5.13 ตัวอย่างเว็บเซอร์วิซปลายทางที่กำกับความหมายด้วยเทอมในออนโทโลจี 2 ..... 56

5.14 ผลการค้นคืนเว็บเซอร์วิซด้วยความคล้ายคลึงเชิงความหมายโดยใช้อัลกอริทึมยู อาร์บีอี ..... 57

5.15 ผลการค้นคืนเว็บเซอร์วิซด้วยความคล้ายคลึงเชิงความหมายโดยใช้ อัลกอริทึมเอ็ม-ยูอาร์บีอี ..... 57

5.16 รายละเอียดของเว็บเซอร์วิซ PortType ContractInquiryService2 จากการ ค้นคืนด้วยการพิจารณาเชิงความหมาย ..... 59

# บทที่ 1

# บทนำ

## 1.1 ความเป็นมาและความสำคัญของปัญหา

เว็บเซอร์วิซ (Web Service) คือระบบซอฟต์แวร์ที่ถูกออกแบบมาเพื่อสนับสนุนการดำเนินงานร่วมกันระหว่างเครื่องคอมพิวเตอร์กับเครื่องคอมพิวเตอร์ โดยติดต่อกันผ่านเครือข่ายผ่านโพรโทคอลเอชทีทีพี (HTTP: Hypertext Transfer Protocol) ซึ่งเครื่องคอมพิวเตอร์ที่ติดต่อกันผ่านเว็บเซอร์วิซทำได้โดยเชื่อมต่อผ่านส่วนต่อประสาน (Interface) และส่วนต่อประสานนี้ถูกอธิบายในรูปแบบของเอกสารวิสเดิล (WSDL: Web Services Description Language) [1], [2] การติดต่อทำโดยการส่งข้อความ (Message) ที่อยู่ในรูปแบบของโซป (SOAP: Simple Object Access Protocol) [3] ซึ่งอยู่ในรูปแบบของเอกสารเอกซ์เอ็มแอล (XML: Extensible Markup Language) [4] ทำให้เว็บเซอร์วิซไม่ขึ้นต่อระบบปฏิบัติการและภาษาโปรแกรมใด ๆ

ตามสถาปัตยกรรมเอสโอเอ เมื่อผู้ใช้งานต้องการใช้บริการเว็บเซอร์วิซ ผู้ใช้จะทำการค้นหาเว็บเซอร์วิซที่สามารถให้บริการได้ในยูดีดีไอ (UDDI: Universal Description, Discovery and Integration) [5] ซึ่งผู้ให้บริการเว็บเซอร์วิซได้เข้ามาประกาศการบริการไว้ก่อนหน้านั้นแล้ว ในการใช้งานเว็บเซอร์วิซอาจเกิดปัญหา เช่น เว็บเซอร์วิซเกิดการขัดข้อง เว็บเซอร์วิซเกิดการเปลี่ยนแปลงขั้นตอนการเข้าถึงโดยไม่ได้แจ้งมาทางช่องทางการประกาศการบริการเพื่อให้ผู้ใช้งานทราบ และเว็บเซอร์วิซยกเลิกการให้บริการ ดังนั้นจึงจำเป็นต้องค้นหาเว็บเซอร์วิซที่คล้ายคลึงกันเพื่อมาใช้งานแทน ได้แก่ (1) การค้นหาโดยใช้คำหลัก (Keyword) ในการหาเว็บเซอร์วิซที่เกี่ยวข้อง เช่น งานวิจัยของ Hatzi และคณะ [6] และ งานวิจัยของ Elgazzar และคณะ [7] (2) การค้นหาโดยใช้การพิจารณาจากความคล้ายคลึงกันในเชิงโครงสร้างของเว็บเซอร์วิซ เช่น งานวิจัยของ Plebani และ Pernici [8] งานวิจัยของ Stroulia และ Wang [9] และ งานวิจัยของ Liu และคณะ [10] และ (3) การค้นหาโดยใช้การพิจารณาจากความคล้ายคลึงกันในเชิงความหมายตามเทอมของออนโทโลจีของเว็บเซอร์วิซ เช่น งานวิจัยของ Plebani และ Pernici [8] และงานวิจัยของ Stroulia และ Wang [9] ผู้วิจัยเล็งเห็นว่าในการค้นคืนเว็บเซอร์วิซ นอกจากจะสามารถค้นคืนเว็บเซอร์วิซตามความต้องการของผู้ใช้งานได้แล้ว ควรที่จะสามารถบอกได้ว่าเว็บเซอร์วิซที่ค้นคืนได้นั้นหากนำไปใช้งาน ในบางกรณีอาจไม่สามารถใช้งานแทนที่กันได้โดยทันที จึงควรที่จะมีระบบที่วิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิซเดิมกับเว็บเซอร์วิซที่คล้ายคลึง และแจ้งให้แก่ผู้ใช้บริการเข้าใจถึงความแตกต่างหรือการดำเนินการที่ต้องทำเพิ่มเติมก่อน หากมีความต้องการที่จะใช้งานเว็บเซอร์วิซที่คล้ายคลึงนี้แทนที่เว็บเซอร์วิซเดิม

ผู้วิจัยพบว่างานวิจัยของ Plebani และ Pernici [8] ครอบคลุมการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันทั้งในเชิงโครงสร้างและเชิงความหมาย จึงมีความน่าสนใจที่จะนำมาพัฒนาเพิ่มเติม แต่ยังพบข้อจำกัดในงานดังกล่าวคือ (1) การพิจารณาการให้คะแนนความคล้ายคลึงในเชิงโครงสร้างในส่วนของการพิจารณาชนิดข้อมูลของพารามิเตอร์อินพุตและเอาต์พุต ซึ่งการเปรียบเทียบชนิดข้อมูลนั้น ทำในระดับกลุ่มโดยไม่สนใจความแตกต่างของชนิดข้อมูล ทำให้การพิจารณาความคล้ายคลึงของพารามิเตอร์ที่อยู่ในกลุ่มชนิดเดียวกัน เช่น กลุ่มชนิดตัวเลขจำนวนเต็ม (Integer) ด้วยกัน (คือ long, int, short, byte) หรืออยู่ในกลุ่มชนิดตัวเลขทศนิยม (Real) ด้วยกัน (คือ decimal, double, float) จะได้คะแนนความคล้ายคลึงเท่ากัน (2) การพิจารณาความคล้ายคลึงของชื่อไม่ได้พิจารณาในส่วนของความคล้ายคลึงของรูปคำ และ (3) การเปรียบเทียบพารามิเตอร์ทั้งอินพุตและเอาต์พุตในเชิงโครงสร้างและเชิงความหมาย ใช้การเปรียบเทียบความคล้ายคลึงกันเท่านั้นโดยไม่ได้พิจารณาหลักการความเข้ากันได้ (Compatibility) แบบ Contravariant Input/Covariant Output [11] ซึ่งหมายถึงการที่เว็บเซอร์วิซที่จะใช้เป็นตัวแทนสามารถมีอินพุตที่มีลักษณะทั่วไปกว่าและเอาต์พุตที่มีลักษณะที่เฉพาะเจาะจงกว่าอินพุตและเอาต์พุตของเว็บเซอร์วิซเดิม ข้อจำกัดดังกล่าวนี้จึงเป็นแนวทางในการปรับปรุงการประเมินความคล้ายคลึงเชิงโครงสร้างและเชิงความหมายที่จะเป็นส่วนหนึ่งของการพัฒนาระบบการค้นคืนเว็บเซอร์วิซในงานวิจัยนี้ นอกจากนี้ระบบที่จะพัฒนาจะสามารถวิเคราะห์ความแตกต่างของเว็บเซอร์วิซตัวแทน สำหรับให้ผู้ใช้บริการเตรียมการในการใช้งานเว็บเซอร์วิซตัวแทนต่อไป

## 1.2 วัตถุประสงค์ของการวิจัย

เพื่อพัฒนาระบบค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกับเว็บเซอร์วิซที่ใช้งานอยู่ ทั้งในเชิงโครงสร้างและเชิงความหมาย และมีการวิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิซ

## 1.3 ขอบเขตของการวิจัย

1.3.1 ใช้แนวทางการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันตามงานวิจัย [8] เป็นหลัก

1.3.2 ปรับปรุงการประเมินความคล้ายคลึงเชิงโครงสร้างในส่วนของการให้คะแนนการเข้าคู่ของชนิดข้อมูลของพารามิเตอร์ซึ่งอยู่ในกลุ่มเดียวกันในฟังก์ชัน *simpleDTSim* และพิจารณาจากความเข้ากันได้ในแบบ Contravariant Input/Covariant Output ในฟังก์ชัน *parSim*

1.3.3 ปรับปรุงการประเมินความคล้ายคลึงเชิงโครงสร้างในส่วนของการพิจารณาการให้คะแนนความคล้ายคลึงกันของชื่อได้แก่ ชื่อของโอเปอเรชัน ชื่อของพารามิเตอร์อินพุตและพารามิเตอร์เอาต์พุต และชื่อของชนิดข้อมูลที่เป็นแบบ Complex Type โดยพิจารณาจากความหมายของคำร่วมกับอัลกอริทึม Levenshtein Distance ในการให้คะแนนความคล้ายคลึงกันของชื่อจากรูปคำในฟังก์ชัน *nameSim*

1.3.4 ปรับปรุงการประเมินความคล้ายคลึงเชิงความหมาย ในส่วนของการพิจารณาการให้คะแนนจากความเข้ากันได้ ในแบบ Contravariant Input/Covariant Output ซึ่งถูกกำกับความหมายด้วยเทอมในออนโทโลจีในฟังก์ชัน *pathSim*

1.3.5 วิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิซที่ผู้ใช้บริการใช้งานอยู่กับเว็บเซอร์วิซที่คล้ายคลึงที่ค้นคืนได้ ตามประเภทความแตกต่างในเชิงโครงสร้างและความหมาย

1.3.6 พัฒนา เอพีไอ สำหรับเว็บเซอร์วิซเบราว์เซอร์เพื่อการค้นคืนเว็บเซอร์วิซที่คล้ายคลึงกันด้วยภาษา C# และ ASP.Net บน Microsoft .Net Framework เวอร์ชัน 4.0 โดย เอพีไอ มี 2 ฟังก์ชันให้เรียกใช้คือ (1) ฟังก์ชันสำหรับการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกัน และ (2) ฟังก์ชันสำหรับเรียกดูผลการวิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิซที่ผู้ใช้บริการใช้งานอยู่กับเว็บเซอร์วิซที่คล้ายคลึงที่ค้นคืนได้

## 1.4 ประโยชน์ที่จะได้รับ

ได้ระบบค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงและสามารถใช้งานแทนที่เว็บเซอร์วิซที่ผู้ใช้บริการใช้งานอยู่ได้ โดยระบบมีการปรับปรุงวิธีพิจารณาความคล้ายคลึงในเชิงโครงสร้างและเชิงความหมาย และสามารถวิเคราะห์ข้อมูลความแตกต่างของเว็บเซอร์วิซที่คล้ายคลึง เพื่อให้ผู้ใช้บริการสามารถพิจารณาถึงผลกระทบและเตรียมการในการเลือกใช้เว็บเซอร์วิซที่คล้ายคลึงนั้นเป็นเว็บเซอร์วิซตัวแทนได้ เช่น ทำการแก้ไขระบบของผู้ใช้เองหากมีความจำเป็น

## 1.5 ขั้นตอนและวิธีการดำเนินงานวิจัย

1.5.1 ศึกษาเทคโนโลยีเว็บเซอร์วิซ

1.5.2 สำรวจและศึกษาแนวคิดในการพิจารณาหาเว็บเซอร์วิซที่สามารถเข้ากันได้และการให้คะแนนการเข้าคู่ของเว็บเซอร์วิซจากงานวิจัยอื่น

1.5.3 สำรวจและศึกษาการกำกับความหมายให้กับเว็บเซอร์วิซรวมถึงกฎที่ใช้ในการพิจารณาสภาพเข้ากันได้และการให้คะแนนการเข้าคู่ของเว็บเซอร์วิซจากงานวิจัยอื่น

1.5.4 ศึกษาข้อกำหนดของเอกสารวิสเดิลและข้อกำหนดของเอสเอวิสเดิล

1.5.5 วิเคราะห์ข้อจำกัดจากการค้นคืนเว็บเซอร์วิซที่สามารถเข้ากันได้ของงานวิจัยที่ได้สำรวจ

1.5.6 ออกแบบและปรับปรุงการให้คะแนนการเข้าคู่ของเว็บเซอร์วิซในเชิงโครงสร้างและเชิงความหมาย

1.5.7 พัฒนาระบบค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึง

1.5.8 พัฒนาระบบวิเคราะห์ความแตกต่างของเว็บเซอร์วิซที่คล้ายคลึง

1.5.9 ประเมินผลการวิจัย

1.5.10 สรุปผลการวิจัย จัดทำบทความวิชาการ และจัดทำรายงานวิทยานิพนธ์

## 1.6 ผลงานตีพิมพ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้ตีพิมพ์และนำเสนอในการประชุมวิชาการดังนี้

1.6.1 บทความชื่อ “Enhancing Structural and Semantic Similarity Evaluation for Web Service Retrieval”

1.6.1.1 ชื่อผู้แต่ง Nitipan Pompan และ Twittie Senivongse [12]

1.6.1.2 ตีพิมพ์และนำเสนอในงานประชุมวิชาการชื่อ International Conference on Computing, Engineering and Communication Technologies (ICCECT 2013) ซึ่งจัดขึ้นในวันที่ 16-17 สิงหาคม 2556 ณ กรุงเทพมหานคร ประเทศไทย

# บทที่ 2

# ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

## 2.1 วิสเดิล

วิสเดิล (WSDL: Web Services Description Language) [1], [2] คือเอกสารที่ใช้อธิบายรายละเอียดเกี่ยวกับเว็บเซอร์วิซ เพื่อให้เครื่องคอมพิวเตอร์ทำการติดต่อใช้บริการเว็บเซอร์วิซได้ เอกสารวิสเดิลนี้ถูกอธิบายในรูปแบบเอกสารเอกซ์เอ็มแอล โดยแบ่งการอธิบายรายละเอียดออกเป็นสองส่วนหลักคือ รายละเอียดเชิงนามธรรม และ รายละเอียดเชิงรูปธรรม ในแต่ละส่วนของวิสเดิลรุ่น 1.1 ประกอบไปด้วยเอลิเมนต์ (Element) ต่าง ๆ ดังนี้

2.1.1 รายละเอียดเชิงนามธรรม (Abstract Definition)

- Type คือข้อกำหนดชนิดข้อมูลที่เกี่ยวข้องทั้งหมดที่ใช้ติดต่อระหว่างเครื่องคอมพิวเตอร์ที่ร้องขอใช้บริการกับเว็บเซอร์วิซซึ่งอธิบายตามข้อกำหนดของเอกซ์เอ็มแอลสกีมา (XML Schema) [13]
- Message คือข้อกำหนดข้อมูลระหว่างเครื่องคอมพิวเตอร์ที่ร้องขอใช้บริการกับเว็บเซอร์วิซ ในเอกสารวิสเดิลรุ่น 2.0 Message จะถูกรวมเข้าไปใน Type โดยตรงแทน [2]
- Operation คือการดำเนินการในเว็บเซอร์วิซ
- PortType คือกลุ่มของการดำเนินการในเว็บเซอร์วิซ ในเอกสารวิสเดิลรุ่น 2.0 ใช้ชื่อว่า Interface [2]

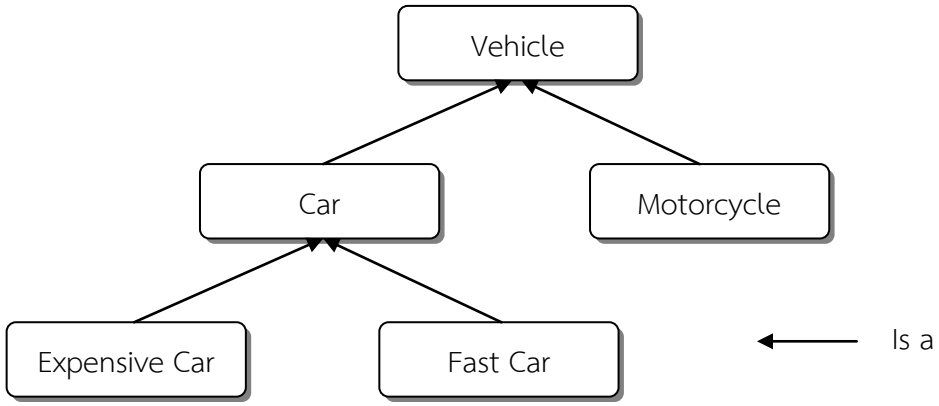
2.1.2 รายละเอียดเชิงรูปธรรม (Concrete Definition)

- Binding คือข้อกำหนดรูปแบบข้อมูลในการเชื่อมต่อ
- Port คือที่อยู่ที่ใช้ติดต่อไปยังเว็บเซอร์วิซอยู่ในรูปแบบของยูอาร์แอล (URL) ในเอกสารวิสเดิลรุ่น 2.0 ใช้ชื่อว่า Endpoint [2]
- Service คือกลุ่มของพอร์ตทั้งหมดที่สามารถติดต่อได้ภายในเอกสารวิสเดิล

## 2.2 ออนโทโลจี

ออนโทโลจี (Ontology) [14] คือการอธิบายขอบเขตองค์ความรู้ที่สนใจให้อยู่ในรูปแบบของแบบจำลอง (Model) ซึ่งในสาขาวิทยาการคอมพิวเตอร์การอธิบายให้อยู่ในแบบจำลองถูกอธิบายด้วยเอกสารเอกซ์เอ็มแอล เช่น ภาษาอาวล์ (OWL) [15] โดยออนโทโลจีจะอธิบายส่วนต่าง ๆ ได้แก่

- อินดิวิดวล (Individuals) แทนวัตถุภายในโดเมนที่สนใจ
- คลาส (Classes) คือ กลุ่มของวัตถุที่ประกอบด้วยอินดิวิดวล เช่น คลาส Vehicle หมายถึง กลุ่มของยานพาหนะที่สนใจ
- แอททริบิวต์ (Attributes) คือ คุณสมบัติที่ใช้อธิบายวัตถุในออนโทโลจี
- ความสัมพันธ์ (Relations) คือ การอธิบายความสัมพันธ์ระหว่างคลาสในออนโทโลจี ดังภาพที่ 2.1 คลาส Fast Car จะเป็นซับคลาส (Subclass) ของคลาสชื่อ Car




ภาพที่ 2.1 ความสัมพันธ์ของวัตถุภายในออนโทโลจี Vehicle

## 2.3 เอสเอวิสเดิล

เอสเอวิสเดิล (SAWSDL: Semantic Annotations for WSDL and XML Schema) [16] เป็นส่วนต่อขยายจากเอกสารวิสเดิลเพื่ออธิบายเพิ่มเติมเชิงความหมาย (Semantic) ให้กับส่วนประกอบต่าง ๆ ภายในเอกสารวิสเดิลรวมถึงเอกซ์เอ็มแอลสกีมา โดยการอธิบายเพิ่มเติมทำได้โดยการกำกับความหมาย (Annotation) เพื่ออ้างอิงกลับไปยังเอกสารออนโทโลจีซึ่งได้อธิบายถึงแบบจำลองไว้อยู่แล้วด้วยยูอาร์ไอ (URI)

การอธิบายเพิ่มเติมเชิงความหมายทำได้โดยการเพิ่ม attribute ที่ชื่อ modelReference ลงไปยังส่วนประกอบต่าง ๆ ของเอกสารวิสเดิล ได้แก่ Interface, Operation และ Type ที่อธิบายในรูปแบบของเอกซ์เอ็มแอลสกีมาเพื่ออ้างอิงไปยังคำศัพท์หรือเทอมในออนโทโลจีซึ่งแสดงถึงความหมายของส่วนประกอบนั้น ๆ ของเอกสารวิสเดิลดังภาพที่ 2.2

```
…
<wsdl:interface name="Order"
sawsdl:modelReference="http://example.org/categorization/products/electronics">
  …
</wsdl:interface>
…
```

```
…
<wsdl:operation name="order" pattern="http://www.w3.org/ns/wsdl/in-out"

sawsdl:modelReference="http://www.w3.org/2002/ws/sawsdl/spec/ontology/purchaseorder#RequestPurchaseOrder">
<wsdl:input element="OrderRequest"/>
<wsdl:output element="OrderResponse"/>
</wsdl:operation>
…
```

```
…
<xs:simpleType name="Confirmation"

sawsdl:modelReference="http://www.w3.org/2002/ws/sawsdl/spec/ontology/purchaseorder#OrderConfirmation">
 …
</xs:simpleType>
…
```

ภาพที่ 2.2 การอธิบายเพิ่มเติมเชิงความหมายโดยอ้างอิงไปยังแบบจำลองออนโทโลจี [16]

## 2.4 ความเข้ากันได้แบบโคแวเรียนซ์/คอนทราแวเรียนซ์

ตามหลักการความเข้ากันได้แบบโคแวเรียนซ์/คอนทราแวเรียนซ์ (Covariance/Contravariance) [11] เมื่อพิจารณาในแง่มุมของชนิดข้อมูลที่มีความสัมพันธ์กันและต้องการใช้งานแทนที่กัน ความเข้ากันได้มีความแตกต่างกันในทิศทางของชนิดข้อมูล หากชนิดข้อมูลที่ต้องการแทนที่เป็นชนิดข้อมูลที่ใช้เป็นพารามิเตอร์อินพุตที่นำไปใช้ในระบบ ชนิดข้อมูลที่ต้องการแทนที่มีความสัมพันธ์กันกับชนิดข้อมูลดั้งเดิมที่มีลักษณะทั่วไปกว่า (Generalized) มีความเข้ากันได้ดีกว่าชนิดข้อมูลที่ต้องการแทนที่มีความสัมพันธ์กันกับชนิดข้อมูลดั้งเดิมที่มีลักษณะเฉพาะเจาะจงกว่า (Specialized) เนื่องจากชนิดข้อมูลที่มีลักษณะเฉพาะเจาะจงกว่าต้องการข้อมูลที่อยู่นอกเหนือจากชนิดข้อมูลดั้งเดิมที่มี ความเข้ากันได้นี้เป็นแบบคอนทราแวเรียนซ์อินพุต ในทางกลับกันชนิดข้อมูลที่เป็นพารามิเตอร์เอาต์พุตที่เป็นผลลัพธ์จากระบบ ชนิดข้อมูลที่ต้องการแทนที่มีความสัมพันธ์กันกับชนิดข้อมูลดั้งเดิมที่มีลักษณะเฉพาะเจาะจงกว่ามีความเข้ากันได้ดีกว่าชนิดข้อมูลที่ต้องการแทนที่มีความสัมพันธ์กันกับชนิดข้อมูลดั้งเดิมที่มีลักษณะทั่วไปกว่า ความเข้ากันได้นี้เป็นแบบโคแวเรียนซ์เอาต์พุต

## 2.5 งานวิจัยที่เกี่ยวข้อง

### 2.5.1 URBE: Web Service Retrieval Based on Similarity Evaluation

งานวิจัยของ Plebani และ Pernici [8] นำเสนออัลกอริทึม (Algorithm) ในการหาเว็บเซอร์วิซที่มีความคล้ายคลึงกันโดยค้นหาจากการเข้าคู่ (Matching) ทั้งเชิงโครงสร้าง (Structure) และเชิงความหมาย (Semantic) โดยทำการค้นหาจากยูดีดีไอ เว็บเซอร์วิซที่ทำการค้นหาได้จะมีคะแนนการเข้าคู่ซึ่งอยู่ในช่วงตั้งแต่ 0 ถึง 1 คะแนนที่ได้บ่งบอกถึงความซับซ้อนในการสร้างตัวกลาง (Mediator) ในการทดแทนการใช้งานเว็บเซอร์วิซเดิมด้วยเว็บเซอร์วิซใหม่ที่ค้นหาได้ หากคะแนนที่ได้น้อยที่สุดคือ 0 หมายถึงมีความซับซ้อนมากในการสร้างตัวกลาง และคะแนนที่สูงที่สุดคือ 1 หมายถึงความซับซ้อนน้อยในการสร้างตัวกลาง

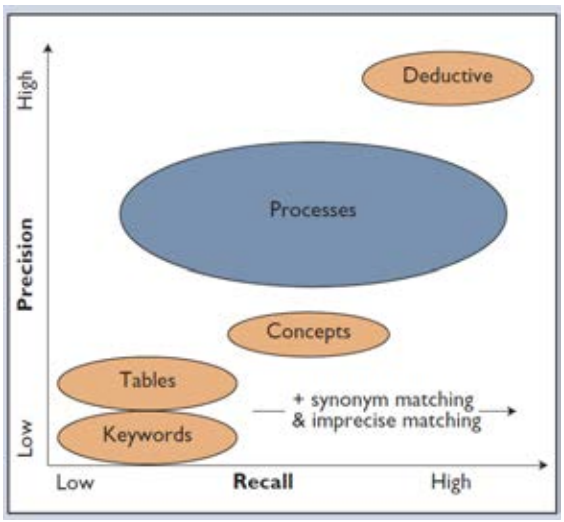
จากที่กล่าวมาแล้ว ผู้วิจัยจะนำอัลกอริทึมของงานวิจัยนี้มาพัฒนาเพิ่มเติม แต่จะทำการปรับปรุงข้อจำกัด อันได้แก่ (1) การพิจารณาการให้คะแนนความคล้ายคลึงในเชิงโครงสร้างในส่วนของการพิจารณาชนิดข้อมูลของพารามิเตอร์อินพุตและเอาต์พุต ซึ่งทำการเปรียบเทียบชนิดข้อมูลในระดับกลุ่ม โดยไม่สนใจความแตกต่างของชนิดข้อมูลที่แท้จริง จึงทำให้การพิจารณาความคล้ายคลึงของพารามิเตอร์ที่อยู่ในกลุ่มชนิดเดียวกัน จะได้คะแนนความคล้ายคลึงเท่ากัน (2) การพิจารณาความคล้ายคลึงของชื่อไม่ได้พิจารณาในส่วนของความคล้ายคลึงของรูปคำ และ (3) การเปรียบเทียบพารามิเตอร์ทั้งอินพุตและเอาต์พุตในเชิงโครงสร้างและเชิงความหมาย ใช้การเปรียบเทียบความคล้ายคลึงกันเท่านั้นโดยไม่ได้พิจารณาหลักการความเข้ากันได้แบบ Contravariant Input/Covariant Output

### 2.5.2 Structural and Semantic Matching for Assessing Web-Service Similarity

งานวิจัยของ Stroulia และ Wang [9] นำเสนอแนวทางการให้คะแนนความเหมือนของเว็บเซอร์วิซโดยการจับคู่เปรียบเทียบในทุก ๆ เอลิเมนต์ ในเชิงโครงสร้างและเชิงความหมาย ซึ่งคะแนนที่ได้ของการจับคู่เปรียบเทียบในทุก ๆ คู่ที่เป็นไปได้จะถูกแสดงในรูปแบบของเมทริกซ์สองมิติ ขนาดที่ได้คือ จำนวนเอลิเมนต์ต้นทางคูณด้วยจำนวนเอลิเมนต์ปลายทาง จากนั้นจะทำการเลือกคะแนนที่สูงที่สุดในแต่ละคะแนนของการจับคู่เปรียบเทียบ คะแนนที่ได้จะเป็นส่วนหนึ่งของการจับคู่เปรียบเทียบที่อยู่ในเมทริกซ์ในลำดับชั้นที่สูงกว่าตามลำดับดังนี้คือ การจับคู่เปรียบเทียบชนิดข้อมูลต้นทางกับชนิดข้อมูลปลายทาง การจับคู่เปรียบเทียบข้อความต้นทางกับข้อความปลายทาง การจับคู่เปรียบเทียบโอเปอเรชันต้นทางกับโอเปอเรชันปลายทาง และการจับคู่เปรียบเทียบเว็บเซอร์วิซต้นทางกับเว็บเซอร์วิซปลายทาง

### 2.5.3 Toward High-Precision Service Retrieval

งานวิจัยของ Klein [17] ได้วิเคราะห์แนวทางการค้นคืนเว็บเซอร์วิซในแต่ละเทคนิคต่าง ๆ โดยแต่ละเทคนิคมีข้อจำกัดที่แตกต่างกันดังภาพที่ 2.3 ได้แก่ การค้นคืนแบบ Keyword-Based เป็นการค้นคืนเว็บเซอร์วิซจากฐานข้อมูลที่จัดเก็บเว็บเซอร์วิซซึ่งจะได้จำนวนที่ค้นคืนและความแม่นยำน้อยที่สุด เนื่องจากไม่ได้พิจารณาถึงความหมายของคำหลักที่ใช้ค้นคืน ส่วนแบบ Table-Based เป็นการค้นคืนที่อิงตามข้อมูลที่มีอยู่ในตารางในฐานข้อมูล ซึ่งข้อมูลดังกล่าวถูกป้อนโดยผู้ให้บริการ เทคนิคนี้พบในยูดีดีไอ โดยมีจำนวนที่ค้นคืนและความแม่นยำเท่ากับเทคนิคการค้นคืนแบบ Keyword-Based จำนวนและความแม่นยำของการค้นคืนถูกทำให้เพิ่มขึ้นด้วยเทคนิค Concept-Based ซึ่งได้เพิ่มเติมการอธิบายด้วยเทอมของออนโทโลจีให้แก่เว็บเซอร์วิซเพื่อเพิ่มจำนวนที่ถูกค้นคืนและความแม่นยำ แต่มีความยากมากขึ้นในการเพิ่มเว็บเซอร์วิซโดยผู้ให้บริการไปยังฐานข้อมูล นอกจากนี้ยังมีการค้นคืนแบบ Deductive ซึ่งเป็นการค้นคืนที่ได้จำนวนการค้นคืนและความแม่นยำสูงที่สุด เนื่องจากผู้ให้บริการเว็บเซอร์วิซได้กำหนดคำอธิบายเพิ่มเติมในเทอมของออนโทโลจีของเว็บเซอร์วิซที่ให้บริการได้แก่ อินพุต เอาต์พุต เงื่อนไขที่ต้องมีก่อนใช้บริการเว็บเซอร์วิซ และผลที่ตามมาหลังจากใช้เว็บเซอร์วิซ




ภาพที่ 2.3 แนวทางการค้นคืนเว็บเซอร์วิซซึ่งให้ความแม่นยำและจำนวนที่ค้นคืนได้ที่แตกต่างกันในแต่ละเทคนิค [17]

### 2.5.4 A Specialized Search Engine for Web Service Discovery

งานวิจัยของ Hatzi และคณะ [6] เป็นการค้นหาเว็บเซอร์วิซโดยใช้คำหลักในการค้นหาเว็บเซอร์วิซที่เกี่ยวข้อง ระบบการค้นหาเว็บเซอร์วิซแยกเป็นสองส่วนหลักได้แก่ ส่วนแรกคือส่วนที่ทำหน้าที่สร้างดัชนีหรือสารบัญ (Index) ของเว็บเซอร์วิซโดยจะทำหน้าที่ค้นหาเอกสารสองชนิด ชนิด

แรกคือเอกสารที่ประกาศวิธีการใช้งานเว็บเซอร์วิซคือเอกสารวิสเดิล และ เอสเอวิสเดิล ชนิดที่สองคือเอกสารที่เกี่ยวกับออนโทโลจีเทอมคืออาวล์-เอส ซึ่งอธิบายเว็บเซอร์วิซในเชิงความหมาย จากนั้นจะเก็บชื่อของเอลิเมนต์ต่าง ๆ ที่อยู่ภายในเอกสารทั้งสองชนิดนี้เพื่อสร้างดัชนีสำหรับค้นหาด้วยคำหลักในส่วนที่สองของระบบ

### 2.5.5 Clustering WSDL Documents to Bootstrap the Discovery of Web Services

งานวิจัยของ Elgazzar และคณะ [7] ได้นำเสนอระบบการค้นหาเว็บเซอร์วิซ โดยระบบค้นหาแยกเป็นสองส่วนหลัก ส่วนแรกคือส่วนที่ทำหน้าที่แยกเอลิเมนต์ต่าง ๆ ที่อยู่ภายในเอกสารวิสเดิลออกเป็น 5 กลุ่ม กลุ่มแรกคือกลุ่มของเนื้อหาของคำศัพท์หรือคำหลักที่อยู่ภายในเอกสารวิสเดิล กลุ่มที่สองคือกลุ่มของชนิดข้อมูล กลุ่มที่สามคือกลุ่มของข้อกำหนดข้อมูล กลุ่มที่สี่คือกลุ่มของการดำเนินการในเว็บเซอร์วิซ และ กลุ่มที่ห้าคือกลุ่มของชื่อของเว็บเซอร์วิซ จากนั้นระบบค้นหาในส่วนที่สองจะค้นหาเว็บเซอร์วิซซึ่งถูกแยกเอลิเมนต์ดังกล่าวด้วยคำหลัก

### 2.5.6 Measuring Similarity of Web Services Based on WSDL

งานวิจัยของ Liu และคณะ [10] เป็นการวัดความคล้ายคลึงกันของเว็บเซอร์วิซโดยได้ใช้งานวิจัย [8] เป็นแนวทางในการวิเคราะห์ความคล้ายคลึงกันของเว็บเซอร์วิซเชิงโครงสร้าง แต่มีการปรับปรุงการให้คะแนนความคล้ายคลึงกันของชื่อโดยวิเคราะห์จากระยะห่างของความสัมพันธ์ระหว่างคำศัพท์ของเอลิเมนต์และเพิ่มการวิเคราะห์ในส่วนของการให้คะแนนความสัมพันธ์ระหว่างเอลิเมนต์ต่าง ๆ ที่อยู่ภายในเอกสารวิสเดิลเพื่อเพิ่มความแม่นยำในการวิเคราะห์ความคล้ายคลึงกันของเว็บเซอร์วิซในเชิงโครงสร้าง

### 2.5.7 On The Evolution of Services

งานวิจัยของ Andrikopoulos และคณะ [18] นำเสนอการเปลี่ยนแปลงของเว็บเซอร์วิซในลักษณะต่าง ๆ ซึ่งเกิดขึ้นได้หลายเหตุผล เช่น เปลี่ยนแปลงเมื่อเว็บเซอร์วิซต้องการแก้ไขปัญหาที่เกิดขึ้นเพื่อไม่ให้เกิดขึ้นอีก และปรับปรุงคุณภาพของเว็บเซอร์วิซให้ดียิ่งขึ้น เป็นต้น ซึ่งการเปลี่ยนแปลงเหล่านี้อาจเกิดผลกระทบต่อผู้ใช้บริการ นำไปสู่อุปสรรคในการทำงานของผู้ใช้บริการได้ งานวิจัยนี้กล่าวถึงการเปลี่ยนแปลงหลัก 3 ประการ ได้แก่ การเปลี่ยนแปลงเชิงโครงสร้าง การเปลี่ยนแปลงเชิงพฤติกรรม และการเปลี่ยนแปลงเชิงนโยบายการใช้งานเว็บเซอร์วิซ นอกจากนี้งานวิจัยได้นำเสนอว่าหากการเปลี่ยนแปลงของเว็บเซอร์วิซไม่กระทบต่อการใช้งานของผู้ใช้บริการจะเรียกว่าเป็นการเปลี่ยนแปลงแบบ มีความเข้ากันได้ย้อนหลัง (Backward Compatibility) และการ

เปลี่ยนแปลงที่ทำให้เว็บเซอร์วิซสามารถรองรับการเรียกใช้จากผู้ใช้บริการซึ่งมีรายละเอียดของเอกสารวิสเดิลที่เป็นรุ่นใหม่กว่า เรียกว่าเป็นการเปลี่ยนแปลงแบบ มีความเข้ากันได้ไปข้างหน้า (Forward Compatibility) สำหรับการเปลี่ยนแปลงที่เป็นแบบเข้ากันได้ย้อนหลัง งานวิจัยได้ระบุเงื่อนไขหลายประการที่ทำให้เว็บเซอร์วิซมีความสามารถรองรับการทำงานเข้ากับผู้ใช้บริการซึ่งมีรายละเอียดของเอกสารวิสเดิลที่เป็นรุ่นเก่ากว่า เช่น เว็บเซอร์วิซรุ่นใหม่กว่ามีการเพิ่มโอเปอเรชัน เพิ่มพารามิเตอร์ที่เป็นแบบ Optional และการเปลี่ยนแปลงโดยแก้ไขกระบวนการทำงานข้างใน อีกทั้งโครงสร้างชนิดข้อมูลมีความสัมพันธ์กันแบบ Subtyping ซึ่งมีความสามารถใช้งานแทนที่ (Substitutability) กันได้

### 2.5.8 End-to-End Versioning Support for Web Services

งานวิจัยของ Leitner และคณะ [19] นำเสนอการเปลี่ยนแปลงในลักษณะต่าง ๆ ของเว็บเซอร์วิซบนพื้นฐานของเอกสารวิสเดิล ได้แก่ การเปลี่ยนแปลงเชิงโครงสร้างคือ การเปลี่ยนแปลงของโอเปอเรชัน การเปลี่ยนแปลงของพารามิเตอร์อินพุต การเปลี่ยนแปลงของพารามิเตอร์เอาต์พุต และการเปลี่ยนแปลงของชนิดข้อมูล เมื่อกำหนดการเปลี่ยนแปลงในลักษณะต่าง ๆ ได้แล้วสามารถนำไปวางแผนการจัดการเวอร์ชัน (Version) ได้เมื่อเกิดการเปลี่ยนแปลงในรูปแบบต่าง ๆ ที่กล่าวมา และงานวิจัยนี้ได้ออกแบบระบบที่ทำให้ผู้ใช้บริการเว็บเซอร์วิซสามารถเลือกเวอร์ชันของเว็บเซอร์วิซได้โดยไม่กระทบต่อโค้ดโปรแกรมของผู้ใช้บริการ โดยเรียกผ่านตัวกลาง (Mediator) ที่ระบบจัดช่องทางการเรียกใช้งานไว้ให้ ซึ่งเรียกว่าการทำ Dynamic Invocation ซึ่งเป็นการเลือกเวอร์ชันของเว็บเซอร์วิซและเรียกใช้งานในขณะที่ระบบกำลังทำงานอยู่ (Runtime)

### 2.5.9 การเปรียบเทียบกับงานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้องกับระบบการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันพิจารณาได้ในหลายแง่มุมของเว็บเซอร์วิซดังตารางที่ 2.1 ซึ่งผู้วิจัยแสดงการเปรียบเทียบความแตกต่างระหว่างการค้นคืนในงานของผู้วิจัยกับงานวิจัยอื่นที่เกี่ยวข้อง

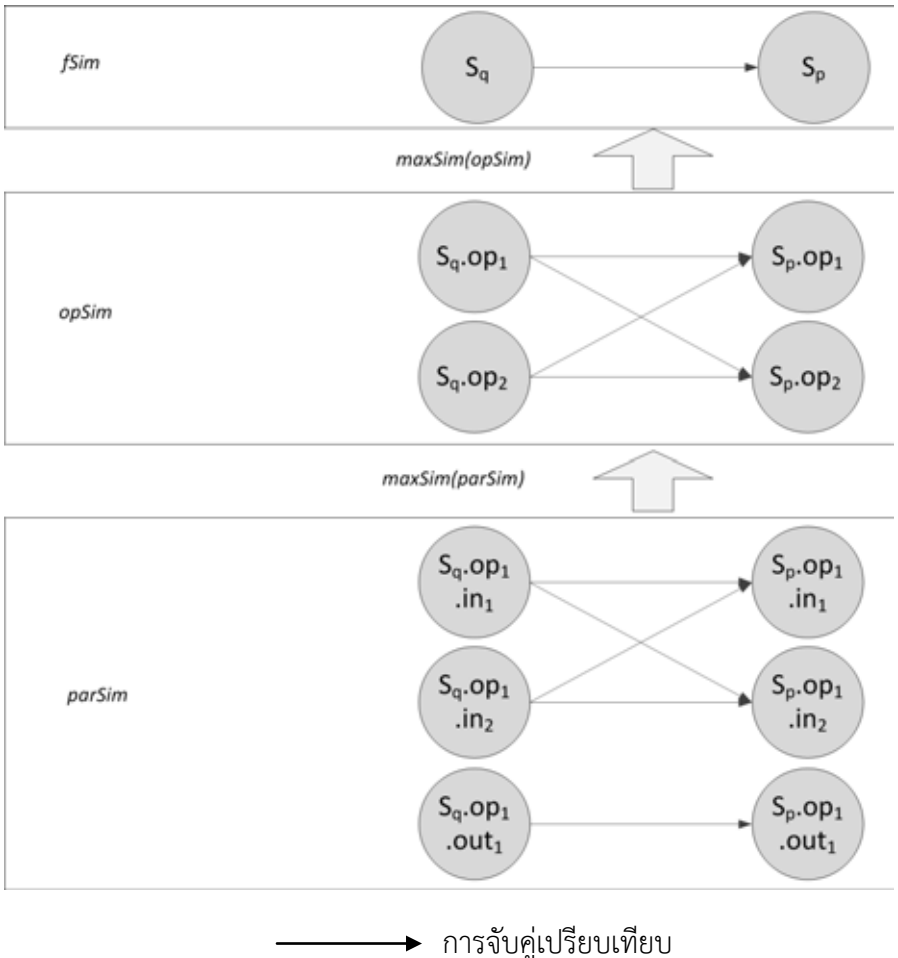
**ตารางที่ 2.1 การเปรียบเทียบระบบการค้นคืนเว็บเซอร์วิซ**

| | งานวิจัยของ Hatzi และคณะ [6] | งานวิจัยของ Elgazzar และคณะ [7] | งานวิจัยของ Plebani และ Pernici [8] | งานวิจัยของ Stroulia และ Wang [9] | งานวิจัยของ Liu และคณะ [10] | งานของผู้วิจัย |
|---|---|---|---|---|---|---|
| ค้นคืนโดยพิจารณาจากคำหลัก | x | x | | | | |
| ค้นคืนโดยพิจารณาจากความคล้ายคลึงกันเชิงโครงสร้าง | | | x | x | x | x |
| ค้นคืนโดยพิจารณาจากความคล้ายคลึงกันเชิงความหมายของเทอมตามออนโทโลจี | x | | x | x | | x |
| การวิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิซ | | | | | | x |

# บทที่ 3

# การเปรียบเทียบความคล้ายคลึงกันของเว็บเซอร์วิซ

การที่จะทราบได้ว่าเว็บเซอร์วิซที่ต้องการใช้งานแทนที่นั้นมีความคล้ายคลึงกันมากน้อยเพียงใดจำเป็นที่จะต้องมีกระบวนการประมวลผลและให้ค่าระดับความคล้ายคลึงโดยรวมในเชิงโครงสร้างของเว็บเซอร์วิซดังกล่าว โดยงานวิจัย [8] เสนออัลกอริทึมยูอาร์บีอี (URBE) ซึ่งทำการเปรียบเทียบความคล้ายคลึงกันด้วยการจับคู่ของเอลิเมนต์ (พอร์ตไทป์-พอร์ตไทป์ โอเปอเรชัน-โอเปอเรชัน อินพุต-อินพุต เอาต์พุต-เอาต์พุต) ในทุก ๆ คู่ที่เป็นไปได้ของเว็บเซอร์วิซต้นทาง (Query) และเว็บเซอร์วิซปลายทาง (Provide) ที่ต้องการเปรียบเทียบกัน ผู้วิจัยเสนอการปรับปรุงอัลกอริทึมยูอาร์บีอี โดยเรียกว่า อัลกอริทึมเอ็ม-ยูอาร์บีอี (M-URBE) ซึ่งมีภาพรวมการประเมินความคล้ายคลึงของเว็บเซอร์วิซดังภาพที่ 3.1




ภาพที่ 3.1 การให้คะแนนความคล้ายคลึงกันของเว็บเซอร์วิซ [8]

จากภาพที่ 3.1 อธิบายฟังก์ชันการคิดคะแนนความคล้ายคลึงกันในแต่ละฟังก์ชันได้ดังนี้

- *parSim* เป็นฟังก์ชันที่อยู่ล่างสุด ใช้ในการให้คะแนนการเข้าคู่พารามิเตอร์ทั้งข้อมูลอินพุตและข้อมูลเอาต์พุตภายใต้โอเปอเรชันนั้น ๆ โดย *parSim* จะคิดคะแนนตามชนิดข้อมูลของพารามิเตอร์ได้แก่ชนิดข้อมูลที่เป็น Complex Type และ Simple Type โดยที่ชนิดข้อมูลจะถูกคำนวณในฟังก์ชัน *datatypeSim* และหากเป็นชนิดข้อมูลที่เป็น Complex Type จะมีการให้คะแนนของความคล้ายคลึงของชื่อด้วยฟังก์ชัน *nameSim* ด้วย คะแนนการเข้าคู่ในทุก ๆ คู่ของเอลิเมนต์ที่เป็นไปได้จะต้องถูกนำไปหาค่าที่เหมาะสมที่ทำให้การเข้าคู่นี้มีคะแนนโดยรวมของการจับคู่ทั้งหมดสูงที่สุดด้วยการนำไปเข้าฟังก์ชัน *maxSim* ก่อนที่จะส่งคะแนนการเข้าคู่ที่สูงที่สุดนี้ให้ฟังก์ชัน *opSim* ซึ่ง ฟังก์ชัน *maxSim* และ *datatypeSim* จะกล่าวในหัวข้อถัดไป
- *opSim* เป็นฟังก์ชันที่อยู่ลำดับชั้นต่อจาก *parSim* ใช้ในการให้คะแนนการเข้าคู่ของทุก ๆ โอเปอเรชันที่อยู่ภายใต้เว็บเซอร์วิซนั้น ซึ่งในแต่ละโอเปอเรชันจะประกอบไปด้วยพารามิเตอร์ของข้อมูลอินพุต และพารามิเตอร์ของข้อมูลเอาต์พุต คะแนนการเข้าคู่ในทุก ๆ คู่ของโอเปอเรชันที่เป็นไปได้จะต้องถูกนำไปหาค่าที่เหมาะสมที่ทำให้การเข้าคู่นี้มีคะแนนโดยรวมของการจับคู่ทั้งหมดสูงที่สุดด้วยการนำไปเข้าฟังก์ชัน *maxSim* และมีการให้คะแนนของความคล้ายคลึงของชื่อด้วยฟังก์ชัน *nameSim* เช่นกัน การให้คะแนนของพารามิเตอร์ทั้งสองแบบนี้จะถูกคิดคะแนนให้เป็นความคล้ายคลึงของโอเปอเรชันแต่ละคู่
- *fSim* เป็นฟังก์ชันที่อยู่ชั้นบนสุด ใช้ในการให้คะแนนการเข้าคู่ของเว็บเซอร์วิซเชิงโครงสร้างทั้งหมดโดยมีช่วงคะแนนเป็นเลขทศนิยมตั้งแต่ 0 ถึง 1 ค่าคะแนนเท่ากับ 0 หมายความว่าเว็บเซอร์วิซไม่มีความคล้ายคลึงกันในเชิงโครงสร้าง และ ค่าคะแนนเท่ากับ 1 หมายความว่าเว็บเซอร์วิซมีความคล้ายคลึงกันในเชิงโครงสร้าง นั่นคือมีความเป็นไปได้ที่จะนำเว็บเซอร์วิซนี้มาใช้งานแทนที่เว็บเซอร์วิซปัจจุบันที่ใช้งานอยู่ได้

ในขั้นตอนก่อนที่จะให้คะแนนความคล้ายคลึงกันระหว่างเว็บเซอร์วิซต้นทางและปลายทาง เอกสารวิสเดิลซึ่งเป็นมาตรฐานสำหรับสร้างข้อกำหนดเว็บเซอร์วิซจะถูกแปลงให้อยู่ในรูปแบบเชิงนามธรรมก่อนที่จะเข้าสู่กระบวนการค้นหาเว็บเซอร์วิซที่มีความคล้ายคลึงกัน แนวคิดคือต้องการแยกส่วนของการประมวลผลเอกสารวิสเดิลเวอร์ชันต่าง ๆและ ส่วนของการประมวลผลของการให้คะแนนความคล้ายคลึงของเว็บเซอร์วิซออกจากกันอย่างชัดเจนในแง่ของพัฒนาโปรแกรม วิธีการดังกล่าวได้รับแนวคิดจาก [8] โดยให้ $\sigma$ แทนเซตของโอเปอเรชันทั้งหมดภายในเว็บเซอร์วิซดังนี้

- $\sigma_i$ = (name, {op}) อธิบายพอร์ตไทป์ของเว็บเซอร์วิซซึ่งประกอบไปด้วยชื่อ (name) และโอเปอเรชัน (op) ทั้งหมดภายในพอร์ตไทป์
- op = (name, {$input_m$, $output_n$}, modelReference) อธิบายโอเปอเรชันซึ่งประกอบไปด้วยชื่อ (name) และอินพุต (input) จำนวน m เอาต์พุต (output) จำนวน n และการอ้างอิงไปยังเทอมในออนโทโลจี (modelReference) ซึ่งอธิบายความหมายของโอเปอเรชันโดยใช้วิธีกำกับความหมายตาม เอสเอวิสเดิล [16]
- input = (name, type, modelReference) อธิบายข้อมูลอินพุตของโอเปอเรชันซึ่งประกอบไปด้วยชื่อ (name) ชนิดข้อมูลของอินพุต (type) และการอ้างอิงไปยังเทอมในออนโทโลจี (modelReference) ซึ่งอธิบายความหมายของข้อมูลอินพุต
- output = (name, type, modelReference) อธิบายข้อมูลเอาต์พุตของโอเปอเรชันซึ่งประกอบไปด้วยชื่อ (name) ชนิดข้อมูลของเอาต์พุต (type) และการอ้างอิงไปยังเทอมในออนโทโลจี (modelReference) ซึ่งอธิบายความหมายของข้อมูลเอาต์พุต

เพราะฉะนั้นตัวอย่างของเว็บเซอร์วิซในภาพที่ 3.2 จะได้เว็บเซอร์วิซที่ถูกแปลงให้อยู่ในรูปแบบเชิงนามธรรมในภาพที่ 3.3

```
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
…
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://tempuri.org/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:sawsdl="http://www.w3.org/ns/sawsdl">
 <wsdl:types>
  <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
    <s:element name="SearchPolicy">
     <s:complexType>
      <s:sequence>
       <s:element minOccurs="0" name="request" type="tns:PolicyRequest" />
      </s:sequence>
     </s:complexType>
    </s:element>
    <s:complexType name="PolicyRequest"
sawsdl:modelReference="http://127.0.0.1/ontology/insurance.owl# Request ">
     <s:sequence>
      <s:element minOccurs="1" name="PolicyNumber" type="s:string" />
      <s:element minOccurs="1" name="ReferenceNumber" type="s:int" />
     </s:sequence>
    </s:complexType>
    …
   </s:schema>
 </wsdl:types>
 <wsdl:message name="SearchPolicySoapIn" >
  <wsdl:part name="parameters" element="tns:SearchPolicy" />
 </wsdl:message>
 <wsdl:message name="SearchPolicySoapOut" >
  <wsdl:part name="parameters" element="tns:SearchPolicyResponse" />
 </wsdl:message>
 <wsdl:portType name="PolicyServiceSoap"
sawsdl:modelReference="http://127.0.0.1/ontology/insurance.owl#PolicyInquiryService">
 <wsdl:operation name="SearchPolicy"
sawsdl:modelReference="http://127.0.0.1/ontology/insurance.owl#PolicySearching">
  <wsdl:input message="tns:SearchPolicySoapIn" />
  <wsdl:output message="tns:SearchPolicySoapOut" />
 </wsdl:operation>
</wsdl:portType>
…
</wsdl:definitions>
```

ภาพที่ 3.2 ตัวอย่างเว็บเซอร์วิซ

```
σ.name = PolicyServiceSoap
σ.op1 = {
        σ.op1.name = SearchPolicy,
                σ.op1.inputPar1 = {
                        σ.op1.inputPar1.name = request,
                        σ.op1.inputPar1.type = PolicyRequest
                        σ.op1.inputPar1.modelReference =
http://127.0.0.1/ontology/insurance.owl#Request
                                σ.op1.inputPar1.input1 = {
                                        σ.op1.inputPar1.input1.name = PolicyNumber,
                                        σ.op1.inputPar1.input1.type = String
                                }
                                σ.op1.inputPar1.input2 = {
                                        σ.op1.inputPar1.input2.name = ReferenceNumber,
                                        σ.op1.inputPar1.input2.type = Int
                                }
                …
}
```

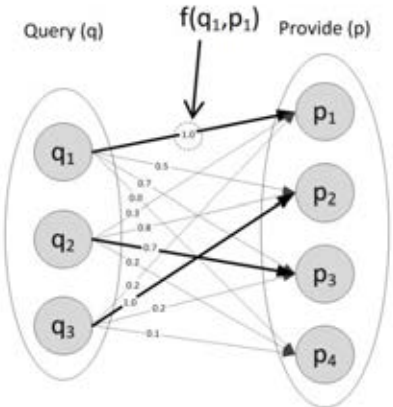
ภาพที่ 3.3 ตัวอย่างเว็บเซอร์วิซที่ถูกแปลงให้อยู่ในรูปแบบเชิงนามธรรม

## 3.1 การเปรียบเทียบความคล้ายคลึงกันของเว็บเซอร์วิซในเชิงโครงสร้าง

ในหัวข้อนี้เป็นการอธิบายรายละเอียดของแต่ละฟังก์ชันการให้คะแนนความคล้ายคลึงในส่วนต่าง ๆ ของเว็บเซอร์วิซในเชิงโครงสร้างที่ได้กล่าวในภาพที่ 3.1 ดังนี้

### 3.1.1 ฟังก์ชันการให้คะแนนที่สูงที่สุด (maxSim)

การเข้าคู่ระหว่างเอลิเมนต์ต้นทาง (p) ไปยังเอลิเมนต์ปลายทาง (q) จะทำกับทุก ๆ คู่ของเอลิเมนต์ที่เป็นไปได้ดังตัวอย่างในภาพที่ 3.4 โดยการเข้าคู่ในแต่ละคู่ที่เป็นไปได้จะถูกคำนวณคะแนนออกมาเป็นค่าความคล้ายคลึงกันระหว่างเอลิเมนต์ ซึ่งก็คือค่าฟังก์ชัน f แล้วจึงทำการรวมคะแนนในทุก ๆ คู่เพื่อหาผลรวมคะแนนการเข้าคู่ทั้งหมดระหว่างเอลิเมนต์ต้นทางไปยังเอลิเมนต์ปลายทาง จากนั้นทำการค้นหาคู่ที่ทำให้ผลรวมคะแนนการเข้าคู่ทั้งหมดมีค่ามากที่สุดแล้วหารด้วยจำนวนเอลิเมนต์ปลายทาง [20] นั่นก็คือ หากจำนวนเอลิเมนต์ต้นทางมีมากกว่าเอลิเมนต์ปลายทางจะได้ผลรวมคะแนนการเข้าคู่มากกว่ากรณีที่จำนวนเอลิเมนต์ต้นทางน้อยกว่าปลายทางดังสมการ (1)




$maxSim(f(q,p)) = (f(q_1,p_1)+f(q_2,p_3)+f(q_3,p_2))/4 = (1.0+0.7+1.0)/4 = 0.675$

ภาพที่ 3.4 การเข้าคู่ในทุก ๆ คู่ที่เป็นไปได้ระหว่างเอลิเมนต์ต้นทางไปยังเอลิเมนต์ปลายทางปรับจาก [8]

$$maxSim(f(q,p)) = \frac{max(\sum_{j=1..|p|}^{i=1..|q|} f(q_i,p_j))}{|p|} \quad (1)$$

โดยที่ q = เซตของเอลิเมนต์ภายในเว็บเซอร์วิซต้นทาง และ $q_i \in q$,

p = เซตของเอลิเมนต์ภายในเว็บเซอร์วิซปลายทาง และ $p_i \in p$,

f = ฟังก์ชันที่ให้คะแนนการเข้าคู่ เช่น *parSim*, *opSim* และ *termSim* ซึ่ง f ให้คะแนนในช่วงตั้งแต่ 0 ถึง 1 และ *maxSim* ให้คะแนนในช่วงตั้งแต่ 0 ถึง 1

|P| = จำนวนเอลิเมนต์ที่อยู่ภายในเว็บเซอร์วิซปลายทาง

จากสมการข้างต้น ยิ่งมีจำนวนเอลิเมนต์ปลายทางที่ไม่สามารถจับคู่กับเอลิเมนต์ต้นทางมากเท่าไร ก็จะยิ่งทำให้คะแนนจากฟังก์ชัน *maxSim* น้อยลงไปด้วย ซึ่งเป็นการสะท้อนว่าการทำแมปปิงหรือการแปลงเพื่อทำให้เว็บเซอร์วิซปลายทางทำงานแทนได้นั้นจะยิ่งยากขึ้น แต่หากเอลิเมนต์ปลายทางถูกจับคู่ได้หมด โดยมีเอลิเมนต์ต้นทางเหลืออยู่ จะถือว่าเว็บเซอร์วิซปลายทางเข้ากันได้ในเชิงโครงสร้างกับเว็บเซอร์วิซต้นทาง เพราะสามารถมั่นใจได้ว่า จะสามารถสร้างแมปปิงจากการทำงานตามแบบของต้นทางไปยังปลายทางได้ [20]

### 3.1.2 ฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันของชื่อ (nameSim)

การให้คะแนนความคล้ายคลึงของชื่อจากงานวิจัย [8] วัดด้วยการนำคำศัพท์สองคำมาเปรียบเทียบกันโดยศัพท์สองคำนี้อาจมีความหมายที่เชื่อมโยงถึงกันได้ซึ่งสะท้อนออกมาเป็นระดับคะแนนของความคล้ายคลึง โดยความคล้ายคลึงนี้อาจคล้ายคลึงในระดับรากศัพท์เดียวกันหรือคล้ายคลึงโดยเป็นคำที่มีความหมายเดียวกันแต่เขียนอยู่ในรูปที่ต่างกัน งานวิจัย [8] ได้ใช้เครื่องมือ WordNet [21] มาใช้ในการเปรียบเทียบความคล้ายคลึงระหว่างคำสองคำ

อย่างไรก็ตามการพิจารณาคำสองคำที่อยู่ภายในเอกสารวิสเดิลนั้น การพัฒนาโปรแกรมมีความเป็นไปได้ที่ผู้พัฒนาจะมีการตั้งชื่อให้แก่ โอเปอเรชัน ชื่อของพารามิเตอร์ข้อมูลอินพุตและข้อมูลเอาต์พุต และชื่อของชนิดข้อมูลด้วยรูปแบบที่หลากหลายเพื่อให้สื่อความหมาย ดังนั้นงานวิจัย [8] จึงมีกฎการแยกชื่อของเอลิเมนต์ตามรูปแบบการตั้งชื่อต่าง ๆเพื่อใช้ในการเปรียบเทียบดังตารางที่ 3.1

**ตารางที่ 3.1 กฎการแยกคำตามรูปแบบการตั้งชื่อของเอลิเมนต์ [8]**

| กฎ | ตัวอย่างคำดั้งเดิม | ตัวอย่างหลังจากแยกคำ |
|---|---|---|
| การเปลี่ยนตัวอักษรตัวพิมพ์เล็กหรือตัวพิมพ์ใหญ่ | PolicyNumber | policy, number |
| การลบขีดล่าง | Policy_Number | policy, number |
| การลบตัวเลขที่ต่อท้ายคำ | PolicyNumber1 | policy, number |

เมื่อได้คำต่าง ๆ ที่ถูกแยกออกมาจากชื่อของแต่ละเอลิเมนต์แล้ว จากนั้นจึงนำมาทำการเข้าคู่และให้คะแนนความคล้ายคลึงของแต่ละคำที่จับคู่กันในทุก ๆ คู่ที่เป็นไปได้ ด้วยฟังก์ชัน *termSim* ในงานวิจัยนี้ผู้วิจัยได้เพิ่มการให้คะแนนความคล้ายคลึงกันของคำด้วยอัลกอริทึม Levenshtein Distance [22] ซึ่งเป็นการให้คะแนนความคล้ายคลึงระหว่างคำโดยพิจารณาจากตัวอักษรที่อยู่ภายในคำว่าแตกต่างกันมากน้อยเพียงใดโดยไม่ได้พิจารณาในความหมายของคำ การเพิ่มการให้คะแนนของ

คำในลักษณะนี้เป็นการเพิ่มความแม่นยำในการเปรียบเทียบทั้งชื่อของเอลิเมนต์ที่คล้ายคลึงกันในแง่ของความหมายและชื่อของเอลิเมนต์ที่มีรูปคำคล้ายกันในรูปแบบของการย่อคำ เช่น คำว่า Num การตั้งชื่อย่อในลักษณะนี้จะไม่พบความหมายใน WordNet แต่สามารถเปรียบเทียบความคล้ายคลึงกันของรูปคำกับคำว่า Number ได้เป็นต้น ดังนั้นฟังก์ชัน *termSim* ที่ปรับปรุงโดยผู้วิจัยเขียนได้ในรูปของสมการ (2)

$$\begin{aligned} termSim(\{t_{q,i}\},\{t_{p,j}\}) &= Weight_{WordNet} \cdot termSim_{WordNet}(\{t_{q,i}\},\{t_{p,j}\}) \\ &+ Weight_{Levenshtein} \cdot termSim_{Levenshtein}(\{t_{q,i}\},\{t_{p,j}\}) \end{aligned} \quad (2)$$

โดยที่ $termSim_{WordNet}$ = ฟังก์ชันการให้คะแนนความคล้ายคลึงกันของคำด้วยโปรแกรม WordNet ซึ่งมีค่าในช่วงตั้งแต่ 0 ถึง 1

$termSim_{Levenshtein}$ = ฟังก์ชันการให้คะแนนความคล้ายคลึงกันของคำด้วยวิธีของ Levenshtein distance ซึ่งถูกนอร์มัลไลซ์ให้อยู่ในช่วงตั้งแต่ 0 ถึง 1

และ $Weight_{WordNet} + Weight_{Levenshtein} = 1$

หลังจากมีการให้คะแนนความคล้ายคลึงกันของคำในทุก ๆ คู่ที่เป็นไปได้แล้วจะทำการเลือกคำที่ทำให้คะแนนความคล้ายคลึงของชื่อมีค่าสูงที่สุดออกมาด้วยฟังก์ชัน *maxSim* ดังนั้นฟังก์ชัน *nameSim* เขียนได้ดังสมการ (3)

$$nameSim(n_q, n_p) = maxSim(termSim(\{t_{q,i}\},\{t_{p,j}\})) \quad (3)$$

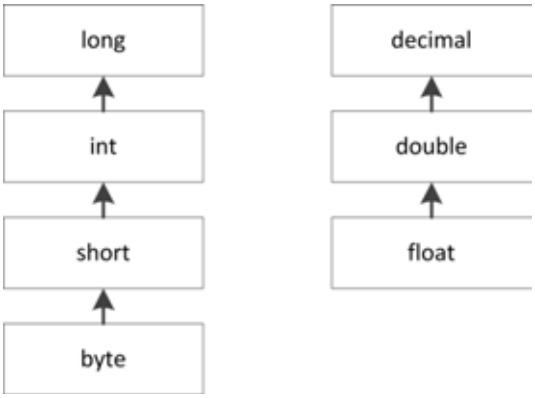
โดยที่ $n_q$ = ชื่อของเอลิเมนต์ต้นทางซึ่งประกอบด้วยคำ $t_{q,i}$ ที่ได้จากการแยกคำ

$n_p$ = ชื่อของเอลิเมนต์ปลายทางซึ่งประกอบด้วยคำ $t_{p,j}$ ที่ได้จากการแยกคำ

### 3.1.3 ฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันของชนิดข้อมูล (datatypeSim)

สำหรับฟังก์ชัน *datatypeSim* ของงานวิจัยนี้ผู้วิจัยได้เพิ่มเติมความสามารถของการพิจารณาชนิดข้อมูลแบบ Simple Type ซึ่งภายใน *datatypeSim* คือฟังก์ชัน *simpleDTSim* ซึ่งพบว่าในการกำหนดชนิดตัวแปรในเอกซ์เอ็มแอลสกีมา มีชนิดตัวแปรที่อยู่ในกลุ่มของชนิดตัวแปรที่เป็นเลขจำนวนเต็ม (Integer) และ กลุ่มของชนิดตัวแปรที่เป็นเลขจำนวนจริงในรูปแบบทศนิยม (Real) ซึ่งมีการสืบทอดกันเป็นลำดับชั้น [13] ดังภาพที่ 3.5 ซึ่ง งานวิจัย [8] ไม่ได้พิจารณาในรายละเอียดถึงความเข้ากันได้ของชนิดข้อมูลต่างชนิดที่อยู่ในกลุ่มเดียวกัน โดยจะถือว่าชนิดข้อมูลที่

อยู่ในกลุ่มเดียวกันจะเข้าคู่กัน งานวิจัยนี้จึงได้นำแนวคิดของงานวิจัย [23] และ [24] ซึ่งมีวิธีการคำนวณการให้คะแนนของชนิดข้อมูลที่มีการสืบทอดกันเป็นลำดับชั้นโดยใช้วิธีการให้คะแนนในลักษณะของ Generalizable Nominal Attribute (GNA) [24] และการพิจารณาตามหลักการของ Covariance/Contravariance กล่าวคือ อินพุต ของ p ที่มีลักษณะทั่วไป (Generalized) กว่าอินพุตของ q จะได้คะแนนมากกว่าอินพุตของ p ที่มีลักษณะเฉพาะเจาะจง (Specialized) กว่า และเอาต์พุตของ p ที่มีลักษณะเจาะจงกว่าเอาต์พุต ของ q ก็จะได้คะแนนมากกว่าเอาต์พุตของ p ที่มีลักษณะทั่วไปกว่า




ภาพที่ 3.5 ลำดับชั้นของการสืบทอดชนิดตัวแปรในกลุ่มของ Integer และ Real

ตารางที่ 3.2 แสดงการให้คะแนนความคล้ายคลึงของชนิดข้อมูลสำหรับตัวแปร Simple Type ในฟังก์ชัน *simpleDTSim* และการปรับปรุงการให้คะแนนความคล้ายคลึงกันของชนิดข้อมูลตัวแปรในกลุ่ม Integer และ Real แสดงได้ดังตารางที่ 3.3 และ ตารางที่ 3.4

**ตารางที่ 3.2 การให้คะแนนการเข้าคู่ชนิดข้อมูลแบบ Simple Type ตามกลุ่มชนิดข้อมูล (ปรับจาก [8])**

| | | Query (q) | | | | |
|---|---|---|---|---|---|---|
| Provide (p) | | **Integer** | **Real** | **String** | **Date** | **Boolean** |
| | **Integer** | $GNA_{Integer}(q,p)$ | 0.5 | 0.3 | 0.1 | 0.1 |
| | **Real** | 1.0 | $GNA_{Real}(q,p)$ | 0.1 | 0.0 | 0.1 |
| | **String** | 0.7 | 0.7 | 1.0 | 0.8 | 0.3 |
| | **Date** | 0.1 | 0.0 | 0.1 | 1.0 | 0.0 |
| | **Boolean** | 0.1 | 0.0 | 0.1 | 0.0 | 1.0 |

**ตารางที่ 3.3 การให้คะแนนแบบ GNA สำหรับกลุ่มข้อมูลที่เป็นชนิด Integer [24]**

| | | Query (q) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Provide (p) | | Input | | | | Output | | | |
| | | **long** | **int** | **short** | **byte** | **long** | **int** | **short** | **byte** |
| | **long** | 1 | 1 | 1 | 1 | 1 | 0.5 | 0.33 | 0.25 |
| | **int** | 0.5 | 1 | 1 | 1 | 1 | 1 | 0.5 | 0.33 |
| | **short** | 0.33 | 0.5 | 1 | 1 | 1 | 1 | 1 | 0.5 |
| | **byte** | 0.25 | 0.33 | 0.5 | 1 | 1 | 1 | 1 | 1 |

**ตารางที่ 3.4 การให้คะแนนแบบ GNA สำหรับกลุ่มข้อมูลที่เป็นชนิด Real [24]**

| | | Query (q) | | | | | |
|---|---|---|---|---|---|---|---|
| Provide (p) | | Input | | | Output | | |
| | | **decimal** | **double** | **float** | **decimal** | **double** | **float** |
| | **decimal** | 1 | 1 | 1 | 1 | 0.5 | 0.3 |
| | **double** | 0.5 | 1 | 1 | 1 | 1 | 0.5 |
| | **float** | 0.33 | 0.5 | 1 | 1 | 1 | 1 |

นอกจากนี้ในงานวิจัยนี้ได้เพิ่มการให้คะแนนโดยหากพบว่าปลายทางเป็น Optional Parameter และไม่สามารถจับคู่ใด ๆได้จาก q จะได้คะแนนเป็น 1

อัลกอริทึมฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันของชนิดข้อมูล แสดงได้ดังภาพที่ 3.6

```
function datatypeSim(ele_q, ele_p)
   if(ele_q.dt is simpleType and ele_p.dt is simpleType)
     return nameSim(ele_q.name, ele_p.name) *
            simpleDTSim(ele_q.dt, ele_p.dt)
   else if (ele_q.dt is complexType and ele_p.dt is complexType)
    return nameSim(ele_q.name, ele_p.name) *
           datatypeSim  (ele_q.dt.elements, ele_p.dt.elements)
   else if ele_p.dt is Optional and ele_p is not matched by any ele_q
     return 1
   else
     return 0
end function
```

ภาพที่ 3.6 อัลกอริทึมฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันของชนิดข้อมูล (*datatypeSim*)

### 3.1.4 ฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันของพารามิเตอร์ (parSim)

การให้คะแนนการเข้าคู่ความคล้ายคลึงกันของพารามิเตอร์สามารถให้คะแนนได้ทั้งพารามิเตอร์ที่เป็นข้อมูลอินพุตและพารามิเตอร์ที่เป็นข้อมูลเอาต์พุต ซึ่งระดับคะแนนที่ได้จะบ่งชี้ว่า โอเปอเรชันของเว็บเซอร์วิซต้นทางนั้นมีพารามิเตอร์ที่มีความคล้ายคลึงกับพารามิเตอร์ของโอเปอเรชันของเว็บเซอร์วิซปลายทางมากน้อยเพียงใด การให้คะแนนความคล้ายคลึงกันของพารามิเตอร์สามารถทำได้ด้วยการวัดความคล้ายคลึงของชนิดข้อมูลและชื่อของพารามิเตอร์อินพุตดังสมการ (4) และของพารามิเตอร์เอาต์พุตดังสมการ (5) [8]

$$\begin{aligned} &parSim_{input}(op_q.input, op_p.input) \\ &= Weight_{namePar} \cdot nameSim(op_q.input.name, op_p.input.name) \\ &+ Weight_{typePar} \cdot datatypeSim(op_q.input.type, op_p.input.type) \end{aligned} \tag{4}$$

$$\begin{aligned} &parSim_{output}(op_q.output, op_p.output) \\ &= Weight_{namePar} \cdot nameSim(op_q.output.name, op_p.output.name) \\ &+ Weight_{typePar} \cdot datatypeSim(op_q.output.type, op_p.output.type) \end{aligned} \tag{5}$$

โดยที่ $Weight_{namePar} + Weight_{typePar} = 1$ และ parSim ให้คะแนนในช่วงตั้งแต่ 0 ถึง 1

### 3.1.5 ฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันของโอเปอเรชัน (opSim)

เมื่อได้คะแนนการเข้าคู่ความคล้ายคลึงกันของพารามิเตอร์แล้ว การให้คะแนนการเข้าคู่ความคล้ายคลึงกันของโอเปอเรชันจะให้คะแนนโดยแบ่งคะแนนของพารามิเตอร์ออกเป็นอัตราส่วนที่เท่ากันระหว่างพารามิเตอร์ข้อมูลอินพุตและพารามิเตอร์ข้อมูลเอาต์พุตและคิดคะแนนความคล้ายคลึงของชื่อโอเปอเรชันโดยอ้างอิงจากงานวิจัย [8] ดังสมการ (6)

$$\begin{aligned} &opSim(op_q, op_p) = \\ &Weight_{OperationName} \cdot nameSim(op_q.name, op_p.name) \\ &+ Weight_{Par} \cdot [0.5 \cdot maxSim(parSim_{input}(op_q.\{input\}, op_p.\{input\})) \\ &+ 0.5 \cdot maxSim(parSim_{output}(op_q.\{output\}, op_p.\{output\}))] \end{aligned} \tag{6}$$

โดยที่ $Weight_{OperationName} + Weight_{Par} = 1$ และ *opSim* ให้คะแนนในช่วงตั้งแต่ 0 ถึง 1

### 3.1.6 ฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันของเว็บเซอร์วิซ (fSim)

ความคล้ายคลึงกันในเชิงโครงสร้างของเว็บเซอร์วิซถูกให้คะแนนด้วยฟังก์ชัน *fSim* ซึ่งเป็นฟังก์ชันที่อยู่ระดับบนสุดที่ได้จากการหาคะแนนที่สูงที่สุดของการเข้าคู่ในทุก ๆโอเปอเรชันระหว่างเว็บ

เซอร์วิซต้นทางและเว็บเซอร์วิซปลายทาง นอกจากจะคิดคะแนนจากทุก ๆ คู่ของโอเปอเรชันแล้ว ยังคิดคะแนนความคล้ายคลึงจากชื่อของกลุ่มของโอเปอเรชันหรือพอร์ตไทป์ ซึ่งอ้างอิงจากงานวิจัย [8] ดังสมการ (7)

$$\begin{aligned} fSim(\sigma_q,\sigma_p) &= Weight_{portTypeName} \cdot nameSim(\sigma_q.name,\sigma_p.name) \\ &+ Weight_{Operations} \cdot \max Sim(opSim(\sigma_q.\{op\},\sigma_p.\{op\})) \end{aligned} \quad (7)$$

โดยที่ $Weight_{portTypeName} + Weight_{Operations} = 1$ และ *fSim* ให้คะแนนในช่วงตั้งแต่ 0 ถึง 1

### 3.2 การเปรียบเทียบความคล้ายคลึงกันของเว็บเซอร์วิซในเชิงความหมาย

การพิจารณาการให้คะแนนการเข้าคู่เว็บเซอร์วิซในเชิงความหมายเป็นอีกหนึ่งวิธีที่สามารถค้นหาเว็บเซอร์วิซที่มีความคล้ายคลึงกันได้ โดยงานวิจัย [8] ทำการให้คะแนนโดยรวมโดยยังยึดหลักของการให้คะแนนความคล้ายคลึงของเว็บเซอร์วิซเชิงโครงสร้างแต่สิ่งที่เปลี่ยนไปคือเปลี่ยนจากการใช้ฟังก์ชันการให้คะแนนความคล้ายคลึงกันของชื่อ (*nameSim*) ในหัวข้อที่ 3.1.2 ไปเป็นการใช้ฟังก์ชันการให้คะแนนความคล้ายคลึงของการกำกับความหมาย (*annSim*) ดังสมการ (8) ตามที่อ้างอิงโดย modelReference ที่ถูกอธิบายในเว็บเซอร์วิซเชิงนามธรรม modelReference นี้สามารถนำมาให้คะแนนความคล้ายคลึงกันของเว็บเซอร์วิซในเชิงความหมายได้บนพื้นฐานที่ว่าเว็บเซอร์วิซดังกล่าวอ้างอิงเทอมจากออนโทโลจีของโดเมนเดียวกัน จึงสามารถหาความสัมพันธ์ระหว่างเทอมที่อ้างอิงได้

$$annSim(a_q,a_p) = \begin{cases} pathSim(a_q,a_p), & a_q,a_p \text{ are both classes/properties} \\ classPropSim(a_q,a_p), & a_q \text{ is class}, a_p \text{ is property} \\ propClassSim(a_q,a_p), & a_q \text{ is property}, a_p \text{ is class} \end{cases} \quad (8)$$

โดยที่ $a_q$ = เทอมของออนโทโลจีที่ถูกกำกับในเว็บเซอร์วิซต้นทาง

$a_p$ = เทอมของออนโทโลจีที่ถูกกำกับในเว็บเซอร์วิซปลายทาง และ *annSim* ให้คะแนนในช่วงตั้งแต่ 0 ถึง 1

#### 3.2.1 ฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันระหว่างคลาส - คลาส และคุณสมบัติ - คุณสมบัติ (pathSim)

ฟังก์ชัน *pathSim* พิจารณาเทอมในออนโทโลจีที่กำกับอยู่ที่เว็บเซอร์วิซต้นทางและปลายทาง หากพบว่าเป็นคลาสหรือคุณสมบัติทั้งคู่จะสามารถให้คะแนนได้โดยวัดระยะห่างของลำดับชั้นที่สืบทอดกันระหว่างคลาสหรือระหว่างคุณสมบัติดังสมการ (9) ผู้วิจัยพบว่าในงานวิจัย [8] ไม่ได้

พิจารณาในทิศทางของความสัมพันธ์ของการเข้าคู่และการให้คะแนนเชิงความหมายโดยเมื่อพิจารณาการให้คะแนนเชิงความหมายของพารามิเตอร์ จะพิจารณาเพียงจำนวนลำดับชั้นเท่านั้น งานวิจัยนี้จึงนำเสนอการพิจารณาทิศทางของความสัมพันธ์ระหว่างคลาสหรือระหว่างคุณสมบัติในส่วนของการให้คะแนนเชิงความหมายของพารามิเตอร์ในลักษณะของความเข้ากันได้ของเว็บเซอร์วิซโดยพิจารณาตามหลักการของ Covariance/Contravariance เช่นเดียวกันกับหัวข้อที่ 3.1.3 และการพิจารณาทิศทางของความสัมพันธ์ระหว่างคลาสหรือระหว่างคุณสมบัติในส่วนของการให้คะแนนเชิงความหมายของพารามิเตอร์ได้แนวคิดจากงานวิจัย [25]

ผู้วิจัยเสนอการปรับปรุงฟังก์ชัน *pathSim* ในส่วนที่เป็นการให้คะแนนความคล้ายคลึงกันของพารามิเตอร์ดังสมการ (10) และ (11)

$$pathSim(a_q, a_p) = \begin{cases} 1 & Exact \\ 0 & Failed \\ \dfrac{1}{pathlength(a_q, a_p)+1} & a_q \text{ related to } a_p \end{cases} \quad (9)$$

$$pathSim_{parInput}(a_q, a_p) = \begin{cases} 1 & Exact \text{ or } a_q \subseteq a_p \\ 0.4 & Partial \\ 0 & Failed \\ \dfrac{1}{pathlength(a_q, a_p)+1} & a_p \subseteq a_q \end{cases} \quad (10)$$

$$pathSim_{parOutput}(a_q, a_p) = \begin{cases} 1 & Exact \text{ or } a_p \subseteq a_q \\ 0.4 & Partial \\ 0 & Failed \\ \dfrac{1}{pathlength(a_q, a_p)+1} & a_q \subseteq a_p \end{cases} \quad (11)$$

โดยที่ $pathlength(a_q,a_p)$ = ระยะห่างที่มากที่สุดของลำดับชั้นของเทอมที่ถูกกำกับความหมายไว้ที่เว็บเซอร์วิซต้นทางและเว็บเซอร์วิซปลายทาง

$a_p \subseteq a_q$ = เทอมที่ถูกกำกับความหมายจากเว็บเซอร์วิซปลายทางมีความสัมพันธ์กับเทอมที่ถูกกำกับความหมายจากเว็บเซอร์วิซต้นทางในระดับเฉพาะเจาะจงกว่า (Specialized)

$a_q \subseteq a_p$ = เทอมที่ถูกกำกับความหมายจากเว็บเซอร์วิซปลายทางมีความสัมพันธ์กับเทอมที่ถูกกำกับความหมายจากเว็บเซอร์วิซต้นทางในระดับทั่วไปกว่า (Generalized)

### 3.2.2 ฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันระหว่างคลาส – คุณสมบัติ (classPropSim)

ฟังก์ชัน *classPropSim* ให้คะแนนความคล้ายคลึงของเทอมในออนโทโลจีเมื่อเว็บเซอร์วิซต้นทางกำกับความหมายด้วยคลาสแต่เว็บเซอร์วิซปลายทางกำกับความหมายด้วยคุณสมบัติ ทำให้เว็บเซอร์วิซปลายทางจะสามารถใช้งานแทนที่ในเชิงของความหมายได้เพียงคุณสมบัติเดียวตามที่ปลายทางมีให้ ดังนั้นเมื่อทำการเข้าคู่เชิงความหมาย คะแนนความคล้ายคลึงที่ได้จะน้อยลงตามจำนวนของคุณสมบัติของคลาสในเว็บเซอร์วิซต้นทางที่ต้องการแทนที่ดังสมการ (12)

$$classPropSim(a_q, a_p) = \begin{cases} \dfrac{1}{\text{number of properties of } a_q}, & a_p \text{ is property of class } a_q \\ 0, & a_p \text{ is not property of class } a_q \end{cases} \quad (12)$$

### 3.2.3 ฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันระหว่างคุณสมบัติ – คลาส (propClassSim)

ฟังก์ชัน *propClassSim* ให้คะแนนความคล้ายคลึงของเทอมในออนโทโลจีเมื่อเว็บเซอร์วิซต้นทางกำกับความหมายด้วยคุณสมบัติแต่เว็บเซอร์วิซปลายทางกำกับความหมายด้วยคลาส หากคุณสมบัติของเว็บเซอร์วิซต้นทางที่กำกับความหมายไว้เป็นคุณสมบัติหนึ่งของคลาสที่ใช้กำกับความหมายให้กับเว็บเซอร์วิซปลายทางแล้วจะทำให้สามารถนำเว็บเซอร์วิซปลายทางมาใช้งานแทนที่ได้โดยสมบูรณ์ในเชิงความหมายดังสมการ (13)

$$propClassSim(a_q, a_p) = \begin{cases} 1, & a_q \text{ is property of class } a_p \\ 0, & a_q \text{ is not property of class } a_p \end{cases} \quad (13)$$

### 3.3 สรุปการปรับปรุงโดยงานของผู้วิจัย

จากที่กล่าวมาในหัวข้อของการให้คะแนนความคล้ายคลึงกันของเว็บเซอร์วิซทั้งในเชิงโครงสร้างและเชิงความหมายสามารถสรุปการปรับปรุงโดยงานของผู้วิจัยได้ดังตารางที่ 3.5

**ตารางที่ 3.5 สรุปการปรับปรุงโดยงานวิจัยของผู้วิจัย**

| **การให้คะแนนความคล้ายคลึงกันเชิงโครงสร้างของเว็บเซอร์วิซ** | | **การให้คะแนนความคล้ายคลึงกันเชิงความหมายของเว็บเซอร์วิซ** | |
|---|---|---|---|
| งานวิจัย [8] | งานของผู้วิจัย | งานวิจัย [8] | งานของผู้วิจัย |
| *fSim* | ไม่ได้ปรับปรุง | *fSim*$^{s}$ | ไม่ได้ปรับปรุง |
| *nameSim* | เพิ่มการให้คะแนนความแตกต่างของรูปคำด้วยอัลกอริทึม Levenshtein Distance | *annSim* | ไม่ได้ปรับปรุง |
| *parSim* | เพิ่มการพิจารณาความเข้ากันได้แบบ Contravariant Input/Covariant Output | *parSim*$^{s}$ | เพิ่มการพิจารณาความเข้ากันได้แบบ Contravariant Input/Covariant Output |
| *datatypeSim* | เพิ่มการพิจารณาชนิดข้อมูลที่เป็นแบบ Optional | *pathSim* | เพิ่มการพิจารณาความเข้ากันได้แบบ Contravariant Input/Covariant Output |
| *simpleDTSim* | ปรับปรุงการประเมินความคล้ายคลึงเชิงโครงสร้างในส่วนของการให้คะแนนการเข้าคู่ของชนิดข้อมูลของพารามิเตอร์ซึ่งอยู่ในกลุ่มเดียวกัน | *classPropSim* | ไม่ได้ปรับปรุง |
| | | *propClassSim* | ไม่ได้ปรับปรุง |

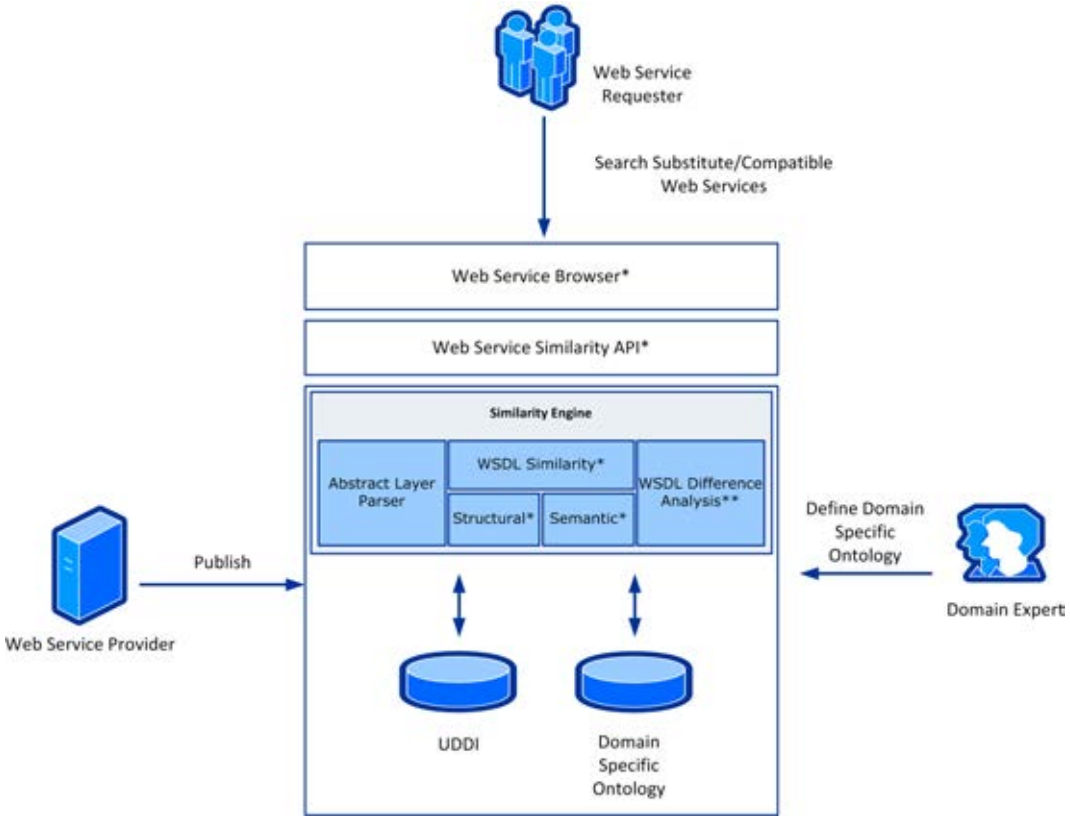
# บทที่ 4

# ระบบค้นคืนและวิเคราะห์ความแตกต่างสำหรับเว็บเซอร์วิซที่มีความคล้ายคลึงกัน

จากการแสดงการปรับปรุงการให้คะแนนความคล้ายคลึงกันของเว็บเซอร์วิซในเชิงโครงสร้างและเชิงความหมายจากที่กล่าวในบทที่ 3 ในบทนี้ได้นำฟังก์ชันต่าง ๆ ที่ให้คะแนนความคล้ายคลึงมาสร้างระบบค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกัน นอกจากนี้ได้เพิ่มส่วนของการวิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิซในหัวข้อต่อไปนี้

## 4.1 ภาพรวมของระบบ

เมื่อเกิดเหตุการณ์ที่ทำให้เว็บเซอร์วิซปัจจุบันที่ถูกใช้งานอยู่ไม่สามารถใช้งานได้ ผู้ใช้งานสามารถนำเอกสารวิสเดิลของเว็บเซอร์วิซนั้นไปค้นคืนยังระบบค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกัน ภาพรวมของระบบเป็นดังภาพที่ 4.1 โดยผู้ให้บริการเว็บเซอร์วิซ (Web Service Provider) สามารถเข้าไปประกาศการให้บริการกับฐานข้อมูลยูดีดีไอ โดยหากผู้ให้บริการเว็บเซอร์วิซต้องการที่จะกำกับความหมายเพิ่มเติมสามารถทำได้ด้วยการอ้างอิงเทอมในออนโทโลจีของโดเมนของเว็บเซอร์วิซ (Domain Specific Ontology) ซึ่งกำหนดโดยผู้เชี่ยวชาญในโดเมน (Domain Expert) และจัดเก็บไว้ในฐานข้อมูลของออนโทโลจีของโดเมน ผู้ใช้บริการเว็บเซอร์วิซ (Web Service Requester) สามารถเข้ามาค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกับเว็บเซอร์วิซที่ใช้งานอยู่โดยส่งเอกสารวิสเดิลเข้ามาประมวลผลในระบบผ่านทางเว็บเซอร์วิซเบราว์เซอร์ (Web Service Browser) ที่จัดทำขึ้นโดยผู้วิจัย โดยสามารถเลือกค้นคืนในเชิงโครงสร้างหรือเชิงความหมาย โดยใช้เครื่องประมวลความคล้ายคลึงของวิสเดิล (WSDL Similarity Engine) ได้ หากเป็นการค้นคืนเชิงโครงสร้าง ระบบจะนำฐานข้อมูลของ WordNet มาใช้ในการค้นหาคำศัพท์ของชื่อหรือคำของเอลิเมนต์ต่าง ๆภายในเอกสารวิสเดิล และหากเป็นการค้นคืนเชิงความหมาย ระบบจะใช้ภาษา SPARQL [26] ในการค้นหาและตรวจสอบความสัมพันธ์ในเชิงความหมายระหว่างเทอมของออนโทโลจีที่กำกับอยู่ในเว็บเซอร์วิซต้นทางกับเว็บเซอร์วิซที่ถูกประกาศในฐานข้อมูลของระบบหรือเว็บเซอร์วิซปลายทาง




* แสดงส่วนที่ทำการปรับปรุงในงานของผู้วิจัย ** แสดงส่วนที่ทำการเพิ่มในงานของผู้วิจัย

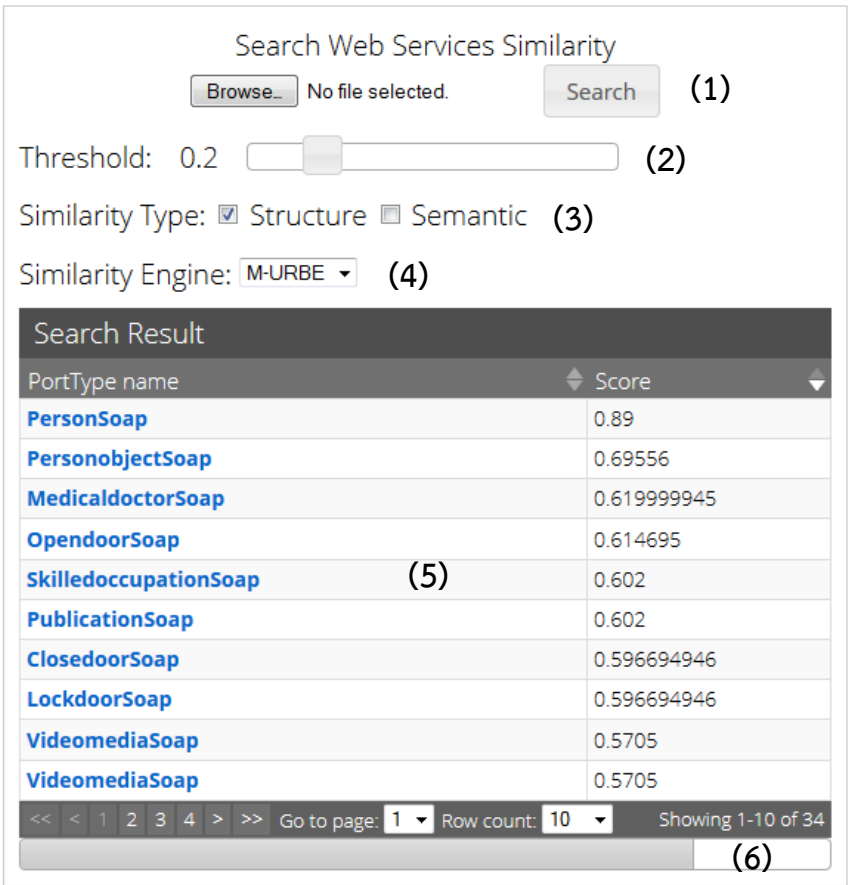
ภาพที่ 4.1 ระบบค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันและรายงานผลการแทนที่กับผู้ใช้งาน (ปรับจาก [8])

## 4.2 เว็บเซอร์วิซเบราว์เซอร์

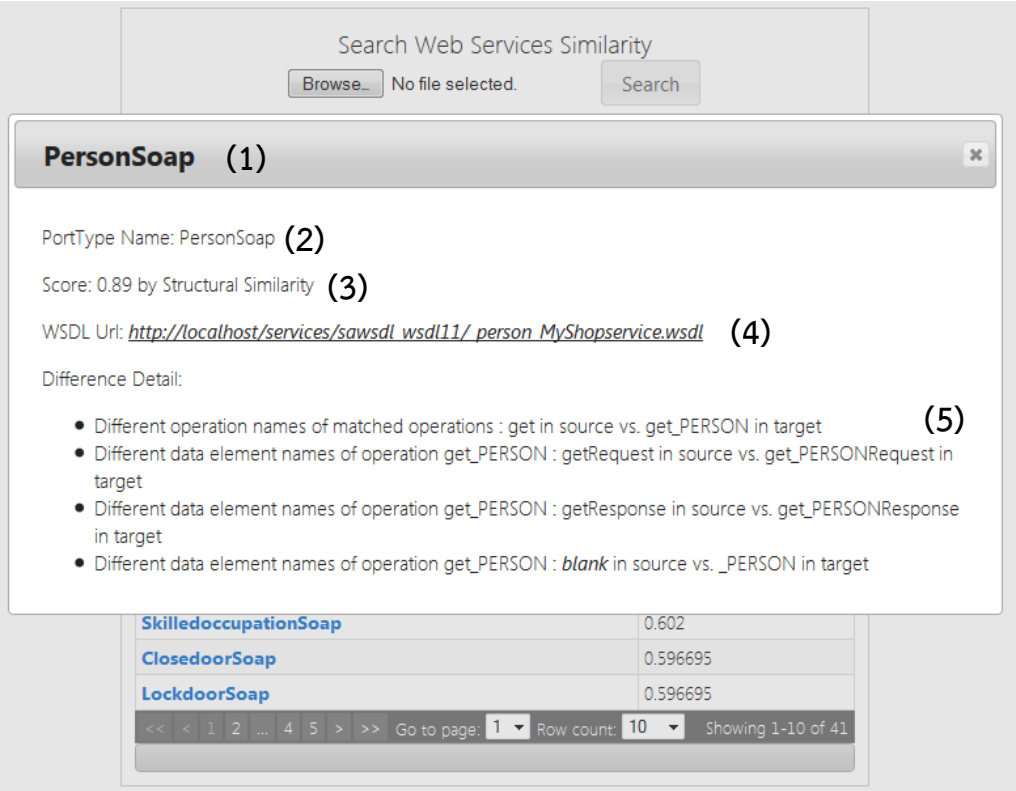
เว็บเซอร์วิซเบราว์เซอร์ (Web Service Browser) เป็นส่วนที่สามารถให้ผู้ใช้บริการเว็บเซอร์วิซเข้ามาค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกับเว็บเซอร์วิซที่ใช้งานอยู่โดยส่งเอกสารวิสเดิลเข้ามาประมวลผลได้ผ่านทางส่วนต่อประสานดังภาพที่ 4.2 โดยมีส่วนประกอบต่าง ๆ ตามหมายเลขดังนี้

- หมายเลข 1 คือส่วนที่สามารถให้ผู้ใช้เลือกเอกสารวิสเดิลที่ต้องการใช้ในการเป็นเว็บเซอร์วิซต้นทางสำหรับการค้นคืนเว็บเซอร์วิซที่คล้ายคลึงกัน
- หมายเลข 2 คือส่วนที่ผู้ใช้สามารถเลือกคะแนนของความเหมือนเว็บเซอร์วิซที่ได้จากฟังก์ชัน *fSim* โดยจะไม่แสดงเว็บเซอร์วิซที่ค้นคืนมาได้หากคะแนนความคล้ายคลึงน้อยกว่าค่าที่กำหนด หรือค่า Threshold

- หมายเลข 3 คือส่วนที่สามารถให้ผู้ใช้เลือกวิธีการให้คะแนนความคล้ายคลึงของเว็บเซอร์วิซได้ 2 วิธีคือ ความคล้ายคลึงกันในเชิงโครงสร้างและความคล้ายคลึงกันในเชิงความหมาย
- หมายเลข 4 คือตัวเลือกอัลกอริทึมในการให้คะแนนความคล้ายคลึงได้ 2 แบบคือ เอ็ม-ยูอาร์บีอี ที่เป็นอัลกอริทึมที่ได้ปรับปรุงในงานวิจัยนี้ และ ยูอาร์บีอี คืองานวิจัยของ Plebani และ Pernici [8] ที่เป็นอัลกอริทึมดั้งเดิม
- หมายเลข 5 คือส่วนที่แสดงเว็บเซอร์วิซที่ค้นคืนมาได้ เว็บเซอร์วิซที่แสดงต้องมีคะแนนมากกว่าค่า Threshold ที่กำหนดในหมายเลข 2 โดยเรียงลำดับตามเว็บเซอร์วิซปลายทางที่มีคะแนนความคล้ายคลึงมากที่สุดไปหาน้อยที่สุดและสามารถเลือกเว็บเซอร์วิซปลายทางที่ต้องการเพื่อดูรายละเอียดได้ในภาพที่ 4.3
- หมายเลข 6 คือแถบ Progress Bar แสดงการทำงานของเว็บเซอร์วิซเบราว์เซอร์




ภาพที่ 4.2 เว็บเซอร์วิซเบราว์เซอร์



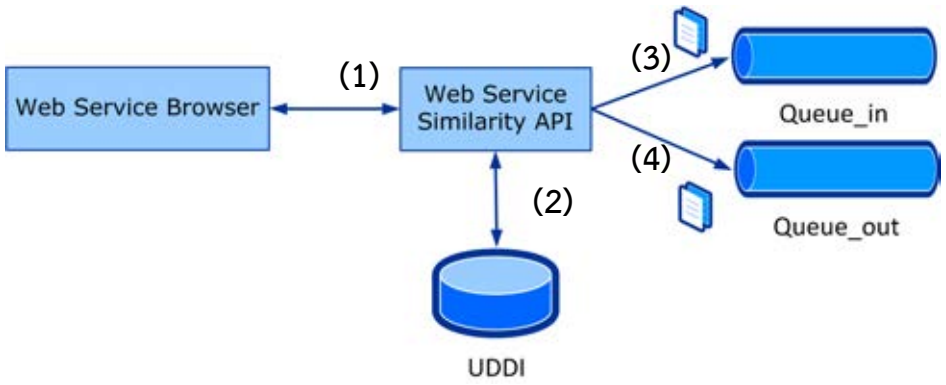

ภาพที่ 4.3 รายละเอียดของเว็บเซอร์วิซปลายทาง

ภาพที่ 4.3 มีส่วนประกอบต่าง ๆ ตามหมายเลขดังนี้

- หมายเลข 1 และหมายเลข 2 แสดงชื่อ พอร์ตไทป์ของเว็บเซอร์วิซปลายทาง
- หมายเลข 3 แสดงคะแนนความคล้ายคลึง
- หมายเลข 4 ที่อยู่ URL เอกสารวิสเดิลของเว็บเซอร์วิซปลายทาง
- หมายเลข 5 แสดงรายละเอียดความแตกต่างระหว่างเว็บเซอร์วิซต้นทางและเว็บเซอร์วิซปลายทาง

### 4.3 เอพีไอด้านความคล้ายคลึงกันของเว็บเซอร์วิซ

เอพีไอด้านความคล้ายคลึงกันของเว็บเซอร์วิซ (Web Service Similarity API) เป็นจุดเชื่อมต่อเพื่อให้เว็บเซอร์วิซเบราว์เซอร์ติดต่อกับส่วนประมวลผลความคล้ายคลึงกันของเว็บเซอร์วิซ โดยเป็นการให้บริการเชื่อมต่อแบบเว็บเซอร์วิซดังภาพที่ 4.4




ภาพที่ 4.4 โครงสร้างของเอพีไอด้านความคล้ายคลึงกันของเว็บเซอร์วิซ

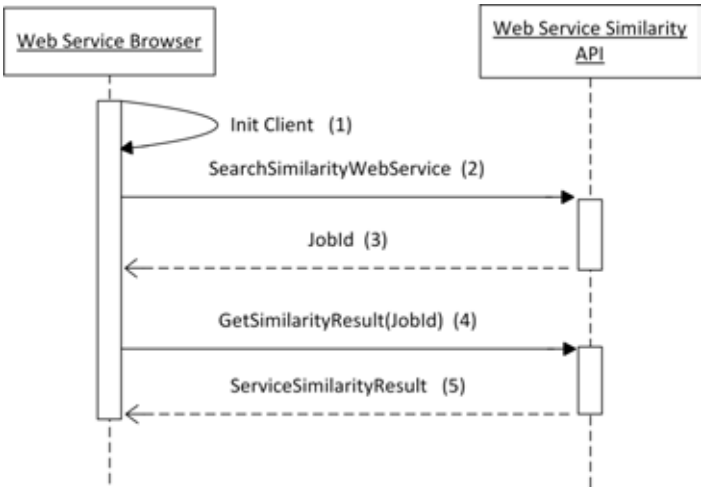
ภาพที่ 4.4 อธิบายส่วนประกอบต่าง ๆ ตามหมายเลขได้ดังนี้

- หมายเลข 1 เป็นจุดที่ให้เว็บเซอร์วิซเบราว์เซอร์เชื่อมต่อไปยังเอพีไอด้านความคล้ายคลึงกันของเว็บเซอร์วิซผ่านช่องทางเว็บเซอร์วิซ
- หมายเลข 2 เมื่อมีการร้องขอไปยังเอพีไอด้านความคล้ายคลึงกันของเว็บเซอร์วิซจะมีการดึงเว็บเซอร์วิซปลายทางทั้งหมดสำหรับการหาเว็บเซอร์วิซที่มีความคล้ายคลึงกันจากยูดีดีไอ
- หมายเลข 3 เอพีไอด้านความคล้ายคลึงกันของเว็บเซอร์วิซทำการจับคู่เว็บเซอร์วิซต้นทางและเว็บเซอร์วิซปลายทางในทุก ๆ คู่ โดยเว็บเซอร์วิซที่จะถูกประมวลผลจะถูกส่ง (Publish) ไปยังคิวของข้อความ (Message Queue) ชื่อ Queue_in สำหรับการประมวลผล
- หมายเลข 4 เอพีไอด้านความคล้ายคลึงกันของเว็บเซอร์วิซทำการดึง (Subscribe) คู่เว็บเซอร์วิซต้นทางและเว็บเซอร์วิซปลายทางที่ประมวลผลเสร็จแล้วจากคิวของข้อความ (Message Queue) ชื่อ Queue_out ส่งกลับไปยังเว็บเซอร์วิซเบราว์เซอร์

เอพีไอด้านความคล้ายคลึงกันของเว็บเซอร์วิซมี 2 เอพีไอ ที่ให้เว็บเซอร์วิซเบราว์เซอร์เรียกใช้งานดังหัวข้อต่อไปนี้

### 4.3.1 เอพีไอสำหรับการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกัน

การค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันสามารถเรียกใช้งานได้ผ่านทางฟังก์ชันที่ชื่อ SearchSimilarityWebService ดังภาพที่ 4.5




ภาพที่ 4.5 วิธีการเรียกใช้งานเอพีไอสำหรับการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกัน

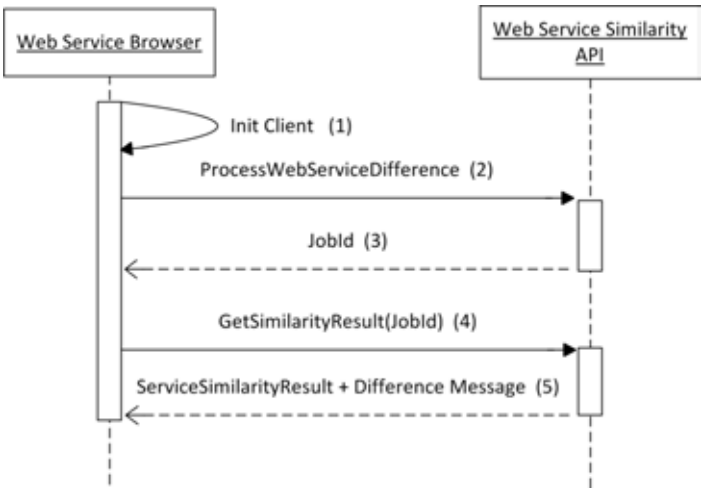
จากภาพที่ 4.5 อธิบายการทำงานได้ดังนี้

- หมายเลข 1 เว็บเซอร์วิซเบราว์เซอร์สร้างตัวเชื่อมต่อไปยังเอพีไอด้านความคล้ายคลึงกันของเว็บเซอร์วิซ
- หมายเลข 2 เว็บเซอร์วิซเบราว์เซอร์เรียกฟังก์ชันสำหรับการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันไปยังเอพีไอด้านความคล้ายคลึงกันของเว็บเซอร์วิซผ่านทางฟังก์ชัน SearchSimilarityWebService โดยส่งพารามิเตอร์ดังนี้
    1) เอกสารวิสเดิลของเว็บเซอร์วิซต้นทางที่ต้องการค้นหาเว็บเซอร์วิซที่มีความคล้ายคลึงกัน
    2) วิธีการให้คะแนนความคล้ายคลึงของเว็บเซอร์วิซสามารถเลือกได้ 2 แบบ คือ ความคล้ายคลึงกันในเชิงโครงสร้างและความคล้ายคลึงกันในเชิงความหมาย
    3) อัลกอริทึมในการให้คะแนนความคล้ายคลึงของเว็บเซอร์วิซสามารถเลือกได้ 2 แบบคือ เอ็ม-ยูอาร์บีอี และ ยูอาร์บีอี
    4) ค่าน้ำหนัก (Weight) ต่าง ๆ สำหรับใช้คำนวณและให้คะแนนความคล้ายคลึง

- หมายเลข 3 เมื่อเรียก SearchSimilarityWebService แล้วจะได้ค่า Job Id กลับมาซึ่งเป็นค่าอ้างอิง (Reference) ที่ใช้สำหรับติดต่อกับเอพีไอด้านความคล้ายคลึงกันของเว็บเซอร์วิซในครั้งต่อไป
- หมายเลข 4 เว็บเซอร์วิซเบราว์เซอร์เรียกฟังก์ชัน GetSimilarityResult สำหรับเรียกผลการให้คะแนนความคล้ายคลึงกันของเว็บเซอร์วิซโดยส่งค่า Job Id เป็นพารามิเตอร์สำหรับฟังก์ชันนี้
- หมายเลข 5 ฟังก์ชัน GetSimilarityResult คืนผลการให้คะแนนความคล้ายคลึงกลับมาในตัวแปรชื่อ ServiceSimilarityResult

### 4.3.2 เอพีไอสำหรับเรียกดูผลการวิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิซ

วิธีการเรียกดูผลการวิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิซเป็นแนวทางเดียวกันกับวิธีการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกัน ซึ่งสามารถเรียกใช้งานได้ผ่านทางฟังก์ชันที่ชื่อ ProcessWebServiceDifference ดังภาพที่ 4.6




ภาพที่ 4.6 วิธีการเรียกใช้งานเอพีไอสำหรับเรียกดูผลการวิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิซ

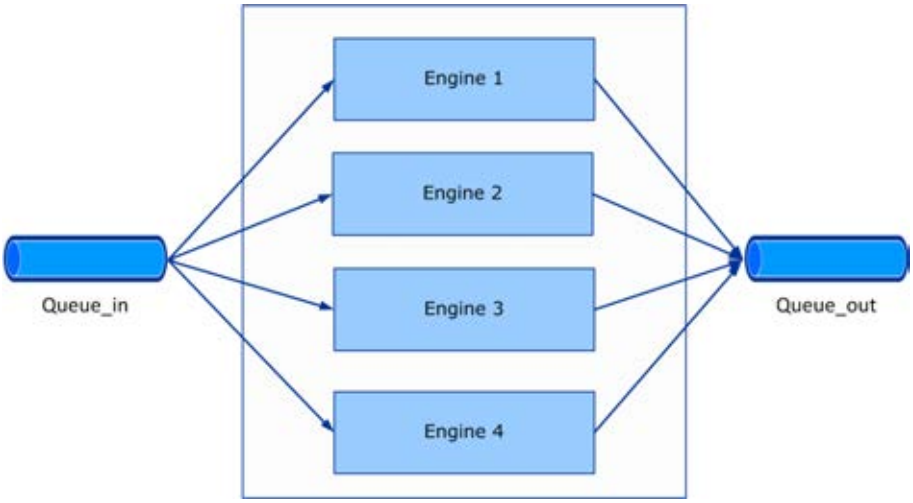
จากภาพที่ 4.6 อธิบายการทำงานได้ดังนี้

- หมายเลข 1 เว็บเซอร์วิซเบราว์เซอร์สร้างตัวเชื่อมต่อไปยังเอพีไอด้านความคล้ายคลึงกันของเว็บเซอร์วิซ

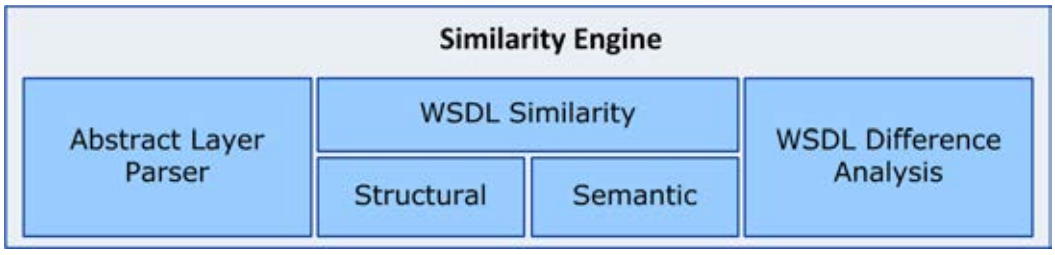
- หมายเลข 2 เว็บเซอร์วิซเบราว์เซอร์เรียกฟังก์ชันสำหรับการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันไปยังเอพีไอด้านความคล้ายคลึงกันของเว็บเซอร์วิซผ่านทางฟังก์ชัน ProcessWebServiceDifference โดยส่งพารามิเตอร์ดังนี้
    1) เอกสารวิสเดิลของเว็บเซอร์วิซต้นทางที่ต้องการเรียกดูผลการวิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิซ
    2) เอกสารวิสเดิลของเว็บเซอร์วิซปลายทางที่ต้องการเรียกดูผลการวิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิซ
    3) วิธีการให้คะแนนความคล้ายคลึงของเว็บเซอร์วิซสามารถเลือกได้ 2 แบบ คือ ความคล้ายคลึงกันในเชิงโครงสร้างและความคล้ายคลึงกันในเชิงความหมาย
    4) อัลกอริทึมในการให้คะแนนความคล้ายคลึงของเว็บเซอร์วิซสามารถเลือกได้ 2 แบบคือ เอ็ม-ยูอาร์บีอี และ ยูอาร์บีอี
    5) ค่าน้ำหนัก (Weight) ต่าง ๆ สำหรับใช้คำนวณและให้คะแนนความคล้ายคลึง
- หมายเลข 3 เมื่อเรียก ProcessWebServiceDifference แล้วจะได้ค่า Job Id กลับมาซึ่งเป็นค่าอ้างอิง (Reference) ที่ใช้สำหรับติดต่อกับเอพีไอด้านความคล้ายคลึงกันของเว็บเซอร์วิซในครั้งต่อไป
- หมายเลข 4 เว็บเซอร์วิซเบราว์เซอร์เรียกฟังก์ชัน GetSimilarityResult สำหรับเรียกผลการให้คะแนนความคล้ายคลึงกันของเว็บเซอร์วิซโดยส่งค่า Job Id เป็นพารามิเตอร์สำหรับฟังก์ชันนี้
- หมายเลข 5 ฟังก์ชัน GetSimilarityResult คืนผลการให้คะแนนความคล้ายคลึงกลับมาในตัวแปรชื่อ ServiceSimilarityResult และได้ผลการวิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิซ

### 4.4 เครื่องประมวลความคล้ายคลึง

เมื่อ Queue_in ได้รับคู่เว็บเซอร์วิซต้นทางและเว็บเซอร์วิซปลายทาง จะมีกลุ่มของเครื่องประมวลความคล้ายคลึง (Similarity Engine) ทำการสมัครรับข้อความ (Subscribe) จาก Queue_in เพื่อประมวลผลในทุก ๆ คู่ของเว็บเซอร์วิซต้นทางและเว็บเซอร์วิซปลายทางที่เข้าคิวมาดังภาพที่ 4.7 โดยมีโปรเซส (Process) Engine 1-4 ทำการประมวลผลพร้อม ๆกันเมื่อมีข้อความถูกส่งเข้าคิวมา หลังจากนั้นผลการให้คะแนนความคล้ายคลึงจะถูกส่งไปยัง Queue_out ส่วนประกอบต่าง ๆ ภายในเครื่องประมวลความคล้ายคลึงอธิบายได้ดังภาพที่ 4.8




ภาพที่ 4.7 กลุ่มของเครื่องประมวลความคล้ายคลึง Engine 1 – 4

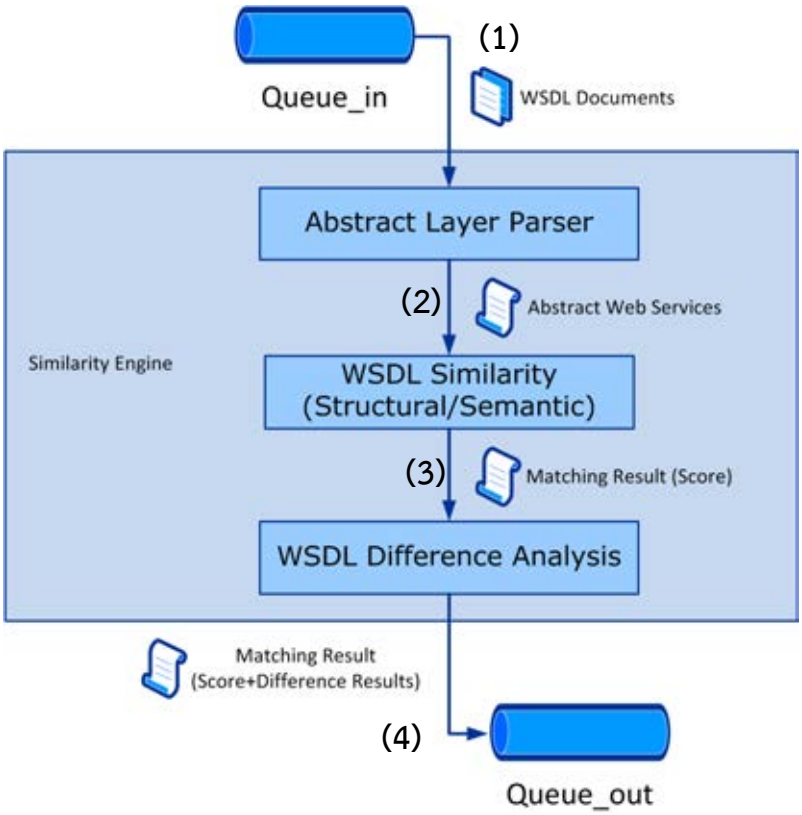



ภาพที่ 4.8 ส่วนประกอบต่าง ๆ ภายในเครื่องประมวลความคล้ายคลึง

จากภาพที่ 4.8 เครื่องประมวลความคล้ายคลึงประกอบไปด้วยส่วนประกอบต่าง ๆดังนี้

- Abstract Layer Parser เป็นส่วนที่ทำหน้าที่แปลงเอกสารวิสเดิลให้อยู่ในรูปแบบนามธรรม
- WSDL Similarity เป็นส่วนที่ทำหน้าที่ให้คะแนนความคล้ายคลึงกันของเว็บเซอร์วิส สามารถให้คะแนนได้ในเชิงโครงสร้างและเชิงความหมาย
- WSDL Difference Analysis ทำหน้าที่วิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิสต้นทางและเว็บเซอร์วิสปลายทาง

กระบวนการการประมวลผลความคล้ายคลึงกันของเว็บเซอร์วิสโดยข้อมูลที่ได้จาก Queue_in อธิบายได้ดังภาพที่ 4.9




ภาพที่ 4.9 กระบวนการการให้คะแนนความคล้ายคลึงกันของเว็บเซอร์วิสและการรายงานความแตกต่างระหว่างเว็บเซอร์วิส

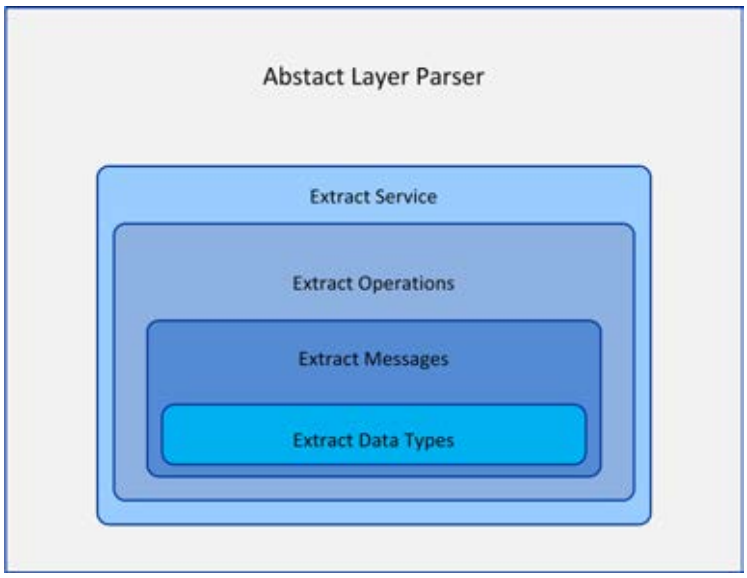
จากภาพที่ 4.9 แสดงกระบวนการการให้คะแนนความคล้ายคลึงกันของเว็บเซอร์วิซและการรายงานความแตกต่างระหว่างเว็บเซอร์วิซซึ่งสามารถอธิบายในส่วนประกอบต่าง ๆ ตามหมายเลขได้ดังนี้

- หมายเลข 1 เอกสารวิสเดิลที่เป็นคู่ของเว็บเซอร์วิซต้นทางและเว็บเซอร์วิซปลายทางที่เข้าคิวใน Queue_in ถูกส่งเข้ามาแปลงให้อยู่ในรูปแบบของเว็บเซอร์วิซเชิงนามธรรมโดย Abstract Layer Parser
- หมายเลข 2 เว็บเซอร์วิซที่อยู่ในรูปแบบเชิงนามธรรมถูกส่งเข้ามาให้คะแนนความคล้ายคลึงใน WSDL Similarity ซึ่งเว็บเซอร์วิซเบราว์เซอร์สามารถเลือกวิธีการให้คะแนนในเชิงโครงสร้างหรือเชิงความหมายได้
- หมายเลข 3 เมื่อได้คะแนนความคล้ายคลึงแล้ว WSDL Similarity จะส่งผลการเข้าคู่ของเอลิเมนต์ระหว่างเว็บเซอร์วิซต้นทางและปลายทาง (Matching Result) ไปยัง WSDL Difference Analysis เพื่อวิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิซต้นทางและปลายทางโดยพิจารณาในทุก ๆ คู่เอลิเมนต์ใน Matching Result
- หมายเลข 4 WSDL Difference Analysis ส่งผลการเข้าคู่และรายงานความแตกต่างระหว่างเว็บเซอร์วิซต้นทางและปลายทางไปยัง Queue_out เพื่อให้เอพีไอด้านความคล้ายคลึงกันของเว็บเซอร์วิซดึงผลการประมวลผลความคล้ายคลึงและส่งคืนให้เว็บเซอร์วิซเบราว์เซอร์ต่อไป

### 4.4.1 Abstract Layer Parser

Abstract Layer Parser ทำหน้าที่แปลงเอกสารวิสเดิลที่อยู่ในรูปแบบของเอกซ์เอมแอลให้เป็นเว็บเซอร์วิซเชิงนามธรรมโดยประกอบไปด้วยส่วนต่าง ๆ ดังนี้

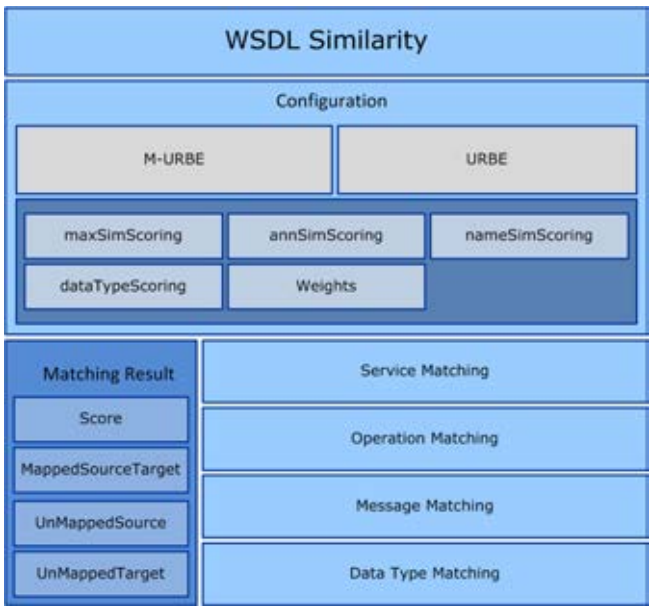
- Extract Service ทำหน้าที่แปลงเอกสารวิสเดิลโดยแปลงพอร์ตไทป์ให้เป็นเว็บเซอร์วิซเชิงนามธรรม
- Extract Operations เมื่อแปลงข้อมูลพอร์ตไทป์แล้วทำการแปลงโอเปอเรชันทั้งหมดที่อยู่ภายใต้พอร์ตไทป์ของเว็บเซอร์วิซ
- Extract Messages แปลงข้อความทั้งอินพุตและเอาต์พุตที่อยู่ภายในโอเปอเรชัน
- Extract Data Types แปลงชนิดข้อมูลทั้ง Simple Type และ Complex Type ที่อยู่ภายในข้อความ




ภาพที่ 4.10 Abstract Layer Parser

### 4.4.2 WSDL Similarity

WSDL Similarity เป็นส่วนประมวลผลหลักที่สำคัญในการให้คะแนนความคล้ายคลึงกันระหว่างเว็บเซอร์วิซต้นทางและปลายทาง ภายในประกอบไปด้วยส่วนประกอบหลาย ๆ ส่วนดังภาพที่ 4.11



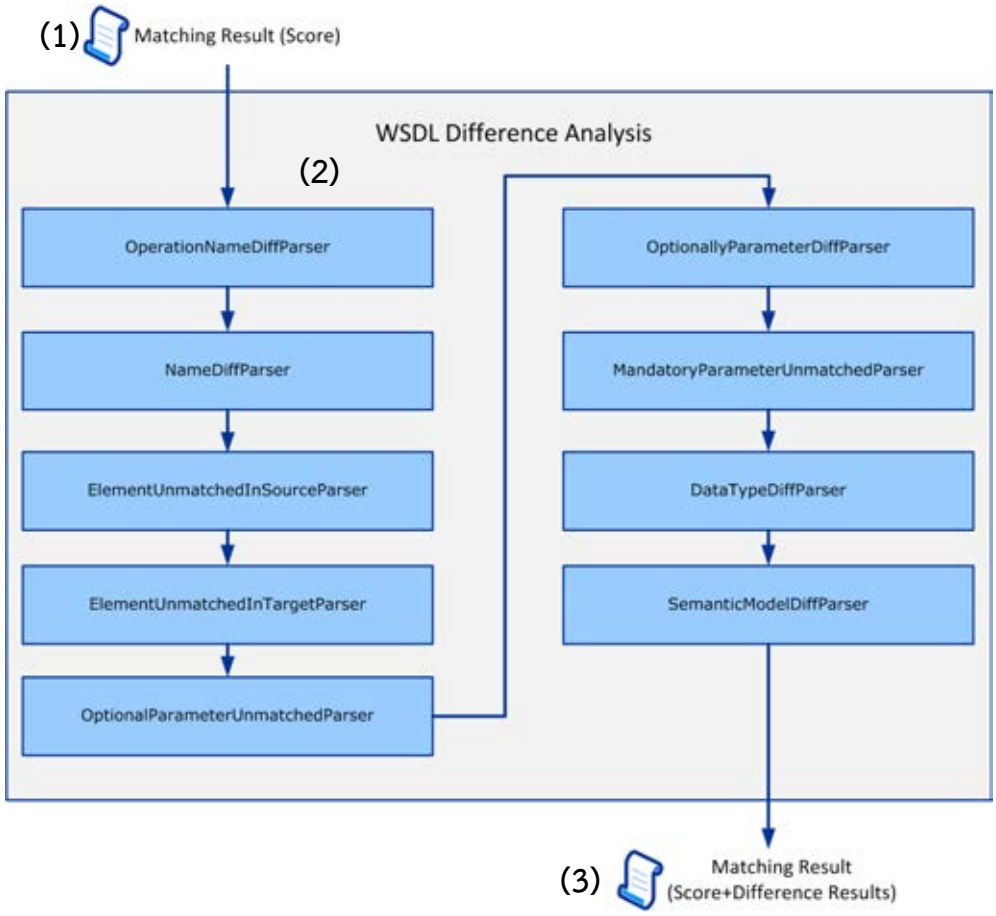

ภาพที่ 4.11 WSDL Similarity

จากภาพที่ 4.11 อธิบายส่วนประกอบต่าง ๆ ได้ดังนี้

- WSDL Similarity เป็นจุดเชื่อมต่อเพื่อรับเว็บเซอร์วิซในรูปแบบเชิงนามธรรมเข้ามาประมวลผลให้คะแนนความคล้ายคลึง
- Configuration เป็นส่วนที่ทำหน้าที่ตั้งค่าเริ่มต้นต่าง ๆ ก่อนการประมวลผลซึ่งสามารถกำหนดกระบวนการให้คะแนนความคล้ายคลึงได้ 2 แบบคือแบบเอ็ม-ยูอาร์บีอี และ แบบ ยูอาร์บีอี กระบวนการการให้คะแนนสามารถเลือกวิธีการให้คะแนนได้แก่ maxSimScoring ที่ผู้วิจัยได้ปรับปรุงในสมการ (1) annSimScoring ในสมการ (10) และ (11) nameSimScoring ในสมการ (2) และ (3) dataTypeScoring ในตารางที่ 3.2 ตารางที่ 3.3 และ ตารางที่ 3.4 และ ค่า Weights ที่กำหนดในสมการที่กล่าวไว้ในบทที่ 3
- Service Matching เป็นส่วนที่ดำเนินการการเข้าคู่และให้คะแนนความคล้ายคลึงของกลุ่มโอเปอเรชันที่อยู่ภายในเว็บเซอร์วิซที่ได้นิยามไว้ในสมการ (7)
- Operation Matching เป็นส่วนที่ดำเนินการการเข้าคู่และให้คะแนนความคล้ายคลึงในแต่ละโอเปอเรชันที่ได้นิยามในสมการ (6)
- Message Matching เป็นส่วนที่ดำเนินการการเข้าคู่และให้คะแนนความคล้ายคลึงกันของพารามิเตอร์ สามารถให้คะแนนได้ทั้งพารามิเตอร์ที่เป็นข้อมูลอินพุตและพารามิเตอร์ที่เป็นข้อมูลเอาต์พุตที่ได้นิยามในสมการ (4) และ (5)
- Data Type Matching เป็นส่วนที่ดำเนินการการเข้าคู่และให้คะแนนความคล้ายคลึงกันของชนิดข้อมูลที่ได้นิยามในภาพที่ 3.6
- Matching Result เป็นผลการให้คะแนนความคล้ายคลึงและผลการเข้าคู่ของเอลิเมนต์ภายในเว็บเซอร์วิซทั้งต้นทางและปลายทาง MappedSourceTarget คือเอลิเมนต์ที่สามารถเข้าคู่ได้ระหว่างเว็บเซอร์วิซต้นทางและปลายทาง UnMappedSource คือเอลิเมนต์ของเว็บเซอร์วิซต้นทางที่ไม่สามารถเข้าคู่ใด ๆ ในเว็บเซอร์วิซปลายทางได้ และ UnMappedTarget คือเอลิเมนต์ของเว็บเซอร์วิซปลายทางที่ไม่สามารถเข้าคู่ใด ๆ ได้จากเว็บเซอร์วิซต้นทาง ข้อมูลที่กล่าวมาใน Matching Result นี้สามารถนำไปวิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิซต้นทางและเว็บเซอร์วิซปลายทางได้ในหัวข้อถัดไป

### 4.4.3 WSDL Difference Analysis

WSDL Difference Analysis ทำหน้าที่วิเคราะห์ความแตกต่างระหว่างเว็บเซอร์วิซต้นทางและปลายทางโดยวิเคราะห์จากผลการจับคู่เปรียบเทียบที่ได้จาก WSDL Similarity (หมายเลข 1) ดังที่กล่าวไว้ก่อนหน้า และทำการส่งผลการจับคู่เปรียบเทียบไปในส่วนต่าง ๆ ภายใน WSDL Difference Analysis เพื่อทำการตรวจสอบความแตกต่างในส่วนต่าง ๆ ที่ทำการเข้าคู่ได้ (หมายเลข 2) เช่น ความแตกต่างในชื่อของโปเปอเรชัน (OperationNameDiffParser) ความแตกต่างของชนิดข้อมูล (DataTypeDiffParser) ดังภาพที่ 4.12 เมื่อผลของการเข้าคู่ประมวลผลผ่านในส่วนต่าง ๆ แล้วสุดท้ายจะได้ผลความแตกต่างระหว่างเว็บเซอร์วิซต้นทางและปลายทาง (หมายเลข 3) แต่ละส่วนภายใน WSDL Difference Analysis อธิบายได้ดังตารางที่ 4.1




ภาพที่ 4.12 WSDL Difference Analysis

**ตารางที่ 4.1 ผลการวิเคราะห์ในแต่ละส่วนประกอบของ WSDL Difference Analysis**

| ส่วนประกอบต่าง ๆ ของ WSDL Difference Analysis | คำอธิบาย | ผลลัพธ์ของการวิเคราะห์ |
|---|---|---|
| OperationNameDiffParser | สำหรับวิเคราะห์ความแตกต่างของชื่อโอเปอเรชัน | Different operation names of matched operations : <$op_q$.name> in source vs. <$op_p$.name> in target |
| NameDiffParser | สำหรับวิเคราะห์ความแตกต่างของชื่อเอลิเมนต์ (Simple Type และ Complex Type) | Different data element names of operation <$op_p$.name> : <$ele_q$.name> in source vs. <$ele_p$.name> in target |
| ElementUnmatchedInSourceParser | สำหรับวิเคราะห์เอลิเมนต์ภายในเอาต์พุตของเว็บเซอร์วิซต้นทางที่ไม่สามารถจับคู่ได้ในเว็บเซอร์วิซปลายทาง | Unmatched output data element in source : element <$op_q$.output.ele> of message <$op_q$.output> of operation <$op_q$.name> |
| ElementUnmatchedInTargetParser | สำหรับวิเคราะห์เอลิเมนต์ภายในเอาต์พุตของเว็บเซอร์วิซปลายทางที่ไม่สามารถจับคู่ได้ในเว็บเซอร์วิซต้นทาง | Unmatched output data element in target : element <$op_p$.output.ele> of message <$op_p$.output> of operation <$op_p$.name> |
| OptionalParameterUnmatchedParser | สำหรับวิเคราะห์พารามิเตอร์อินพุตที่เป็น Optional ที่ไม่เข้าคู่ | - Unmatched optional input in target : element <$op_p$.input.ele> of operation <$op_p$.name><br>- Unmatched optional input in source : element <$op_q$.input.ele> of operation <$op_q$.name> |
| OptionallyParameterDiffParser | สำหรับวิเคราะห์พารามิเตอร์อินพุตที่เป็น Optional หรือ Mandatory ซึ่งต่างจากพารามิเตอร์อินพุตของเว็บเซอร์วิซปลายทางที่เข้าคู่กัน | - Different modes of matched input element <$op_q$.input.ele> of operation <$op_q$.name> : optional in source but mandatory in target<br>- Different modes of matched input element <$op_q$.input.ele> of operation <$op_q$.name> : mandatory in source but optional in target |
| MandatoryParameterUnmatchedParser | สำหรับวิเคราะห์พารามิเตอร์อินพุตที่เป็น Mandatory ที่ไม่เข้าคู่ | - Unmatched mandatory input in target : element <$op_p$.input.ele> of operation <$op_p$.name><br>- Unmatched mandatory input in source : <$op_q$.input.ele> of operation <$op_q$.name> |
| DataTypeDiffParser | สำหรับวิเคราะห์ความแตกต่างของชนิดข้อมูล | Different data types of matched parameter message <$op_p$.input\|output> of operation <$op_p$.name> : <$op_q$.input\|output.type> in source vs. <$op_p$.input\|output.type> in target |
| SemanticModelDiffParser | สำหรับวิเคราะห์ความแตกต่างของเทอมในออนโทโลจี | Different semantic of matched data elements : <$ele_q$.modelReference> of element <$ele_q$> in source vs. <$ele_p$.modelReference> of element <$ele_p$> in target |

## 4.5 เครื่องมือที่ใช้ในการพัฒนา

ในการพัฒนาระบบค้นคืนและวิเคราะห์ความแตกต่างสำหรับเว็บเซอร์วิซที่มีความคล้ายคลึงกันผู้วิจัยได้ใช้เครื่องมือต่าง ๆ ดังนี้

1) Visual Studio 2010 ใช้สำหรับการพัฒนาระบบ
2) ภาษา C# เวอร์ชัน 4.0 ใช้งานบน .Net Framework 4.0
3) ASP.Net MVC3 สำหรับพัฒนาเว็บเซอร์วิซเบราว์เซอร์
4) jQuery 1.10.2 สำหรับพัฒนาเว็บเซอร์วิซเบราว์เซอร์
5) Windows Communication Foundation (WCF) 4.0 สำหรับสร้างเว็บเซอร์วิซและใช้งานร่วมกับ Message Queue
6) dotNetRDF 0.6.2.2180 สำหรับอ่านเอกสาร RDF ที่นิยามออนโทโลจี โดยใช้ร่วมกับภาษา SPARQL
7) WordNet.Net ใช้สำหรับติดต่อฐานข้อมูล WordNet เวอร์ชัน 3.0
8) Gallio Icarus เวอร์ชัน 3.2 และ MBUnit เวอร์ชัน 2.4 สำหรับใช้ในการทดสอบระดับหน่วยและวัดประสิทธิภาพระบบค้นคืน
9) Microsoft SQL Server 2008 สำหรับใช้ในการจัดเก็บฐานข้อมูล
10) Protégé ใช้เป็นเครื่องมือสำหรับการนิยามออนโทโลจี
11) Internet Information Services (IIS) เวอร์ชัน 7.5 ใช้สำหรับให้บริการเว็บเซอร์วิซเบราว์เซอร์

# บทที่ 5

# การทดสอบความคล้ายคลึงและวิเคราะห์ความแตกต่างของเว็บเซอร์วิซ

## 5.1 การวัดประสิทธิภาพการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกัน

เมื่อทำการปรับปรุงฟังก์ชันที่ให้คะแนนความคล้ายคลึงทั้งในเชิงโครงสร้างและเชิงความหมายแล้ว ในหัวข้อนี้จะทดสอบการค้นคืนเว็บเซอร์วิซ โดยเปรียบเทียบกับอัลกอริทึมยูอาร์บีอีของงานวิจัย [8] โดยใช้ชุดทดสอบเดียวกัน คือ SAWSDL-TC3 [27] [28] ซึ่งเป็นชุดทดสอบเว็บเซอร์วิซจำนวน 1080 เว็บเซอร์วิซ ประกอบด้วยเอกสารวิสเดิลที่ถูกกำกับความหมายด้วยเทอมในออนโทโลจีด้วยเอสเอวิสเดิล โดยในแต่ละเอกสารวิสเดิลจะอยู่ในโดเมนใดโดเมนหนึ่งใน 9 กลุ่มโดเมน ได้แก่ Communication, Economy, Education, Food, Geography, Medical, Simulation, Travel และ Weapon เอกสารวิสเดิลดังกล่าวจะถูกติดตั้งในยูดีดีไอและออนโทโลจีของชุดทดสอบที่ถูกอ้างถึงด้วยเอสเอวิสเดิลจะถูกเก็บไว้ใน Domain Specific Ontology และชุดทดสอบ SAWSDL-TC3 ได้เตรียมเว็บเซอร์วิซต้นทาง (Query) จำนวน 42 เว็บเซอร์วิซในรูปแบบเอกสารวิสเดิลให้สำหรับใช้ค้นหาเว็บเซอร์วิซที่มีความคล้ายคลึงในแต่ละโดเมน ดังภาพที่ 4.1

ในการทดสอบนี้ได้ทำการสุ่มเว็บเซอร์วิซต้นทางจากเว็บเซอร์วิซต้นทางทั้งหมดขึ้นมา 5 เว็บเซอร์วิซจากเว็บเซอร์วิซต้นทางจำนวน 42 เว็บเซอร์วิซ ซึ่งเป็นจำนวนเดียวกันกับการวัดประสิทธิภาพการค้นคืนเว็บเซอร์วิซในงานวิจัย [8]

เว็บเซอร์วิซต้นทางที่สุ่มขึ้นมาจำนวน 5 เว็บเซอร์วิซดังนี้

- 1personbicyclecar_price_service.wsdl
- bookpersoncreditcardaccount__service.wsdl
- citycountry_hotel_service.wsdl
- surfinghiking_destination_service.wsdl
- userscience-fiction-novel_price_service.wsdl

5 เว็บเซอร์วิซต้นทางดังกล่าวสามารถนำไปใช้หาเว็บเซอร์วิซปลายทางที่มีความคล้ายคลึงกันได้ทั้งในเชิงโครงสร้างและเชิงความหมายผ่านทางฟังก์ชัน *fSim* โดยกำหนดค่า Threshold ที่อยู่ในช่วงค่าตั้งแต่ 0 ถึง 1 หากคะแนนความคล้ายคลึงของเว็บเซอร์วิซที่ได้จากฟังก์ชัน *fSim* น้อยกว่าค่า Threshold เว็บเซอร์วิซปลายทางดังกล่าวจะไม่ถูกนำมาคิดค่า F-Measure [29] ซึ่งเป็นการวัด

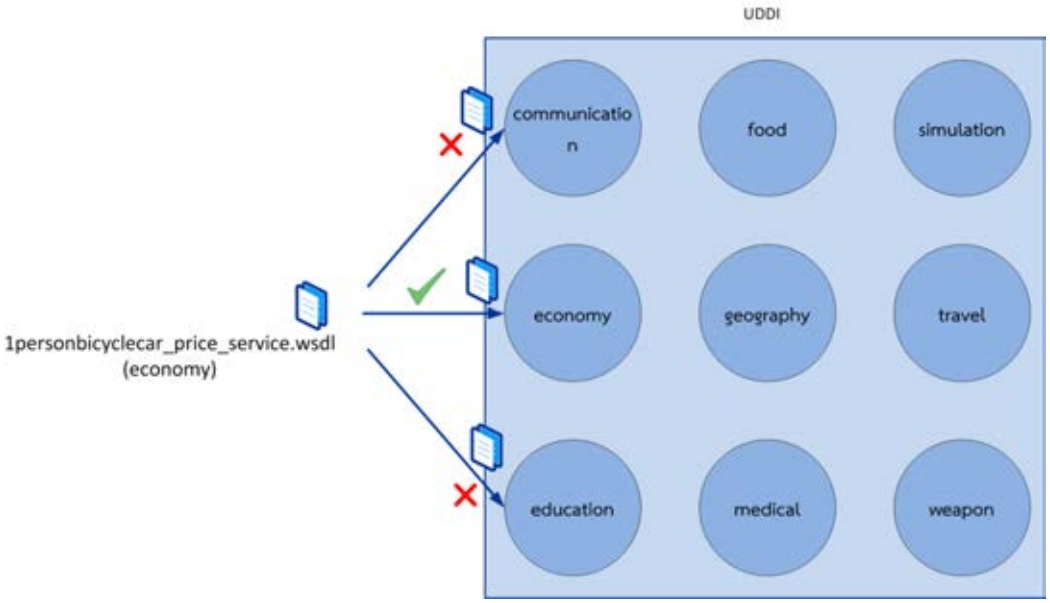
ประสิทธิภาพโดยรวมดังสมการ (14) และค่า F-Measure ในแต่ละเว็บเซอร์วิซต้นทางที่สุ่มขึ้นมาจะถูกนำมาหาค่าเฉลี่ย

$$F - Measure = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (14)$$

โดยที่ Precision = จำนวนเว็บเซอร์วิซที่เกี่ยวข้องที่ค้นคืนมาได้ / จำนวนเว็บเซอร์วิซที่ค้นคืนมาได้ทั้งหมด

Recall = จำนวนเว็บเซอร์วิซที่เกี่ยวข้องที่ค้นคืนมาได้ / จำนวนเว็บเซอร์วิซที่เกี่ยวข้องทั้งหมดในระบบ

และ เว็บเซอร์วิซที่เกี่ยวข้องคือเว็บเซอร์วิซปลายทางที่อยู่ในโดเมนเดียวกันกับเว็บเซอร์วิซต้นทางดังภาพที่ 5.1



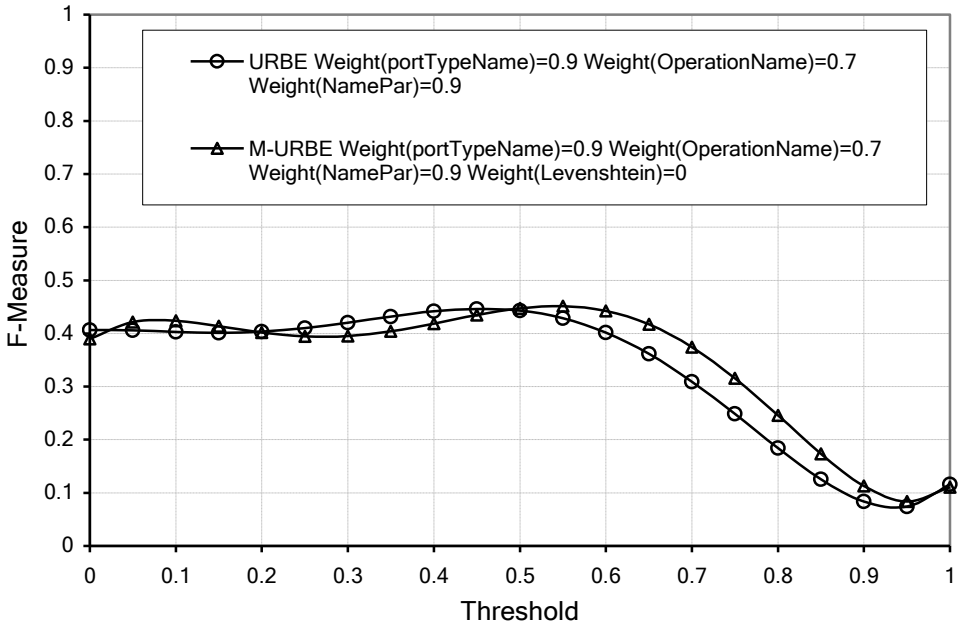

ภาพที่ 5.1 เว็บเซอร์วิซที่เกี่ยวข้องในโดเมนเดียวกัน

### 5.1.1 การวัดประสิทธิภาพการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันเชิงโครงสร้าง

การทดสอบเชิงโครงสร้างจะทำการค้นหาเว็บเซอร์วิซปลายทางและให้คะแนนความคล้ายคลึงจากเว็บเซอร์วิซต้นทางที่สุ่มขึ้นมา 5 เว็บเซอร์วิซ โดยในแต่ละเว็บเซอร์วิซต้นทาง จะทำการกำหนดค่าน้ำหนักเป็นค่าต่าง ๆ สำหรับการหาคะแนนความคล้ายคลึงจากฟังก์ชัน *fSim* ได้แก่ $Weight_{portTypeName}$ = {0.1,0.3,0.5,0.7,0.9}, $Weight_{OperationName}$ = {0.1,0.3,0.5,0.7,0.9} และ

$Weight_{NamePar}$ = {0.1,0.3,0.5,0.7,0.9} โดยกำหนดให้ ค่า $Weight_{Levenshtein}$ = 0 เพื่อทดสอบเฉพาะฟังก์ชัน *nameSim* ที่ให้คะแนนความเหมือนของคำด้วย WordNet เท่านั้น จำนวนชุดทดสอบที่สุ่มขึ้นมา 5 เว็บเซอร์วิซและแต่ละเว็บเซอร์วิซที่ถูกกำหนดด้วยค่าน้ำหนักต่าง ๆจะถูกทดสอบจำนวน 5x5x5 = 125 ครั้ง ดังนั้นจำนวนครั้งการทดสอบทั้งหมดจะเท่ากับ 125x5 = 625 ครั้ง การทดสอบอัลกอริทึมเอ็ม-ยูอาร์บีอีทำเช่นเดียวกันกับยูอาร์บีอีของงานวิจัย [8] โดยทดสอบตามค่าน้ำหนักต่าง ๆ แบบเดียวกัน

เมื่อทดสอบการให้คะแนนความคล้ายคลึงพบว่า ค่าน้ำหนักที่ทำให้ค่า F-Measure ของเอ็ม-ยูอาร์บีอีให้ผลที่ดีขึ้นกว่าของยูอาร์บีอี คือ $Weight_{portTypeName}$ = 0.9, $Weight_{OperationName}$ = 0.7 $Weight_{NamePar}$ = 0.9 จากนั้นทำการทดสอบอีกครั้งโดยใช้ค่าน้ำหนักที่ให้ผลดีขึ้นดังกล่าวเป็นค่าคงที่และทำการกำหนดค่า $Weight_{Levenshtein}$ = {0,0.1,0.3,0.5,0.7,0.9} เพื่อทดสอบว่าการให้คะแนนความเหมือนของรูปคำมีผลให้ค่า F-Measure ดีขึ้นหรือไม่ ผลการทดสอบค่า F-Measure ที่ดีที่สุดใน ทุก ๆ Threshold ยังคงไม่เปลี่ยนแปลงในทางที่ดีขึ้นคือยังเป็นค่า $Weight_{portTypeName}$ = 0.9, $Weight_{OperationName}$ = 0.7 $Weight_{NamePar}$=0.9 และ $Weight_{Levenshein}$ = 0 ดังภาพที่ 5.2

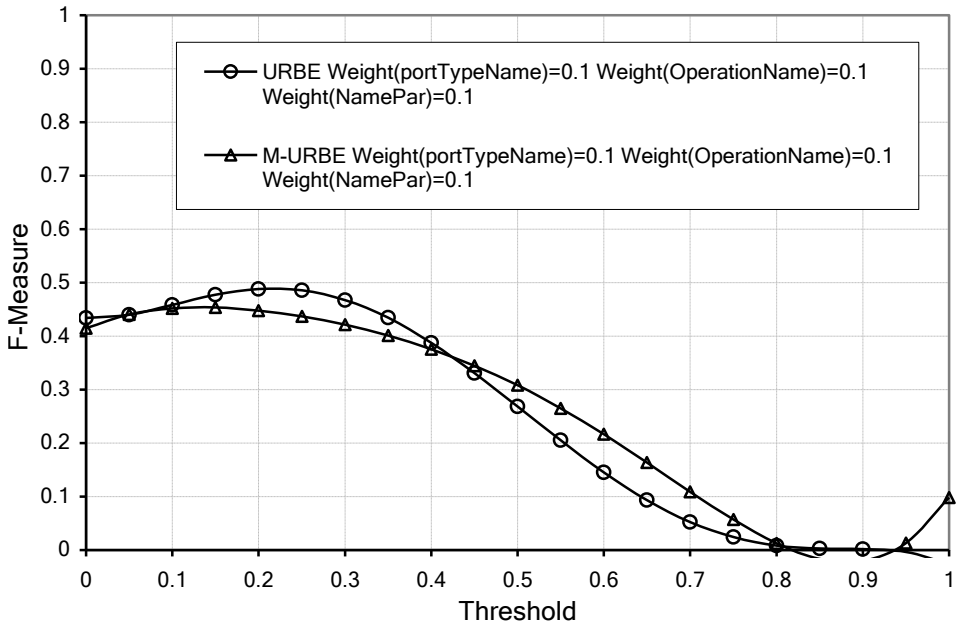



ภาพที่ 5.2 ผลการวัดประสิทธิภาพการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันเชิงโครงสร้าง

### 5.1.2 การวัดประสิทธิภาพการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันเชิงความหมาย

การทดสอบเชิงความหมายใช้แนวทางเดียวกันกับการทดสอบเชิงโครงสร้าง โดยทำการกำหนดค่าน้ำหนักเป็นค่าต่าง ๆ สำหรับการหาคะแนนความคล้ายคลึงจากฟังก์ชัน *fSim* และไม่มีการ

นำ $Weight_{Levenshtein}$ มาคิด เนื่องจากใช้ฟังก์ชัน *annSim* ให้คะแนนความเหมือนในเทอมของออนโทโลจี แทนที่การให้คะแนนความเหมือนของคำจากฟังก์ชัน *nameSim* เมื่อทดสอบการให้คะแนนความคล้ายคลึง ผลการทดสอบค่า F-Measure ที่ดีที่สุดใน ทุก ๆ Threshold ของเอ็ม-ยูอาร์บีอีให้ผลที่ดีขึ้นกว่ายูอาร์บีอี ที่ค่า $Weight_{portTypeName}$ = 0.1 $Weight_{OperationName}$ = 0.1 $Weight_{NamePar}$= 0.1 ดังภาพที่ 5.3




ภาพที่ 5.3 ผลการวัดประสิทธิภาพการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันเชิงความหมาย

### 5.1.3 สรุปผลการทดสอบ

จากการทดสอบในหัวข้อที่ 5.1.1 และ 5.1.2 การปรับปรุงการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันในเชิงโครงสร้างและเชิงความหมายทำให้ค่า F-Measure ซึ่งเป็นค่าประสิทธิภาพโดยรวมของอัลกอริทึมเอ็ม-ยูอาร์บีอีมีค่าที่สูงกว่าอัลกอริทึมยูอาร์บีอีในช่วง Threshold 0.5 ถึง 0.8 หมายความว่าเมื่อผู้ใช้ต้องการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันค่อนข้างมากโดยการกรองเว็บเซอร์วิซที่มีคะแนนความคล้ายคลึงสูงด้วยค่า Threshold โดยรวมแล้วระบบสามารถค้นคืนเว็บเซอร์วิซที่เกี่ยวข้องได้อย่างแม่นยำและได้จำนวนมากกว่าอัลกอริทึมยูอาร์บีอี ส่วนในการปรับปรุงการให้คะแนนความคล้ายคลึงของรูปคำด้วย Levenshtein Distance นั้น ไม่ส่งผลต่อการทดสอบดังกล่าว เนื่องจากว่าในชุดทดสอบ ภายในเอกสารวิสเดิลอาจมีคำที่มีความหมายสมบูรณ์และสามารถค้นหาได้ใน WordNet อยู่แล้ว

## 5.2 การทดสอบการค้นคืนเว็บเซอร์วิซโดยกรณีต่าง ๆ

การทดสอบหัวข้อนี้แสดงความแตกต่างระหว่างอัลกอริทึมเอ็ม-ยูอาร์บีอีและอัลกอริทึมยูอาร์บีอี เพื่อให้เห็นถึงความแตกต่างของคะแนนความคล้ายคลึงกันของเว็บเซอร์วิซ ผู้วิจัยได้ใช้ค่าน้ำหนักที่ดีที่สุดที่ทำให้เอ็ม-ยูอาร์บีอีนี้มีค่าประสิทธิภาพโดยรวมหรือ F-Measure ดีกว่ายูอาร์บีอี โดยจะทำการสร้างเว็บเซอร์วิซตัวอย่างที่แตกต่างกัน 2 เว็บเซอร์วิซลงในยูดีดีไอและใช้ตัวอย่างเว็บเซอร์วิซต้นทางทำการค้นคืนเว็บเซอร์วิซผ่านทางเว็บเซอร์วิซเบราว์เซอร์ตามหัวข้อดังนี้

### 5.2.1 การทดสอบการค้นคืนเว็บเซอร์วิซโดยพิจารณาเชิงโครงสร้าง

การทดสอบกรณีการค้นคืนเว็บเซอร์วิซโดยพิจารณาเชิงโครงสร้างใช้ค่าน้ำหนักที่ได้จากการทดลองในหัวข้อ 5.1.1 ได้แก่ $Weight_{portTypeName}$ = 0.9 $Weight_{OperationName}$ = 0.7 $Weight_{NamePar}$ = 0.9 และ $Weight_{WordNet}$ = 1

ผู้วิจัยได้สร้างตัวอย่างเว็บเซอร์วิซต้นทางชื่อ PolicyServiceSoap สำหรับใช้ในการค้นหาเว็บเซอร์วิซปลายทางดังภาพที่ 5.4

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
...
targetNamespace="http://tempuri.org/">
<wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
    <s:element name="SearchPolicy">
        <s:complexType>
            <s:sequence>
                <s:element name="request" type="tns:PolicyRequest"/>
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:complexType name="PolicyRequest">
        <s:sequence>
            <s:element name="PolicyNumber" type="s:string"/>
            <s:element name="ReferenceNumber" type="s:int"/>
        </s:sequence>
    </s:complexType>
    <s:element name="SearchPolicyResponse">
        <s:complexType>
            <s:sequence>
                <s:element name="SearchPolicyResult" type="tns:PolicyItem"/>
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:complexType name="PolicyItem">
        <s:sequence>
            <s:element name="PolicyNumber" type="s:string"/>
            <s:element name="PolicyHolder" type="s:string"/>
            <s:element name="Age" type="s:int"/>
        </s:sequence>
    </s:complexType>
</s:schema>
</wsdl:types>
<wsdl:message name="SearchPolicySoapIn">
    <wsdl:part name="parameters" element="tns:SearchPolicy"/>
</wsdl:message>
<wsdl:message name="SearchPolicySoapOut">
    <wsdl:part name="parameters" element="tns:SearchPolicyResponse"/>
</wsdl:message>
<wsdl:portType name="PolicyServiceSoap">
    <wsdl:operation name="SearchPolicy">
        <wsdl:input message="tns:SearchPolicySoapIn"/>
        <wsdl:output message="tns:SearchPolicySoapOut"/>
    </wsdl:operation>
</wsdl:portType>
...
</wsdl:definitions>
```

ภาพที่ 5.4 ตัวอย่างเว็บเซอร์วิซต้นทางที่ใช้ในการการทดสอบในเชิงโครงสร้าง

ตามที่ผู้วิจัยได้นำเสนอแนวทางการปรับปรุงในการให้คะแนนความคล้ายคลึงกันของเว็บเซอร์วิซเชิงโครงสร้างคือ ปรับปรุงการประเมินความคล้ายคลึงเชิงโครงสร้างในส่วนของการให้คะแนนการเข้าคู่ของชนิดข้อมูลของพารามิเตอร์ซึ่งอยู่ในกลุ่มเดียวกัน (Integer และ Real) และพิจารณาจากความเข้ากันได้ในแบบ Contravariant Input/Covariant Output เพราะฉะนั้นจะได้เว็บเซอร์วิซปลายทางตัวอย่างสำหรับทดสอบการให้คะแนนความคล้ายคลึงกันได้แก่ ContractInquiryService1 และ ContractInquiryService2 ดังภาพที่ 5.5 และ ภาพที่ 5.6

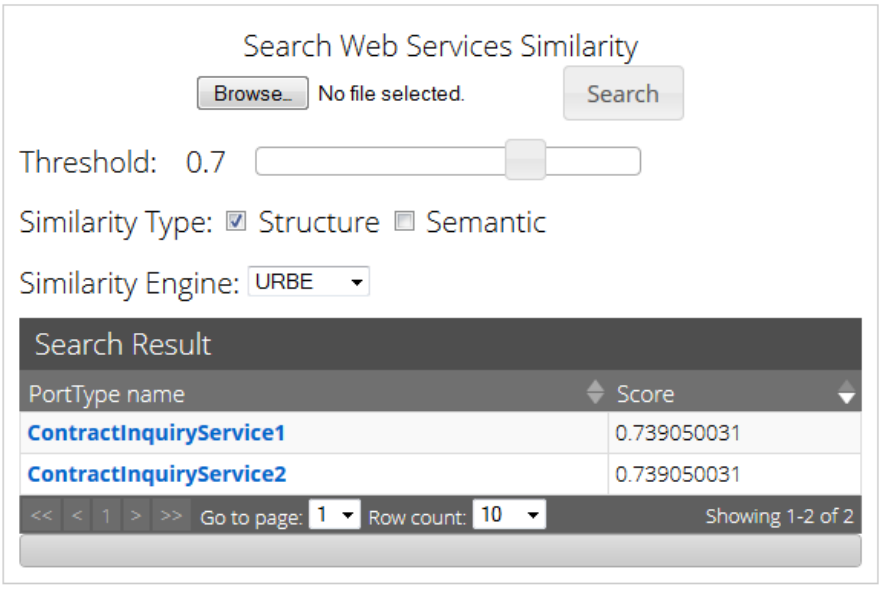
```
<?xml version="1.0"encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
...
targetNamespace="http://tempuri.org/">
<wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
    <s:element name="SearchPolicy">
        <s:complexType>
            <s:sequence>
                <s:element name="contractRequest" type="tns:PolicyRequest"/>
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:complexType name="PolicyRequest">
        <s:sequence>
            <s:element name="PolicyNumber" type="s:string"/>
            <s:element name="ReferenceNumber" type="s:short"/>
        </s:sequence>
    </s:complexType>
    <s:element name="SearchPolicyResponse">
        <s:complexType>
            <s:sequence>
                <s:element name="SearchPolicyResult" type="tns:PolicyItem"/>
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:complexType name="PolicyItem">
        <s:sequence>
            <s:element name="PolicyNumber" type="s:string"/>
            <s:element name="PolicyHolder" type="s:string"/>
            <s:element name="Age" type="s:long"/>
        </s:sequence>
    </s:complexType>
</s:schema>
</wsdl:types>
<wsdl:message name="SearchPolicySoapIn">
    <wsdl:part name="parameters" element="tns:SearchPolicy"/>
</wsdl:message>
<wsdl:message name="SearchPolicySoapOut">
    <wsdl:part name="parameters" element="tns:SearchPolicyResponse"/>
</wsdl:message>
<wsdl:portType name="ContractInquiryService1">
    <wsdl:operation name="InquiryContract">
        <wsdl:input message="tns:SearchPolicySoapIn"/>
        <wsdl:output message="tns:SearchPolicySoapOut"/>
    </wsdl:operation>
</wsdl:portType>
...
</wsdl:definitions>
```

ภาพที่ 5.5 ตัวอย่างเว็บเซอร์วิซปลายทาง 1

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
...
targetNamespace="http://tempuri.org/">
<wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
    <s:element name="SearchPolicy">
        <s:complexType>
            <s:sequence>
                <s:element name="contractRequest" type="tns:PolicyRequest"/>
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:complexType name="PolicyRequest">
        <s:sequence>
            <s:element name="PolicyNumber" type="s:string"/>
            <s:element name="ReferenceNumber" type="s:long"/>
        </s:sequence>
    </s:complexType>
    <s:element name="SearchPolicyResponse">
        <s:complexType>
            <s:sequence>
                <s:element name="SearchPolicyResult" type="tns:PolicyItem"/>
            </s:sequence>
        </s:complexType>
    </s:element>
    <s:complexType name="PolicyItem">
        <s:sequence>
            <s:element name="PolicyNumber" type="s:string"/>
            <s:element name="PolicyHolder" type="s:string"/>
            <s:element name="Age" type="s:short"/>
        </s:sequence>
    </s:complexType>
</s:schema>
</wsdl:types>
<wsdl:message name="SearchPolicySoapIn">
    <wsdl:part name="parameters" element="tns:SearchPolicy"/>
</wsdl:message>
<wsdl:message name="SearchPolicySoapOut">
    <wsdl:part name="parameters" element="tns:SearchPolicyResponse"/>
</wsdl:message>
<wsdl:portType name="ContractInquiryService2">
    <wsdl:operation name="InquiryContract">
        <wsdl:input message="tns:SearchPolicySoapIn"/>
        <wsdl:output message="tns:SearchPolicySoapOut"/>
    </wsdl:operation>
</wsdl:portType>
...
</wsdl:definitions>
```

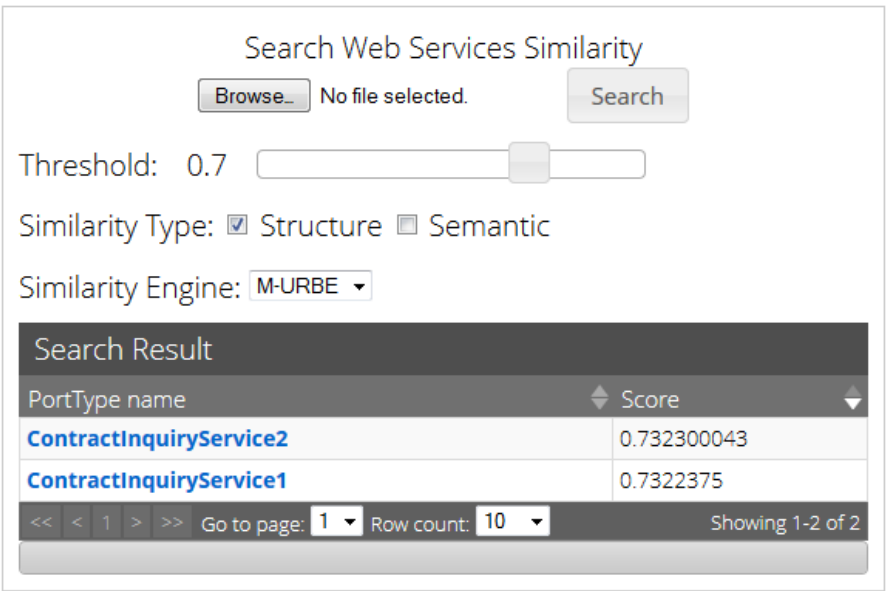
ภาพที่ 5.6 ตัวอย่างเว็บเซอร์วิซปลายทาง 2

การทดลองนี้แสดงให้เห็นถึงความแตกต่างของผลการค้นคืนผ่านเว็บเซอร์วิซเบราว์เซอร์ เมื่อใช้อัลกอริทึมเอ็ม-ยูอาร์บีอีและยูอาร์บีอี โดยขั้นตอนแรกจะค้นคืนเว็บเซอร์วิซโดยใช้อัลกอริทึมยูอาร์บีอีจะได้ผลค้นคืนดังภาพที่ 5.7



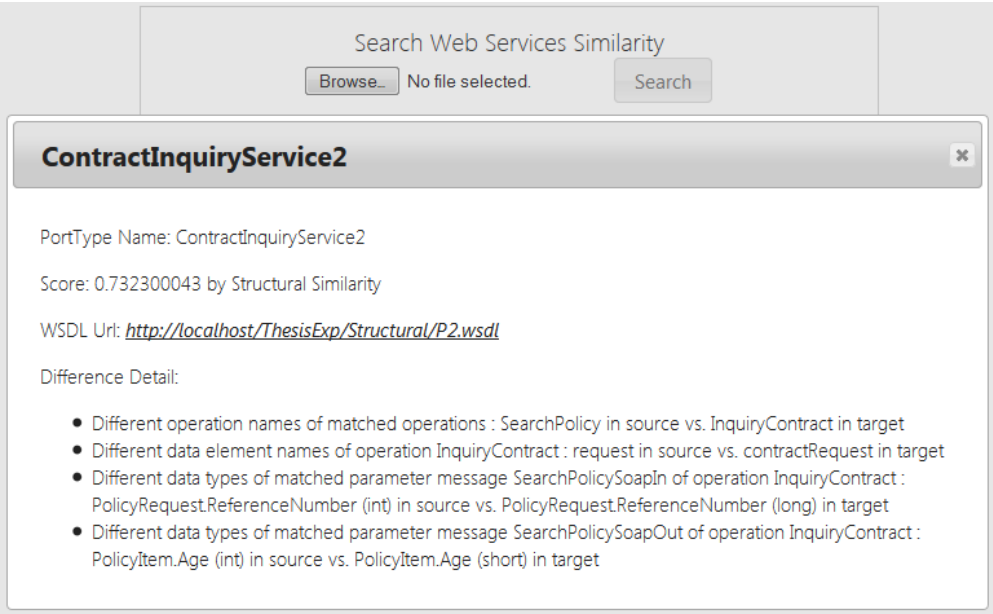

ภาพที่ 5.7 ผลการค้นคืนเว็บเซอร์วิซด้วยความคล้ายคลึงเชิงโครงสร้างโดยใช้อัลกอริทึมยูอาร์บีอี

จากภาพที่ 5.7 จะเห็นว่าเว็บเซอร์วิซตัวอย่างทั้งสองได้คะแนนความคล้ายคลึงเชิงความหมายเท่ากันคือ 0.739050031 นั่นคือผู้ใช้งานมีโอกาสตัดสินใจเลือกใช้เว็บเซอร์วิซเท่ากัน ซึ่งหากใช้อัลกอริทึมของเอ็ม-ยูอาร์บีอีจะได้ผลลัพธ์ดังภาพที่ 5.8




ภาพที่ 5.8 ผลการค้นคืนเว็บเซอร์วิซด้วยความคล้ายคลึงเชิงโครงสร้างโดยใช้อัลกอริทึมเอ็ม-ยูอาร์บีอี

จากภาพที่ 5.8 แสดงให้เห็นว่าเว็บเซอร์วิซ PortType ชื่อ ContractInquiryService2 มีคะแนนความคล้ายคลึงเชิงโครงสร้างมากกว่า ContractInquiryService1 เนื่องจากใน ContractInquiryService1 ชนิดข้อมูล ComplexType ชื่อ PolicyRequest.ReferenceNumber มีชนิดข้อมูลเป็น short ซึ่งอยู่ในกลุ่มชนิดข้อมูล Integer และ PolicyRequest เป็นข้อมูลอินพุตจะได้คะแนนเข้าคู่ของชนิดข้อมูลเมื่อเทียบกับเว็บเซอร์วิซต้นทางคือ PolicyRequest.ReferenceNumber ที่มีชนิดข้อมูลเป็น int เท่ากับ 0.5 เมื่อใช้ตารางที่ 3.3 ในการให้คะแนน ซึ่งเป็นคะแนนที่น้อยกว่ากรณีของ ContractInquiryService2 ซึ่งชนิดข้อมูล PolicyRequest.ReferenceNumber มีชนิดข้อมูลเป็น long และเมื่อใช้ตารางที่ 3.3 จะได้คะแนนการเข้าคู่เท่ากับ 1 นอกจากนี้ข้อมูลเอาต์พุตใน ContractInquiryService1 คือ PolicyItem.Age มีชนิดข้อมูลเป็น long อยู่ในกลุ่มชนิดข้อมูล Integer จะได้คะแนนเข้าคู่ชนิดข้อมูลเมื่อเทียบกับเว็บเซอร์วิซต้นทางคือ PolicyItem.Age ซึ่งมีชนิดข้อมูลเป็น int เท่ากับ 0.5 เมื่อใช้ตารางที่ 3.3 ในการให้คะแนน ซึ่งเป็นคะแนนที่น้อยกว่ากรณีของ ContractInquiryService2 ซึ่งชนิดข้อมูล PolicyItem.Age มีชนิดข้อมูลเป็น short และเมื่อใช้ตารางที่ 3.3 จะได้คะแนนการเข้าคู่เท่ากับ 1 ดังนั้นแล้ว คะแนนความคล้ายคลึงของ ContractInquiryService2 เมื่อคำนวณโดยค่าน้ำหนักดังที่กล่าวมา จึงมีความคล้ายคลึงกันกับเว็บเซอร์วิซต้นทางมากกว่า ContractInquiryService1 และสามารถแสดงรายละเอียดของเว็บเซอร์วิซ PortType ContractInquiryService2 รวมถึงรายงานผลความแตกต่างได้ดังภาพที่ 5.9




ภาพที่ 5.9 รายละเอียดของเว็บเซอร์วิซ PortType ContractInquiryService2 จากการค้นคืนด้วยการพิจารณาเชิงโครงสร้าง

### 5.2.2 การทดสอบการค้นคืนเว็บเซอร์วิซโดยพิจารณาเชิงความหมาย

การทดสอบกรณีการค้นคืนเว็บเซอร์วิซโดยพิจารณาเชิงความหมายใช้ค่าน้ำหนักที่ได้จากการทดลองให้หัวข้อ 5.1.2 ได้แก่ $Weight_{portTypeName}$ = 0.1 $Weight_{OperationName}$ = 0.1 $Weight_{NamePar}$ = 0.1 และทำการสร้างออนโทโลจีสำหรับใช้ในการกำกับความหมายดังภาพที่ 5.10

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
<!ENTITY owl "http://www.w3.org/2002/07/owl#">
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
]>
<rdf:RDF xmlns="http://127.0.0.1/ontology/insurance.owl#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xmlns:rdf="http://www.w3.org/1999/02/22-
rdf-syntax-ns#" xml:base="http://127.0.0.1/ontology/insurance.owl">
<owl:Ontology rdf:about="http://127.0.0.1/ontology/insurance.owl"/>
<!--

////////////////////////////////////////////////////////////////////////////////
/
    //
    // Classes
    //

////////////////////////////////////////////////////////////////////////////////
/
     -->
<!-- http://127.0.0.1/ontology/insurance.owl#Contract -->
<owl:Class rdf:about="http://127.0.0.1/ontology/insurance.owl#Contract"/>
<!-- http://127.0.0.1/ontology/insurance.owl#ContractRequest -->
<owl:Class rdf:about="http://127.0.0.1/ontology/insurance.owl#ContractRequest">
<rdfs:subClassOf rdf:resource="http://127.0.0.1/ontology/insurance.owl#Request"/>
</owl:Class>
<!-- http://127.0.0.1/ontology/insurance.owl#PersonPolicy -->
<owl:Class rdf:about="http://127.0.0.1/ontology/insurance.owl#PersonPolicy">
<rdfs:subClassOf rdf:resource="http://127.0.0.1/ontology/insurance.owl#Policy"/>
</owl:Class>
<!-- http://127.0.0.1/ontology/insurance.owl#Policy -->
<owl:Class rdf:about="http://127.0.0.1/ontology/insurance.owl#Policy">
<rdfs:subClassOf rdf:resource="http://127.0.0.1/ontology/insurance.owl#Contract"/>
</owl:Class>
<!-- http://127.0.0.1/ontology/insurance.owl#PolicyRequest -->
<owl:Class rdf:about="http://127.0.0.1/ontology/insurance.owl#PolicyRequest">
<rdfs:subClassOf
rdf:resource="http://127.0.0.1/ontology/insurance.owl#ContractRequest"/>
</owl:Class>
<!-- http://127.0.0.1/ontology/insurance.owl#Request -->
<owl:Class rdf:about="http://127.0.0.1/ontology/insurance.owl#Request"/>
</rdf:RDF>
<!-- Generated by the OWL API (version 3.4.2) http://owlapi.sourceforge.net -->
```

ภาพที่ 5.10 ตัวอย่างออนโทโลจีสำหรับการทดสอบการค้นคืนเว็บเซอร์วิซโดยพิจารณาเชิงความหมาย

เมื่อได้ตัวอย่างออนโทโลจีแล้ว เว็บเซอร์วิซต้นทางจึงทำการกำกับความหมายด้วยเทอมในออนโทโลจีดังภาพที่ 5.11

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
...
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:sawsdl="http://www.w3.org/ns/sawsdl" targetNamespace="http://tempuri.org/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
      <s:element name="SearchPolicy">
        <s:complexType>
          <s:sequence>
            <s:element name="request" type="tns:PolicyRequest"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="PolicyRequest"
sawsdl:modelReference="http://127.0.0.1/ontology/insurance.owl#ContractRequest">
        <s:sequence>
          <s:element name="PolicyNumber" type="s:string"/>
          <s:element name="ReferenceNumber" type="s:int"/>
        </s:sequence>
      </s:complexType>
      <s:element name="SearchPolicyResponse">
        <s:complexType>
          <s:sequence>
            <s:element name="SearchPolicyResult" type="tns:PolicyItem"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="PolicyItem"
sawsdl:modelReference="http://127.0.0.1/ontology/insurance.owl#Policy">
        <s:sequence>
          <s:element name="PolicyNumber" type="s:string"/>
          <s:element name="PolicyHolder" type="s:string"/>
          <s:element name="Age" type="s:int"/>
        </s:sequence>
      </s:complexType>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="SearchPolicySoapIn">
    <wsdl:part name="parameters" element="tns:SearchPolicy"/>
  </wsdl:message>
  <wsdl:message name="SearchPolicySoapOut">
    <wsdl:part name="parameters" element="tns:SearchPolicyResponse"/>
  </wsdl:message>
  <wsdl:portType name="PolicyServiceSoap">
    <wsdl:operation name="SearchPolicy">
      <wsdl:input message="tns:SearchPolicySoapIn"/>
      <wsdl:output message="tns:SearchPolicySoapOut"/>
    </wsdl:operation>
  </wsdl:portType>
...
</wsdl:definitions>
```

ภาพที่ 5.11 ตัวอย่างเว็บเซอร์วิซต้นทางที่ทำการกำกับความหมายด้วยเทอมในออนโทโลจีตัวอย่าง

เช่นเดียวกันกับแนวทางการปรับปรุงเชิงโครงสร้าง ในการพิจารณาความคล้ายคลึงกันของเว็บเซอร์วิซในเชิงความหมายผู้วิจัยได้พิจารณาจากความเข้ากันได้ในแบบ Contravariant Input/Covariant Output เช่นกัน เพราะฉะนั้นจะได้เว็บเซอร์วิซปลายทางตัวอย่างสำหรับทดสอบการให้คะแนนความคล้ายคลึงกันคือ ContractInquiryService1 และ ContractInquiryService2 ดังภาพที่ 5.12 และ ภาพที่ 5.13

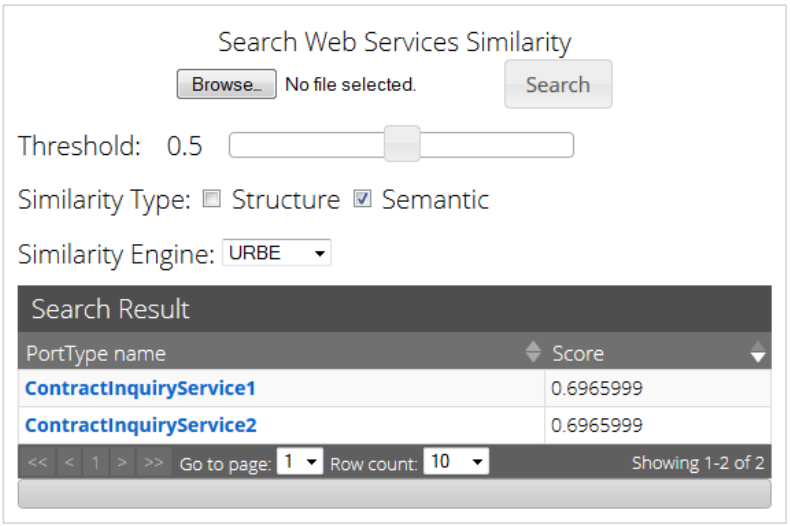
```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
...
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:sawsdl="http://www.w3.org/ns/sawsdl" targetNamespace="http://tempuri.org/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
      <s:element name="SearchPolicy">
        <s:complexType>
          <s:sequence>
            <s:element name="request" type="tns:PolicyRequest"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="PolicyRequest"
sawsdl:modelReference="http://127.0.0.1/ontology/insurance.owl#PolicyRequest">
        <s:sequence>
          <s:element name="PolicyNumber" type="s:string"/>
          <s:element name="ReferenceNumber" type="s:int"/>
        </s:sequence>
      </s:complexType>
      <s:element name="SearchPolicyResponse">
        <s:complexType>
          <s:sequence>
            <s:element name="SearchPolicyResult" type="tns:PolicyItem"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="PolicyItem"
sawsdl:modelReference="http://127.0.0.1/ontology/insurance.owl#Contract">
        <s:sequence>
          <s:element name="PolicyNumber" type="s:string"/>
          <s:element name="PolicyHolder" type="s:string"/>
          <s:element name="Age" type="s:int"/>
        </s:sequence>
      </s:complexType>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="SearchPolicySoapIn">
    <wsdl:part name="parameters" element="tns:SearchPolicy"/>
  </wsdl:message>
  <wsdl:message name="SearchPolicySoapOut">
    <wsdl:part name="parameters" element="tns:SearchPolicyResponse"/>
  </wsdl:message>
  <wsdl:portType name="ContractInquiryService1">
    <wsdl:operation name="InquiryContract">
      <wsdl:input message="tns:SearchPolicySoapIn"/>
      <wsdl:output message="tns:SearchPolicySoapOut"/>
    </wsdl:operation>
  </wsdl:portType>
...
</wsdl:definitions>
```

ภาพที่ 5.12 ตัวอย่างเว็บเซอร์วิซปลายทางที่กำกับความหมายด้วยเทอมในออนโทโลจี 1

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
...
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:sawsdl="http://www.w3.org/ns/sawsdl" targetNamespace="http://tempuri.org/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
      <s:element name="SearchPolicy">
        <s:complexType>
          <s:sequence>
            <s:element name="request" type="tns:PolicyRequest"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="PolicyRequest"
sawsdl:modelReference="http://127.0.0.1/ontology/insurance.owl#Request">
        <s:sequence>
          <s:element name="PolicyNumber" type="s:string"/>
          <s:element name="ReferenceNumber" type="s:int"/>
        </s:sequence>
      </s:complexType>
      <s:element name="SearchPolicyResponse">
        <s:complexType>
          <s:sequence>
            <s:element name="SearchPolicyResult" type="tns:PolicyItem"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="PolicyItem"
sawsdl:modelReference="http://127.0.0.1/ontology/insurance.owl#PersonPolicy">
        <s:sequence>
          <s:element name="PolicyNumber" type="s:string"/>
          <s:element name="PolicyHolder" type="s:string"/>
          <s:element name="Age" type="s:int"/>
        </s:sequence>
      </s:complexType>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="SearchPolicySoapIn">
    <wsdl:part name="parameters" element="tns:SearchPolicy"/>
  </wsdl:message>
  <wsdl:message name="SearchPolicySoapOut">
    <wsdl:part name="parameters" element="tns:SearchPolicyResponse"/>
  </wsdl:message>
  <wsdl:portType name="ContractInquiryService2">
    <wsdl:operation name="InquiryContract">
      <wsdl:input message="tns:SearchPolicySoapIn"/>
      <wsdl:output message="tns:SearchPolicySoapOut"/>
    </wsdl:operation>
  </wsdl:portType>
...
</wsdl:definitions>
```

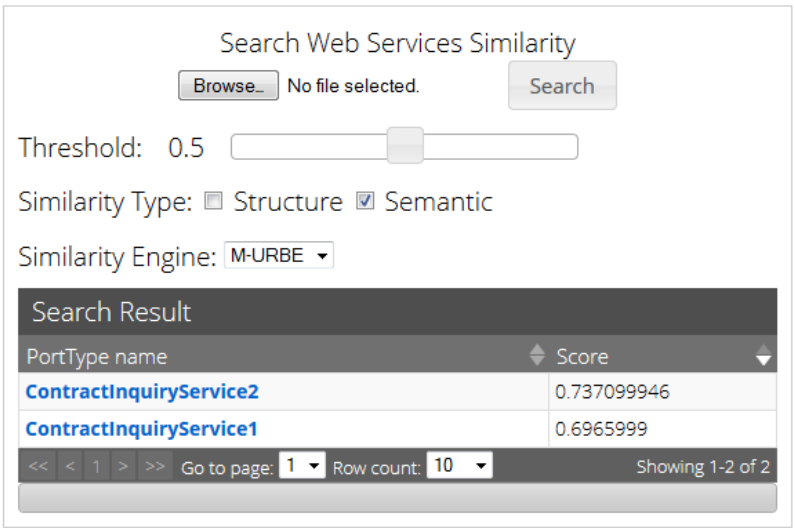
ภาพที่ 5.13 ตัวอย่างเว็บเซอร์วิซปลายทางที่กำกับความหมายด้วยเทอมในออนโทโลจี 2

เช่นเดียวกันกับการทดลองการค้นคืนเว็บเซอร์วิซในเชิงโครงสร้าง ที่แสดงให้เห็นถึงความแตกต่างของผลการค้นคืนผ่านเว็บเซอร์วิซเบราว์เซอร์เมื่อใช้อัลกอริทึมเอ็ม-ยูอาร์บีอีและยูอาร์บีอี โดยขั้นตอนแรกจะค้นคืนเว็บเซอร์วิซโดยใช้อัลกอริทึมยูอาร์บีอี จะได้ผลค้นคืนดังภาพที่ 5.14



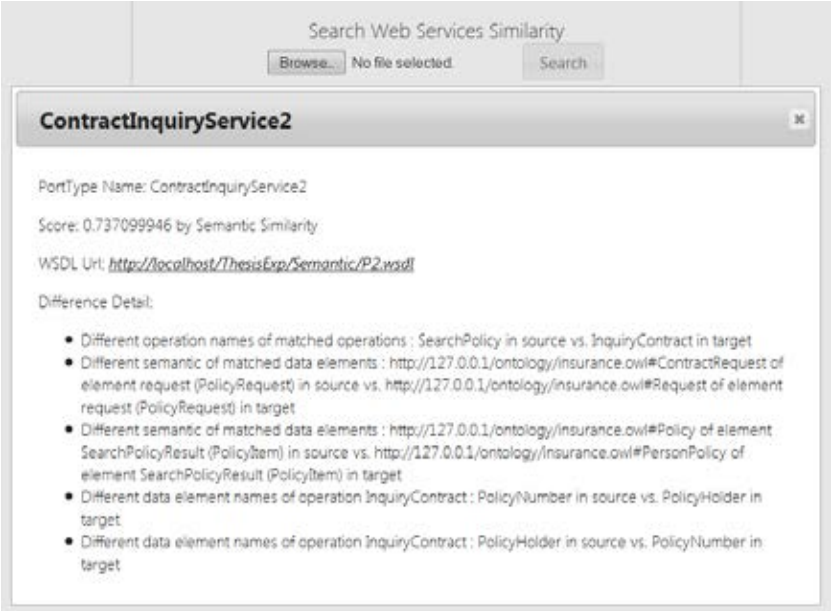

ภาพที่ 5.14 ผลการค้นคืนเว็บเซอร์วิซด้วยความคล้ายคลึงเชิงความหมายโดยใช้อัลกอริทึมยูอาร์บีอี

จากภาพที่ 5.14 จะเห็นว่าเว็บเซอร์วิซตัวอย่างทั้งสองได้คะแนนความคล้ายคลึงเชิงความหมายเท่ากันคือ 0.6965999 นั่นคือผู้ใช้งานมีโอกาสตัดสินใจเลือกใช้เว็บเซอร์วิซเท่ากัน ซึ่งหากใช้อัลกอริทึมเอ็ม-ยูอาร์บีอี จะได้ผลลัพธ์ดังภาพที่ 5.15




ภาพที่ 5.15 ผลการค้นคืนเว็บเซอร์วิซด้วยความคล้ายคลึงเชิงความหมายโดยใช้อัลกอริทึมเอ็ม-ยูอาร์บีอี

จากภาพที่ 5.15 แสดงให้เห็นว่าเว็บเซอร์วิซ PortType ชื่อ ContractInquiryService2 มีคะแนนความคล้ายคลึงเชิงความหมายมากกว่า ContractInquiryService1 เนื่องจากใน ContractInquiryService1 ชนิดข้อมูล ComplexType ชื่อ PolicyRequest ได้กำกับความหมายด้วยเทอมในออนโทโลจีด้วยคลาส PolicyRequest ซึ่งเมื่อให้คะแนน pathSim สำหรับพารามิเตอร์อินพุตในสมการ (10) โดยเทียบกับเว็บเซอร์วิซต้นทางที่กำกับความหมายด้วยเทอมในออนโทโลจีด้วยคลาส ContractRequest จึงพบว่าเว็บเซอร์วิซปลายทางดังกล่าวถูกกำกับด้วยเทอมออนโทโลจีที่มีความเฉพาะเจาะจงกว่าจึงทำให้ได้คะแนนเท่ากับ 0.5 (1/(1+1)) ซึ่งได้คะแนนน้อยกว่า ContractInquiryService2 ซึ่งมีชนิดข้อมูล PolicyRequest ที่ถูกกำกับความหมายด้วยเทอมในออนโทโลจีไปยังคลาส Request ดังนั้นเมื่อเทียบกับคลาส ContractRequest ที่กำกับโดยเว็บเซอร์วิซต้นทางจึงมีความทั่วไปมากกว่าจึงทำให้คะแนนเท่ากับ 1 นอกจากนี้ในพารามิเตอร์เอาต์พุตของ ContractInquiryService1 ชนิดข้อมูล ComplexType ชื่อ PolicyItem ได้กำกับความหมายด้วยเทอมในออนโทโลจีด้วยคลาส Contract ซึ่งเมื่อให้คะแนน pathSim สำหรับพารามิเตอร์เอาต์พุตในสมการ (11) โดยเทียบกับเว็บเซอร์วิซต้นทางที่กำกับความหมายด้วยเทอมในออนโทโลจีไปยังคลาส Policy ทำให้เว็บเซอร์วิซปลายทางดังกล่าวมีความทั่วไปกว่าจึงทำให้ได้คะแนนเท่ากับ 0.5 (1/(1+1)) ซึ่งได้คะแนนน้อยกว่า ContractInquiryService2 ที่ PolicyItem ที่ถูกกำกับความหมายด้วยคลาส PersonPolicy จึงทำให้ได้คะแนนเท่ากับ 1 และเมื่อคำนวณโดยค่าน้ำหนักดังที่กล่าวมา จึงมีความคล้ายคลึงกันเชิงความหมายกับเว็บเซอร์วิซต้นทางมากกว่า ContractInquiryService1 และสามารถแสดงรายละเอียดของเว็บเซอร์วิซ PortType ContractInquiryService2 รวมถึงรายงานผลความแตกต่างได้ดังภาพที่ 5.16




ภาพที่ 5.16 รายละเอียดของเว็บเซอร์วิซ PortType ContractInquiryService2 จากการค้นคืนด้วยการพิจารณาเชิงความหมาย

### 5.2.3 สรุปผลการทดสอบ

การทดสอบในหัวข้อ 5.2.1 และ 5.2.2 แสดงให้เห็นถึงประโยชน์ของการปรับปรุงอัลกอริทึม ซึ่งสามารถให้คะแนนความคล้ายคลึงกันของเว็บเซอร์วิซที่มีความคล้ายคลึงกันมากที่สุดตามหลักการที่ได้นำมาปรับปรุงอัลกอริทึมดังกล่าว โดยสามารถแยกแยะความแตกต่างเชิงความหมายระหว่างเว็บเซอร์วิซปลายทางต่าง ๆ ได้อย่างละเอียดมากขึ้น

# บทที่ 6

# บทสรุป

## 6.1 สรุปผลการวิจัย

งานวิจัยนี้ได้แสดงถึงวิธีการปรับปรุงการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันทั้งในเชิงโครงสร้างและเชิงความหมายโดยปรับปรุงอัลกอริทึมยูอาร์บีอีของ Plebani และ Pernici [8] ซึ่งผู้วิจัยเห็นว่าอัลกอริทึมดังกล่าวมีความครอบคลุมในการให้คะแนน ความคล้ายคลึงกันของเว็บเซอร์วิซแต่ยังมีข้อจำกัดต่าง ๆ ในอัลกอริทึมได้แก่ (1) การพิจารณาการให้คะแนนความคล้ายคลึงในเชิงโครงสร้างในส่วนของการพิจารณาชนิดข้อมูลของพารามิเตอร์อินพุตและเอาต์พุต ซึ่งการเปรียบเทียบชนิดข้อมูลนั้น ทำในระดับกลุ่มโดยไม่สนใจความแตกต่างของชนิดข้อมูล ทำให้การพิจารณาความคล้ายคลึงของพารามิเตอร์ที่อยู่ในกลุ่มชนิดเดียวกัน เช่น กลุ่มชนิดตัวเลขจำนวนเต็ม (Integer) ด้วยกัน (คือ long, int, short, byte) หรืออยู่ในกลุ่มชนิดตัวเลขทศนิยม (Real) ด้วยกัน (คือ decimal, double, float) จะได้คะแนนความคล้ายคลึงเท่ากัน (2) การพิจารณาความคล้ายคลึงของชื่อไม่ได้พิจารณาในส่วนของความคล้ายคลึงของรูปคำ และ (3) การเปรียบเทียบพารามิเตอร์ทั้งอินพุตและเอาต์พุตในเชิงโครงสร้างและเชิงความหมาย ใช้การเปรียบเทียบความคล้ายคลึงกันเท่านั้นโดยไม่ได้พิจารณาหลักการความเข้ากันได้ (Compatibility) แบบ Contravariant Input/Covariant Output ผู้วิจัยได้เสนออัลกอริทึมเอ็ม-ยูอาร์บีอีเพื่อปรับปรุงข้อจำกัดดังกล่าว โดยเมื่อเทียบกับอัลกอริทึมยูอาร์บีอี การปรับปรุงทำให้ค่า F-Measure ซึ่งบ่งบอกประสิทธิภาพโดยรวมของอัลกอริทึมมีค่าสูงกว่าอัลกอริทึมยูอาร์บีอีในช่วง Threshold 0.5 ถึง 0.8 ซึ่งหมายความว่าเมื่อผู้ใช้ต้องการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันค่อนข้างมากโดยการกรองเว็บเซอร์วิซที่มีคะแนนความคล้ายคลึงสูงด้วยค่า Threshold โดยรวมแล้วระบบสามารถค้นคืนเว็บเซอร์วิซที่เกี่ยวข้องได้อย่างแม่นยำและได้จำนวนมากกว่าอัลกอริทึมยูอาร์บีอี ส่วนในการปรับปรุงการให้คะแนนความคล้ายคลึงของรูปคำด้วย Levenshtein Distance นั้น ไม่ส่งผลต่อการทดสอบเนื่องจากว่าในชุดทดสอบ ภายในเอกสารวิสเดิลอาจมีคำที่มีความหมายสมบูรณ์และสามารถค้นหาได้ใน WordNet อยู่แล้ว

ผู้วิจัยได้พัฒนาระบบการค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันในเชิงโครงสร้างและเชิงความหมาย นอกจากนี้ยังได้เพิ่มส่วนของการวิเคราะห์ความแตกต่างกันระหว่างเว็บเซอร์วิซอีกด้วย และวิธีประเมินความคล้ายคลึงกันของเว็บเซอร์วิซในงานวิจัยนี้ใช้กับวิวัฒนาการของเว็บเซอร์วิซของผู้ให้บริการรายหนึ่ง ๆ เพื่อดูความคล้ายคลึงของเว็บเซอร์วิซต่างเวอร์ชันกันได้

## 6.2 ปัญหาและข้อจำกัดที่พบจากการวิจัย

ปัญหาและข้อจำกัดที่พบจากการวิจัยมีดังนี้

1) การให้คะแนนความคล้ายคลึงกันของชนิดข้อมูลในฟังก์ชัน *datatypeSim* เป็นการพิจารณาชนิดข้อมูลบางส่วนในข้อกำหนดเอกซ์เอ็มแอลสกีมาเท่านั้น โดยไม่ได้พิจารณาชนิดข้อมูลทั้งหมด ยกตัวอย่างเช่น ไม่ได้พิจารณาชนิดข้อมูลที่เป็น anyType และ anyURI เป็นต้น
2) การให้คะแนนการเข้าคู่ชนิดข้อมูลใน ตารางที่ 3.2 ที่ไม่ใช่การพิจารณาลำดับชั้นของข้อมูล (GNA) มีผลต่อการค้นคืนเว็บเซอร์วิซด้วยเช่นกัน ในที่นี้ผู้วิจัยใช้คะแนนการเข้าคู่ตาม [8] แต่อาจมีค่าที่เหมาะสมกว่าในการให้คะแนนความคล้ายคลึง ซึ่งจะส่งผลต่อประสิทธิภาพโดยรวมของการค้นคืนเว็บเซอร์วิซ
3) อัลกอริทึมฟังก์ชันการให้คะแนนการเข้าคู่ความคล้ายคลึงกันของชนิดข้อมูลในภาพที่ 3.6 เป็นการให้คะแนนการเข้าคู่อย่างง่าย โดยการให้คะแนนความคล้ายคลึงกันของ Complex Type เอลิเมนต์ในเว็บเซอร์วิซต้นทางและเว็บเซอร์วิซปลายทางต้องเป็น Complex Type ทั้งคู่ แต่อาจมีในบางกรณีที่ชนิดข้อมูลที่เป็นแบบ Simple Type สามารถเข้าคู่ได้กับ Complex Type ได้โดยไม่จำเป็นต้องเป็นชนิดข้อมูลแบบ Complex Type ทั้งคู่ได้
4) การให้คะแนนการเข้าคู่ความคล้ายคลึงกันของชื่อในฟังก์ชัน *nameSim* มีข้อจำกัดของจำนวนคำศัพท์ในฐานข้อมูล WordNet ซึ่งสามารถให้คะแนนความคล้ายคลึงกันของชื่อได้ก็ต่อเมื่อชื่อนั้นอยู่ในฐานข้อมูล WordNet
5) การประมวลผลในระบบค้นคืนเว็บเซอร์วิซที่มีความคล้ายคลึงกันจะใช้เวลานานโดยขนาดของเอกสารวิสเดิลมีผลต่อเวลาโดยตรง เนื่องจากต้องใช้เวลาในการพิจารณาการให้คะแนนความคล้ายคลึงในส่วนต่าง ๆ ของเอกสารวิสเดิล

## 6.3 ข้อเสนอแนะ

ผู้วิจัยมีข้อเสนอแนะในบางส่วนที่งานวิจัยนี้ไม่ได้ศึกษา และสามารถพัฒนาเพิ่มเติมในอนาคตได้ดังนี้

1) เพิ่มการพิจารณาการให้คะแนนความคล้ายคลึงกันของชนิดข้อมูลในฟังก์ชัน *datatypeSim* โดยวิเคราะห์ความเข้ากันได้ของชนิดข้อมูลให้ครบทุกประเภทตามข้อกำหนดเอกซ์เอ็มแอลสกีมา
2) ปรับปรุงอัลกอริทึม *datatypeSim* ให้สามารถให้คะแนนความคล้ายคลึงกันระหว่าง Simple Type และ Complex Type
3) เพิ่มความสามารถการให้คะแนนความคล้ายคลึงกันของชื่อในฟังก์ชัน *nameSim* ให้สามารถใช้คำศัพท์จากฐานข้อมูลชนิดอื่นที่ไม่ใช่ WordNet ได้
4) อัลกอริทึมการจับคู่เปรียบเทียบในทุก ๆ เอลิเมนต์ที่เป็นไปได้มีความสัมพันธ์ระหว่างเวลาและขนาดของเอลิเมนต์เท่ากับ $O(n^2)$ ซึ่งสามารถใช้ฮิวริสติก (Heuristics) ในการจับคู่ เช่น พิจารณาพารามิเตอร์ที่ชื่อเหมือนกัน เพื่อกรองพารามิเตอร์ที่ชื่อแตกต่างกันออกไปในขั้นแรก เพื่อลดจำนวนคู่ที่ต้องพิจารณาความเหมือนในด้านอื่นในลำดับต่อไป

# รายการอ้างอิง


[1] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., Web Services Description Language (WSDL) 1.1 [Online]. 2001. Available from: http://www.w3.org/TR/wsdl [2013, July]

[2] Chinnici, R., Moreau, J.-J., Ryman, A., and Weerawarana, S., Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language [Online]. 2007. Available from: http://www.w3.org/TR/wsdl20/ [2013, July]

[3] Simple Object Access Protocol (SOAP) 1.1 [Online]. 2000. Available from: http://www.w3.org/TR/2000/NOTE-SOAP-20000508/ [2013, July]

[4] Extensible Markup Language (XML) 1.1 (Second Edition) [Online]. 2006. Available from: http://www.w3.org/TR/xml11/ [2013, July]

[5] Universal Description, Discovery, and Integration of Web Services (UDDI) [Online]. Available from: http://uddi.xml.org/ [2013, July]

[6] Hatzi, O., Batistatos, G., Nikolaidou, M., and Anagnostopoulos, D., A Specialized Search Engine for Web Service Discovery. Proceedings of Web Services (ICWS), 2012 IEEE 19th International Conference on, Honolulu, HI, USA, pp. 448-455, 2012.

[7] Elgazzar, K., Hassan, A.E., and Martin, P., Clustering WSDL Documents to Bootstrap the Discovery of Web Services. Proceedings of Web Services (ICWS), 2010 IEEE International Conference on, Miami, FL, pp. 147-154, 2010.

[8] Plebani, P. and Pernici, B., URBE: Web Service Retrieval Based on Similarity Evaluation. Knowledge and Data Engineering, IEEE Transactions on, pp. 1629-1642, 2009.

[9] Stroulia, E. and Wang, Y., Structural and semantic matching for assessing web-service similarity. International Journal of Cooperative Information Systems, vol. 14, pp. 407–437, 2005.

[10] Liu, F., Shi, Y., Yu, J., Wang, T., and Wu, J., Measuring Similarity of Web Services Based on WSDL. Proceedings of Web Services (ICWS), 2010 IEEE International Conference on, Miami, FL, pp. 155-162, 2010.

[11] Castagna, G., Covariance and contravariance: conflict without a cause. ACM Transactions on Programming Languages and Systems (TOPLAS), vol. 3, pp. 431-447, May 1995.

[12] Pompan, N. and Senivongse, T., Enhancing Structural and Semantic Similarity Evaluation for Web Service Retrieval. Proceedings of International Conference on Computing, Engineering and Communication Technologies (ICCECT 2013), Bangkok, Thailand, pp. 22-27, 2013.

[13] Biron, P.V., Permanente, K., and Malhotra, A., XML Schema Part 2: Datatypes Second Edition [Online]. 2004. Available from: http://www.w3.org/TR/xmlschema-2/ [2013, July]

[14] Ontology [Online]. Available from: http://semanticweb.org [2013, July]

[15] McGuinness, D.L. and Harmelen, F.v., OWL Web Ontology Language [Online]. 2004. Available from: http://www.w3.org/TR/owl-features/ [2013, July]

[16] Farrell, J. and Lausen, H., Semantic Annotations for WSDL and XML Schema [Online]. 2007. Available from: http://www.w3.org/TR/sawsdl/ [2013, July]

[17] Klein, M., Toward High-Precision Service Retrieval. IEEE Internet Computing, vol. 8, pp. 30-36, 2004.

[18] Andrikopoulos, V., Benbernou, S., and Papazoglou, M.P., On the Evolution of Services. Proceedings of Software Engineering, IEEE Transactions on, pp. 609-628, 2012.

[19] Leitner, P., Michlmayr, A., Rosenberg, F., and Dustdar, S., End-to-End Versioning Support for Web Services. Proceedings of Services Computing, 2008. SCC '08. IEEE International Conference on, 2008.

[20] Andrikopoulos, V. and Plebani, P., Retrieving Compatible Web Services. Proceedings of Web Services (ICWS), 2011 IEEE International Conference on, 2011.

[21] WordNet a lexical database for English [Online]. Available from: http://wordnet.princeton.edu/ [2013, July]

[22] Navarro, G., A guided tour to approximate string matching. ACM Computing Surveys (CSUR), vol. 33, pp. 31 - 88, 2001.

[23] Andrikopoulos, V., Benbernou, S., and Papazoglou, M.P., On the Evolution of Services. Software Engineering, IEEE Transactions on, vol. 38, pp. 609 - 628, 2012.

[24] Shen, X., Jin, X., Bie, R., and Sun, Y., MSC: A Semantic Ranking for Hitting Results of Matchmaking of Services. Proceedings of Computer Software and Applications Conference, 2006. COMPSAC '06. 30th Annual International pp. 291-296, 2006.

[25] Sriharee, N. and Senivongse, T., Matchmaking and ranking of semantic web services using integrated service profile. International Journal of Metadata, Semantics and Ontologies, vol. 1, pp. 100-118, 2006.

[26] Prud'hommeaux, E. and Seaborne, A., SPARQL Query Language for RDF [Online]. 2008. Available from: http://www.w3.org/TR/rdf-sparql-query/ [2013, July]

[27] Klusch, M. and Kapahnke, P., Semantic Web Service Selection with SAWSDL-MX. Proceedings of the Second International Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web, Karlsruhe, Germany, pp. 3-17, 2008.

[28] SemWebCentral, SAWSDL-TC [Online]. Available from: http://projects.semwebcentral.org/projects/sawsdl-tc/ [2013, August]

[29] Manning, C.D., Raghavan, P., and Schütze, H., Introduction to Information Retrieval [Online]. 2009. Available from: http://www-nlp.stanford.edu/IR-book/ [2013, August]

# ประวัติผู้เขียนวิทยานิพนธ์

นายนิติพันธ์ ป้อมพันธ์ เกิดเมื่อวันที่ 27 มกราคม พ.ศ. 2530 ที่จังหวัดอุดรธานี สำเร็จการศึกษาระดับปริญญาบัณฑิต หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ เกียรตินิยมอันดับ 1 จากมหาวิทยาลัยรังสิต ปีการศึกษา 2551 และได้เข้าศึกษาในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ณ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2554 งานวิจัยที่สนใจ ได้แก่ เว็บเซอร์วิช สถาปัตยกรรมเอสโอเอ การประมวลผลแบบกระจาย