การค้นหาเว็บเซอร์วิสเชิงความหมายด้วยข้อกำหนดอาวล์-เอสโพรเซสโมเดล

นายปิยะ สุวรรณโนภาส

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2549
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

DISCOVERING SEMANTIC WEB SERVICES WITH OWL-S PROCESS MODEL SPECIFICATIONS

Mr.Piya Suwannopas

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering Program in Computer Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2006

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การค้นหาเว็บเซอร์วิสเชิงความหมายด้วยข้อกำหนดอาวล์-เอสโพรเซส โมเดล
โดย	นายปิยะ สุวรรณโนภาส
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร.ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ

(อาจารย์ ดร.ยรรยง เต็งอำนวย)

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา)

..... กรรมการ

(อาจารย์ ดร.วิษณุ โกตรจรัส)

.....กรรมการ

(อาจารย์ ดร.เนตรนภา สีหารี)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ปิยะ สุวรรณโนภาส : การค้นหาเว็บเซอร์วิสเชิงความหมายด้วยข้อกำหนดอาวล์-เอสโพรเซสโมเดล. (DISCOVERING SEMANTIC WEB SERVICES WITH OWL-S PROCESS MODEL SPECIFICATIONS) อ. ที่ปรึกษา : ผศ. ดร.ทวีตีย์ เสนีวงศ์ ณ อยุธยา, 71 หน้า.

การค้นหาบริการเป็นประเด็นหนึ่งที่สำคัญสำหรับโมเดลสถาปัตยกรรมแบบอิงบริการ ปัจจุบันการค้นหามีแนวโน้มไปในทางการค้นหาเชิงความหมายโดยใช้คำอธิบายเชิงความหมายเป็นพื้นฐานในการเข้าคู่ของบริการแทนการใช้แอททริบิวท์อย่างง่าย อาวล์-เอสได้ถูกนำมาใช้เป็นข้อกำหนดเชิงความหมายสำหรับเว็บเซอร์วิสซึ่งประกอบด้วยโพรไฟล์ 3 โพรไฟล์ โดยหนึ่งในนั้นโพรเซสโมเดลเป็นโพรไฟล์ที่อธิบายพฤติกรรมเชิงพลวัตของเว็บเซอร์วิสในรูปแบบของแง่มุมเชิงหน้าที่และกระแสกระบวนการ และโดยทั่วไปจะใช้ในการกำหนดบริการ การประกอบบริการและการเฝ้าสังเกตบริการ งานวิจัยนี้นำเสนอวิธีการใหม่ในการใช้อาวล์-เอสโพรเซสโมเดลสำหรับการค้นหาบริการ เว็บเซอร์วิสสามารถอธิบายกระบวนการภายในด้วยอาวล์-เอสโพรเซสโมเดล เพื่อให้ผู้ใช้บริการสามารถค้นหาเว็บเซอร์วิสที่มีรายละเอียดกระบวนการบางอย่างตามที่ต้องการ การเข้าคู่ของบริการจะอยู่บนพื้นฐานของการเข้าคู่เชิงออนโทโลจีแบบยืดหยุ่นและการประเมินค่าข้อบังคับของพฤติกรรมเชิงหน้าที่และกระแสกระบวนการของเว็บเซอร์วิส โดยโครงสร้างการควบคุมในกระแสกระบวนการที่งานวิจัยนี้สนใจคือกระบวนการที่มีโครงสร้างแบบลำดับ แบบทำงานพร้อมกัน แบบตัดสินใจ และแบบทำงานวนซ้ำ งานวิจัยนี้ยังได้นำเสนอสถาปัตยกรรมการค้นหาบริการโดยใช้กระบวนการเป็นพื้นฐานอีกด้วย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา.....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต.....ปิยะ สุวรรณโนภาส.....
สาขาวิชา...วิศวกรรมคอมพิวเตอร์..... ลายมือชื่ออาจารย์ที่ปรึกษา...ทวีตีย์ เสนีวงศ์ ณ อยุธยา
ปีการศึกษา.....2549.....

4870382021: MAJOR COMPUTER ENGINEERING
KEY WORD: SERVICE DISCOVERY / SEMANTIC WEB SERVICES / OWL-S PROCESS MODEL / ONTOLOGY

PIYA SUWANNOPAS : DISCOVERING SEMANTIC WEB SERVICES WITH OWL-S PROCESS MODEL SPECIFICATIONS. THESIS ADVISOR : ASST. PROF. TWITTIE SENIVONGSE, Ph.D., 71 pp.

Service discovery is one of the crucial issues for service-oriented architectural model. Recently the trend is towards semantic discovery by which semantic descriptions are the basis for service matchmaking instead of simple search based on service attributes. OWL-S is a widely adopted semantic specification for Web Services which comprises three profiles. Among those, process model is the profile that describes dynamic bahaviour of Web Services in terms of functional aspects and process flows, and is generally aimed for service enactment, composition, and monitoring. This research presents a new approach to use OWL-S process model for service discovery purpose. A Web Service can have its internal process described as an OWL-S process model specification, and a service consumer can query for a Web Service with a particular process detail. Matchmaking will be based on flexible ontological matching and evaluation of constraints on the functional behaviour and process flow of the Web Service. The control structures in the process flows which the matchmaking considers are Sequence, Split-Join, If-Then-Else and Repeat-While. The architecture for process-based discovery is also presented.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Department.....Computer Engineering.....Student's Signature.....Piya Suwannopas
Field of Study.....Computer Engineering.....Advisor's Signature.....Twittie Senivongse
Academic Year.....2006.....

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลงด้วยความกรุณาเป็นอย่างสูงของผู้ช่วยศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา อาจารย์ที่ปรึกษาวิทยานิพนธ์ของข้าพเจ้า ที่ให้ความรู้ทั้งหลายมากไปกว่าทางด้านการเรียน อาจารย์ได้ให้คำปรึกษาดี ๆ ที่นำมาใช้ได้ในชีวิต ให้กำลังใจและแก้ปัญหาให้ข้าพเจ้ามาตลอด สิ่งต่าง ๆ ที่ข้าพเจ้าทำผิดพลาดไปข้าพเจ้ากราบขออภัย และกราบขอบพระคุณอาจารย์เป็นอย่างสูงไว้ ณ ที่นี้ด้วย

ขอขอบพระคุณ อาจารย์ ดร.ยรรยง เต็งอำนวย ประธานกรรมการสอบ อาจารย์ ดร.วิษณุ โคตรจรัส และ อาจารย์ ดร.เนตรนภา สีหารี กรรมการสอบของข้าพเจ้าที่กรุณาให้คำแนะนำทำให้งานวิจัยชิ้นนี้มีความถูกต้องและสมบูรณ์มากขึ้น และขอขอบพระคุณคำชี้แนะดี ๆ ที่จะเป็นประโยชน์ต่อการดำรงชีวิตของข้าพเจ้าต่อไปในภายภาคหน้า

ขอขอบพระคุณอาจารย์ทุกท่านที่ให้ความรู้แก่ข้าพเจ้าเพื่อใช้ในการทำงานวิจัยนี้ รวมถึงคำแนะนำดี ๆ ที่ให้แก่ข้าพเจ้าตลอดมา

ขอขอบคุณพี่ ๆ น้อง ๆ ที่ห้องปฏิบัติการวิศวกรรมระบบสารสนเทศ ที่ช่วยให้ชีวิตการทำงานวิจัยมีความสุขไม่เงียบเหงาและความอบอุ่นที่มีให้ ขอบคุณสำหรับคำแนะนำดี ๆ และกำลังใจที่ให้มาตลอด

ขอขอบพระคุณ คุณพ่อ และคุณแม่ที่เลี้ยงดูข้าพเจ้า ให้กำลังใจข้าพเจ้า ให้อภัยข้าพเจ้าตลอดมา คุณความดีของวิทยานิพนธ์ฉบับนี้ ขอมอบเป็นเครื่องบูชาคุณบิดา มารดา ไว้ ณ โอกาสนี้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ..... จ

กิตติกรรมประกาศ..... ฉ

สารบัญ..... ช

สารบัญภาพ..... ญ

สารบัญตาราง..... ฏ

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของการวิจัย 3

1.3 ขอบเขตของการวิจัย 3

1.4 ขั้นตอนการวิจัย 3

1.5 ประโยชน์ที่คาดว่าจะได้รับ 4

1.6 ผลงานตีพิมพ์ 4

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 5

2.1 แนวคิดและทฤษฎี 5

2.1.1 ยูดีดีไอ 5

2.1.2 อาวล์-เอสโพรเซสโมเดล 6

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง 7

2.2.1 การเข้าคู่และการจัดลำดับเว็บเซอร์วิสเชิงความหมายโดยใช้เซอร์วิสโพรไฟล์รวม 7

2.2.2 การค้นหาบริการเชิงความหมายด้วยออนโทโลจีกระบวนการ 8

2.2.3 การเข้าคู่เว็บเซอร์วิสเชิงความหมายด้วยข้อกำหนดบนโพรเซสโมเดล 9

บทที่ 3 การค้นหาเว็บเซอร์วิสด้วยอาวล์-เอสโพรเซสโมเดล 13

3.1 ข้อกำหนดกระบวนการ 13

3.2 เงื่อนไขในการเข้าคู่ 18

3.2.1 เงื่อนไขทางออนโทโลจี 18

3.2.2 เงื่อนไขแบบช่วงตัวเลข 18

3.2.3 เงื่อนไขทางตรรกะ 19

หน้า

3.2.4 เงื่อนไขทางรูปแบบกระบวนการ 19
3.3 ตัวอย่างการค้นหาบริการจากข้อกำหนดกระบวนการ 19
บทที่ 4 สถาปัตยกรรมการค้นหาเว็บเซอร์วิสด้วยอาวล์-เอสโพรเซสโมเดล 23
4.1 การออกแบบการทำงานของตัวกลาง 23
4.2 การออกแบบตัวกลาง 24
4.3 เครื่องมือที่ใช้พัฒนา 25
บทที่ 5 การทดสอบตัวกลางในการค้นหาเว็บเซอร์วิสด้วยกรณีศึกษา 27
5.1 กรณีศึกษาที่มีโครงสร้างกระแสงานแบบตัดสินใจ 27
5.2 กรณีศึกษาที่มีโครงสร้างกระแสงานแบบทำงานพร้อมกันและทำงานแบบลำดับ 29
5.3 กรณีศึกษาที่มีโครงสร้างกระแสงานแบบทำงานวนซ้ำและทำงานแบบลำดับ 37
บทที่ 6 สรุปผลการวิจัย 44
6.1 สรุปผลการวิจัย 44
6.2 ปัญหาและอุปสรรค 44
6.3 แนวทางการวิจัยต่อไป 44
รายการอ้างอิง 46
ภาคผนวก 48
ภาคผนวก ก ตัวอย่างอาวล์-เอสโพรเซสโมเดลที่ใช้ในการทดสอบ 49
ภาคผนวก ข ผลงานตีพิมพ์ 55
ประวัติผู้เขียนวิทยานิพนธ์ 70

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

ภาพประกอบ หน้า

รูปที่ 2.1 โครงสร้างยูดีดีไอ..... 5
รูปที่ 2.2 อัปเปอร์ออนโทโลจีของอาวล์-เอสโพรเซสโมเดล 7
รูปที่ 3.1 อาวล์-เอสโพรเซสโมเดลของส่วนแสดงพฤติกรรมเชิงหน้าที่ของบริการของธนาคารแรก 10
รูปที่ 3.2 แผนภาพแสดงกระบวนการภายในบริการเงินกู้ของธนาคารแรก 12
รูปที่ 3.3 อาวล์-เอสโพรเซสโมเดลของส่วนแสดงกระบวนการภายในของบริการของธนาคารแรก 13
รูปที่ 3.4 แผนภาพแสดงกระบวนการภายในบริการเงินกู้ของธนาคารที่สอง..... 15
รูปที่ 3.5 แผนภาพแสดงกระบวนการภายในบริการเงินกู้ของธนาคารที่สาม 15
รูปที่ 3.6 ออนโทโลจีกระบวนการที่ใช้ร่วมกันภายในโดเมนการกู้เงิน 16
รูปที่ 3.7 แฟ้มข้อมูลอาวล์-เอสที่แสดงความสัมพันธ์ระหว่างคลาสในออนโทโลจีกระบวนการของโดเมนการกู้เงินกับกระบวนการภายในบริการเงินกู้ของธนาคารแรก 17
รูปที่ 3.8 บางส่วนของโดเมนออนโทโลจีของอสังหาริมทรัพย์ 17
รูปที่ 4.1 เฟรมเวิร์กสำหรับการค้นหาบริการโดยข้อกำหนดอาวล์-เอสโพรเซสโมเดล 23
รูปที่ 4.2 แผนภาพคลาสของตัวกลางสำหรับการประกาศและค้นหาเว็บเซอร์วิส 25
รูปที่ 5.1 การลงทะเบียนบริการกู้เงินของธนาคารแรก 27
รูปที่ 5.2 การลงทะเบียนบริการกู้เงินของธนาคารที่สอง 27
รูปที่ 5.3 การลงทะเบียนบริการกู้เงินของธนาคารที่สาม 28
รูปที่ 5.4 การกรอกข้อมูลยูอาร์แอลของออนโทโลจีกระบวนการที่ใช้ร่วมกันของบริการกู้เงิน 28
รูปที่ 5.5 การกรอกข้อมูลการค้นหาบริการกู้เงิน 29
รูปที่ 5.6 ผลลัพธ์ของการค้นหาบริการกู้เงิน 30
รูปที่ 5.7 กระแสงานของบริการสั่งซื้อไวน์บริการแรก..... 30
รูปที่ 5.8 กระแสงานของบริการสั่งซื้อไวน์บริการที่สาม.....31
รูปที่ 5.9 การลงทะเบียนของบริการสั่งซื้อไวน์บริการแรก 32
รูปที่ 5.10 การลงทะเบียนของบริการสั่งซื้อไวน์บริการที่สอง..... 32
รูปที่ 5.11 การลงทะเบียนของบริการสั่งซื้อไวน์บริการที่สาม 32
รูปที่ 5.12 การกรอกข้อมูลยูอาร์แอลของออนโทโลจีกระบวนการที่ใช้ร่วมกันของบริการสั่งซื้อไวน์..... 33
รูปที่ 5.13 การกรอกข้อมูลค้นหาบริการสั่งซื้อไวน์ของกรณีแรก..... 33

ภาพประกอบ หน้า

รูปที่ 5.14 ผลการค้นหาบริการสั่งซื้อไวน์ของกรณีแรก..... 35
รูปที่ 5.15 การกรอกข้อมูลค้นหาบริการสั่งซื้อไวน์ของกรณีที่สอง..... 36
รูปที่ 5.16 ผลการค้นหาบริการสั่งซื้อไวน์ของกรณีที่สอง..... 37
รูปที่ 5.17 กระแสงานของบริการจองโรงแรมบริการแรก..... 37
รูปที่ 5.18 กระแสงานของบริการจองโรงแรมบริการที่สอง.....38
รูปที่ 5.19 การลงทะเบียนของบริการจองโรงแรมบริการแรก.....38
รูปที่ 5.20 การลงทะเบียนของบริการจองโรงแรมบริการที่สอง.....39
รูปที่ 5.21 การกรอกข้อมูลยูอาร์แอลของออนโทโลจีกระบวนการที่ใช้ร่วมกันของบริการจองโรงแรม.....39
รูปที่ 5.22 การกรอกข้อมูลค้นหาบริการจองโรงแรมของกรณีแรก..... 40
รูปที่ 5.23 ผลการค้นหาบริการจองโรงแรมของกรณีแรก.....41
รูปที่ 5.24 การกรอกข้อมูลค้นหาบริการจองโรงแรมของกรณีที่สอง..... 42
รูปที่ 5.25 ผลการค้นหาบริการจองโรงแรมของกรณีที่สอง..... 43
รูปที่ ก.1 อาวล์-เอสโพรเซสโมเดลของบริการเงินกู้ของธนาคารแรก..... 49
รูปที่ ก.2 อาวล์-เอสโพรเซสโมเดลของบริการสั่งซื้อไวน์บริการแรก..... 52
รูปที่ ก.3 อาวล์-เอสโพรเซสโมเดลของบริการจองโรงแรมบริการแรก..... 53

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

ตาราง หน้า

ตารางที่ 3.1 ตารางเปรียบเทียบพฤติกรรมเชิงหน้าที่ของบริการกู้เงินของธนาคารทั้งสามกับข้อความค้นหา.....20
ตารางที่ 5.1 ตารางเปรียบเทียบพฤติกรรมเชิงหน้าที่ของบริการสั่งซื้อไวน์ทั้งสามกับข้อความค้นหา..... 34

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เว็บเซอร์วิส (Web Services) [1] เป็นมาตรฐานที่ได้รับความนิยมอย่างมากในปัจจุบัน เนื่องจากเป็นมาตรฐานเปิดในการติดต่อสื่อสารแลกเปลี่ยนข้อมูลและบริการ (Services) ผ่านเครือข่ายอินเทอร์เน็ต (Internet) โดยไม่ขึ้นกับระบบปฏิบัติการ หรือภาษาโปรแกรมที่ใช้ในการพัฒนา ซึ่งเป็นประโยชน์ในการเพิ่มความยืดหยุ่นและลดต้นทุนในการพัฒนาแอพพลิเคชัน (Application)

การค้นหาบริการ (Service Discovery) เป็นส่วนสำคัญในการประมวลผลแบบอิงบริการ (Service-Oriented Computing) [2] โดยทั่วไปการค้นหาบริการจะค้นหาผ่านยูดีดีไอ (UDDI) [3] ซึ่งเป็นส่วนรับลงทะเบียนจากผู้ให้บริการ (Service Provider) ในสถาปัตยกรรมเว็บเซอร์วิส (Web Service Architecture) แต่การค้นหาบริการจากยูดีดีไอมีข้อจำกัดเนื่องจากระบบลงทะเบียนยูดีดีไอ (UDDI Registry) นิยามข้อมูลสำหรับอธิบายบริการในรูปแบบของแอททริบิวท์ (Attribute) การค้นหาบริการด้วยยูดีดีไอสามารถทำได้โดยใช้ชื่อ (Name) รหัส (Key) หมวดหมู่ (Category) ของเอนติตีธุรกิจ (Business Entities) และของบริการ หรือ ใช้ทีโมเดล (tModel) ในการค้นหา อย่างไรก็ตามกลุ่มของแอททริบิวท์ที่ไม่ยืดหยุ่นทำให้การสอบถามข้อมูล (Query) มีข้อจำกัด โดยการค้นหาจากแอททริบิวท์นั้น ผู้ใช้บริการจะไม่สามารถทราบได้ว่าแท้ที่จริงแล้วลักษณะของการให้บริการเป็นอย่างไร ผู้ให้บริการอาจใช้ยูดีดีไอเป็นเพียงช่องทางหนึ่งในการโฆษณาโฮมเพจ (Homepage) ของตนเอง แต่ยังต้องการให้ผู้ใช้บริการเข้าไปดูรายละเอียดของบริการภายในโฮมเพจนั้นซึ่งทำให้ไม่สะดวกต่อผู้ใช้บริการ จึงมีความพยายามที่จะทำให้การค้นหาบริการเป็นไปในลักษณะเชิงความหมายมากยิ่งขึ้น

วิธีหนึ่งที่ใช้อธิบายบริการให้มีลักษณะเชิงความหมายคือ การใช้ภาษาออนโทโลจี (Ontology Language) ซึ่งเป็นวิธีที่ใช้กันอย่างแพร่หลายในงานวิจัยเกี่ยวกับเว็บเซอร์วิส เนื่องจากภาษาออนโทโลจีสามารถอธิบายบริการได้หลายแง่มุม และยังสามารถทำการคิดเหตุผล (Reasoning) ได้ [4] คำอธิบายบริการซึ่งใช้ภาษาออนโทโลจีและเป็นที่รู้จักคือ อาวล์-เอส (OWL-S) [5] ซึ่งประกอบด้วย 3 โพรไฟล์ (Profile) คือ เซอร์วิสโพรไฟล์ (Service Profile) โพรเซสโมเดล (Process Model) และเซอร์วิสกราวน์ดิง (Service Grounding) เซอร์วิสโพรไฟล์ทำหน้าที่อธิบายลักษณะและพฤติกรรมเชิงหน้าที่ (Functional Behaviour) เช่น อินพุท (Input) เอาท์พุท (Output) เงื่อนไขก่อนการทำงาน (Precondition) เอฟเฟ็กต์ (Effect) โพรเซสโมเดลอธิบายการดำเนินการ (Operation) ของบริการในรูปแบบของพฤติกรรมเชิงหน้าที่ โครงสร้างควบคุม (Control Structure)

และโครงสร้างการไหลของข้อมูล (Data Flow Structure) เซอร์วิสกราวน์ดิงทำหน้าที่แมป (Map) โพรเซสโมเดลไปยังดับเบิลยูเอสดีแอล (WSDL)

อีกแนวทางหนึ่งสำหรับการอธิบายบริการในลักษณะเชิงความหมายคือ เว็บเซอร์วิสโมเดลลิงออนโทโลจี (Web Services Modeling Ontology, WSMO) [6] ซึ่งจัดเตรียมเฟรมเวิร์ค (Framework) สำหรับการอธิบายเว็บเซอร์วิสด้วยภาษาเว็บเซอร์วิสโมเดลลิง (Web Services Modeling Language, WSML) [7] ภาษาเว็บเซอร์วิสโมเดลลิงอธิบายความหมายด้วย 4 ส่วนย่อย (Element) คือ ออนโทโลจี (Ontology) เป้าหมาย (Goal) คำอธิบายเว็บเซอร์วิส (Web Service Description) และตัวกลาง (Mediator) ออนโทโลจีจัดเตรียมคำศัพท์ (Vocabulary) คอนเซ็ปต์ (Concept) ตัวอย่าง (Instance) และแอ็กซ์เซียม (Axiom) ซึ่งจะถูกส่วนย่อยอื่นเรียกใช้ เป้าหมายจะเปรียบได้กับการค้นหา คำอธิบายเว็บเซอร์วิสจะอธิบายการให้บริการด้วยข้อสมมติเกี่ยวกับเว็บเซอร์วิส (Assumption) เงื่อนไขก่อนการทำงาน เงื่อนไขหลังการทำงาน (Postcondition) เอฟเฟ็กต์ ข้อกำหนดของอินเทอร์เฟส (Interface Specification) และข้อกำหนดของการทำการร่วมกันระหว่างเว็บเซอร์วิส (Orchestration Specification) ส่วนตัวกลางจะเชื่อมโยงแต่ละส่วนย่อยของภาษาเว็บเซอร์วิสโมเดลลิงเข้าด้วยกันและจัดการความแตกต่างระหว่างส่วนย่อยเหล่านั้น เว็บเซอร์วิสโมเดลลิงออนโทโลจีมีความคล้ายคลึงกับอาวล์-เอสในเรื่องการสนับสนุนการค้นหาบริการโดยใช้ออนโทโลจี

งานวิจัยเกี่ยวกับการค้นหาเว็บเซอร์วิสส่วนใหญ่มุ่งเน้นไปยังหน้าที่ของบริการ มีเพียงส่วนน้อยที่สนใจกระบวนการทำงานภายใน เช่น ยูดีดีไอรุ่น (Version) 4 [8] มีการแบ่งหมวดหมู่ของบิสิเนสเอนติตี (Business Entity) และบิสิเนสเซอร์วิส (Business Service) โดยใช้ออนโทโลจีเป็นพื้นฐาน ซึ่งทำให้ยูดีดีไอสามารถค้นหาบริการในหมวดหมู่อย่างกว้างขวางหรือหมวดหมู่ที่จำเพาะเจาะจง งานวิจัย [9] นำเสนอโมเดลการอธิบายเว็บเซอร์วิสที่เรียกว่า เซอร์วิสโพรไฟล์รวม (Integrated Service Profile) ซึ่งอธิบายความสามารถของเว็บเซอร์วิสในหลายแง่มุม และนำไปใช้ในการค้นหาเว็บเซอร์วิส งานวิจัย [10] นำออนโทโลจีกระบวนการ (Process Ontology) มาเป็นพื้นฐานในการค้นหาบริการ แต่อยู่ในรูปแบบของการค้นหาเป้าหมาย-เป้าหมายย่อย (Goal-Subgoal) ของกระบวนการ งานวิจัย [11] นำอาวล์-เอสโพรเซสโมเดลมาอธิบายพฤติกรรมของเว็บเซอร์วิสเพื่อการค้นหาแต่ก็สนใจเฉพาะพฤติกรรมเชิงหน้าที่ แม้ว่างานวิจัย [10], [11] จะนำออนโทโลจีกระบวนการมาใช้ แต่ไม่ได้พิจารณาถึงกระบวนการย่อยภายใน

งานวิจัยนี้นำเสนอแนวทางในการค้นหาบริการด้วยข้อกำหนดกระบวนการ โดยสนใจทั้งพฤติกรรมเชิงหน้าที่และกระแสงาน (Workflow) ภายในกระบวนการ เช่น ต้องการค้นหาบริการขายซอฟต์แวร์ที่เมื่อเสร็จสิ้นกระบวนการซื้อซอฟต์แวร์แล้ว ร้านค้านั้นจะทำการลงทะเบียนลูกค้าในคอร์สสอนการใช้งานให้โดยอัตโนมัติ ดังนั้นจะต้องนำข้อกำหนดกระบวนการของเว็บเซอร์วิส

ขายซอฟต์แวร์แต่ละรายมาพิจารณา การเขียนข้อกำหนดกระบวนการอาจทำได้โดยใช้อาวล์-เอส หรือเว็บเซอร์วิสโมเดลลิงออนโทโลจี แต่อาวล์-เอสนั้นได้มีการพัฒนามาก่อน และยังมีเครื่องมือที่สนับสนุนอีกมากมาย งานวิจัยนี้จึงนำอาวล์-เอสโพรเซสโมเดลมาใช้เป็นข้อกำหนดกระบวนการสำหรับการค้นหาเว็บเซอร์วิส และจะแตกต่างจากงานวิจัยส่วนใหญ่ที่มักใช้อาวล์-เอสโพรเซสโมเดลเพื่อวัตถุประสงค์อื่น ได้แก่ การประกอบบริการ (Composition) การประสานการทำงานของกระแสงาน (Coordination) และการเฝ้าสังเกตกระแสงาน (Monitoring) นอกจากนี้ภายในกระแสงานของบริการอาจมีส่วนเงื่อนไขกระบวนการ (Guard) เช่น การลงทะเบียนเรียนการใช้ซอฟต์แวร์ให้โดยอัตโนมัตินั้น จะเกิดขึ้นก็ต่อเมื่อผู้ใช้บริการมียอดสั่งซื้อเกินกว่าห้าแสนบาท ซึ่งต้องนำเงื่อนไขเหล่านี้มาพิจารณาในการค้นหาด้วย งานวิจัยนี้ได้นำเสนอตัวอย่างบริการ และมีเงื่อนไขกระบวนการซึ่งอธิบายด้วยภาษากฎ (Rule Language) เช่น ภาษาสเวิร์ล (SWRL) [12] โดยผู้ใช้บริการสามารถค้นหาบริการด้วยพฤติกรรมเชิงหน้าที่และกระแสงานของกระบวนการได้

1.2 วัตถุประสงค์ของการวิจัย

เพื่อพัฒนาตัวกลางในการรับลงทะเบียนและค้นหาเว็บเซอร์วิสด้วยข้อกำหนดกระบวนการ โดยตัวกลางนี้จะเพิ่มความสามารถจากยูดีดีไอ

1.3 ขอบเขตของการวิจัย

1.3.1 ตัวกลางจะมีความสามารถในการรับลงทะเบียนข้อกำหนดกระบวนการจากผู้ให้บริการ และช่วยผู้ใช้บริการในการค้นหาเว็บเซอร์วิส โดยพิจารณาข้อกำหนดกระบวนการ

1.3.2 งานวิจัยนี้ใช้ข้อกำหนดกระบวนการตามอาวล์-เอสโพรเซสโมเดล และพิจารณาพฤติกรรมเชิงหน้าที่ร่วมกับกระแสงานที่มีโครงสร้างแบบการตัดสินใจ แบบลำดับ แบบทำงานพร้อมกัน และแบบทำงานวนซ้ำ

1.3.3 เงื่อนไขการเข้าคู่จะพิจารณาจากเงื่อนไขทางรูปแบบกระบวนการ ซึ่งพิจารณาจากเงื่อนไขทางออนโทโลจี เงื่อนไขแบบช่วงตัวเลข และเงื่อนไขทางตรรกะ

1.3.4 งานวิจัยนี้จะไม่พิจารณาการจัดลำดับ (Ranking) การเข้าคู่ของบริการที่ตรงตามความต้องการของผู้ใช้บริการ

1.4 ขั้นตอนการวิจัย

1.4.1 ศึกษาการทำงานของอาวล์-เอสโพรเซสโมเดล

1.4.2 ศึกษางานวิจัยเกี่ยวกับการค้นหาเว็บเซอร์วิส

1.4.3 ศึกษาเครื่องมือเกี่ยวกับการค้นหาเว็บเซอร์วิส

1.4.4 พัฒนาตัวกลางสำหรับการประกาศและการค้นหาเว็บเซอร์วิสด้วยอาวล์-เอสโพรเซสโมเดล

1.4.5 ทดสอบการทำงานของตัวกลางและปรับปรุงตามความเหมาะสม

1.4.6 สรุปผลการวิจัย และจัดทำรายงานวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ได้ตัวกลางที่สามารถใช้ในการประกาศและค้นหาเว็บเซอร์วิสด้วยข้อกำหนดกระบวนการตามอาวล์-เอสโพรเซสโมเดล อันจะเป็นการเพิ่มความสามารถในการค้นหาเว็บเซอร์วิสให้สามารถทำการค้นหาในเชิงกระบวนการซึ่งยังทำไม่ได้ในปัจจุบัน

1.6 ผลงานตีพิมพ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้ตีพิมพ์และนำเสนอในการประชุมวิชาการดังนี้

1. Proceedings of 6th International Conference on Distributed Applications and Interoperable Systems, LNCS 4025, Bologna, Italy (2006): 113-127 ในบทความเรื่อง Discovering Semantic Web Services with Process Specifications โดยผู้แต่งคือ Piya Suwannopas และ Twittie Senivongse

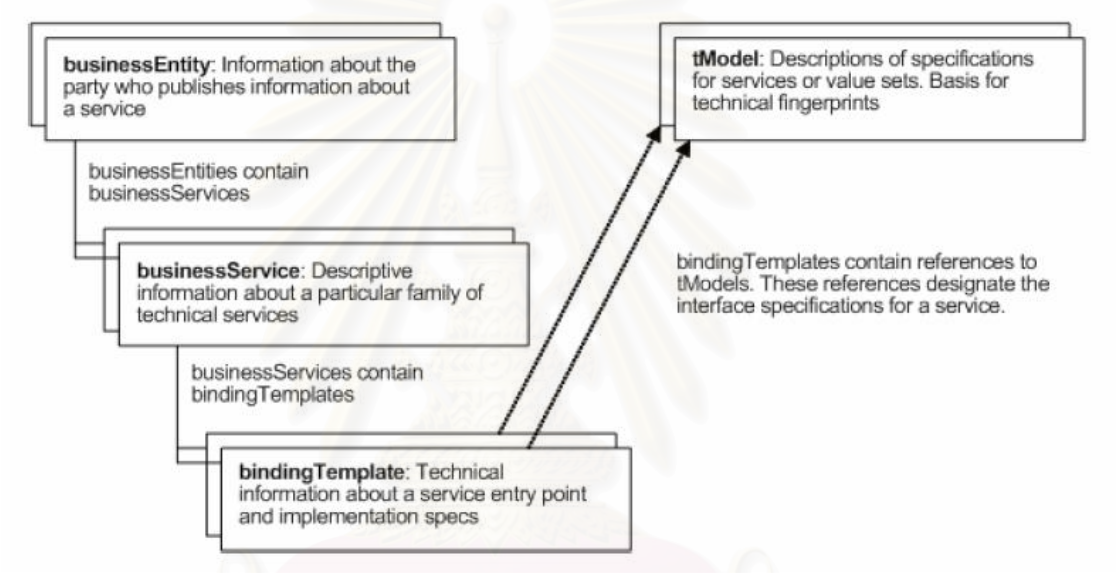
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 แนวคิดและทฤษฎี

2.1.1 ยูดีดีไอ (UDDI: Universal Description Discovery and Integration)

ยูดีดีไอ [3] เป็นมาตรฐานที่ใช้ในการโฆษณาและค้นหาบริการในเว็บเซอร์วิส โดยกำหนดให้การระบุข้อมูลของบริการเป็นไปในรูปแบบเดียวกัน ทำให้การค้นหาบริการทำได้ง่าย โครงสร้างของคำอธิบายบริการในยูดีดีไอแสดงในรูปที่ 2.1



รูปที่ 2.1 โครงสร้างยูดีดีไอ

จากรูป คำอธิบายบริการในยูดีดีไอประกอบด้วย 4 ส่วนหลัก ได้แก่

1. บิสิเนสเอนติตี (Business Entity) อธิบายรายละเอียดเกี่ยวกับองค์กรธุรกิจของผู้ให้บริการ เช่น ชื่อขององค์กรธุรกิจที่ให้บริการ (Name) คำอธิบายเกี่ยวกับองค์กรนั้น (Description) ข้อมูลสำหรับติดต่อองค์กรนั้น (Contacts) รหัสประจำตัวของธุรกิจนั้น (Identifier) บริการที่องค์กรนี้ให้บริการ (Business Services) รหัสที่ใช้ในการจำแนกประเภทธุรกิจ (Category) ซึ่งอาจแบ่งได้ตามอุตสาหกรรม ผลิตภัณฑ์ และรหัสทางภูมิศาสตร์ เช่น NAICS (North American Industry Classification System) UNSPSC 3.1 (United Nation Standard Product and Service Code System) SIC (Standard Industry Classification Code)

2. บิสิเนสเซอร์วิส (Business Service) อธิบายรายละเอียดของการบริการ เช่น ชื่อของบริการ (Name) คำอธิบายของบริการ (Description) รหัสประจำตัวของบริการ (Identifier) และรหัสที่ใช้ในการจำแนกประเภทธุรกิจ

3. ไบน์ดิงเทมเพลต (Binding Template) คือ แบบจำลองสำหรับการผูกมัดบริการ เช่น คำอธิบายสำหรับไบน์ดิงเทมเพลต (Description) ที่อยู่สำหรับอ้างอิงถึงบริการ (Access Point)

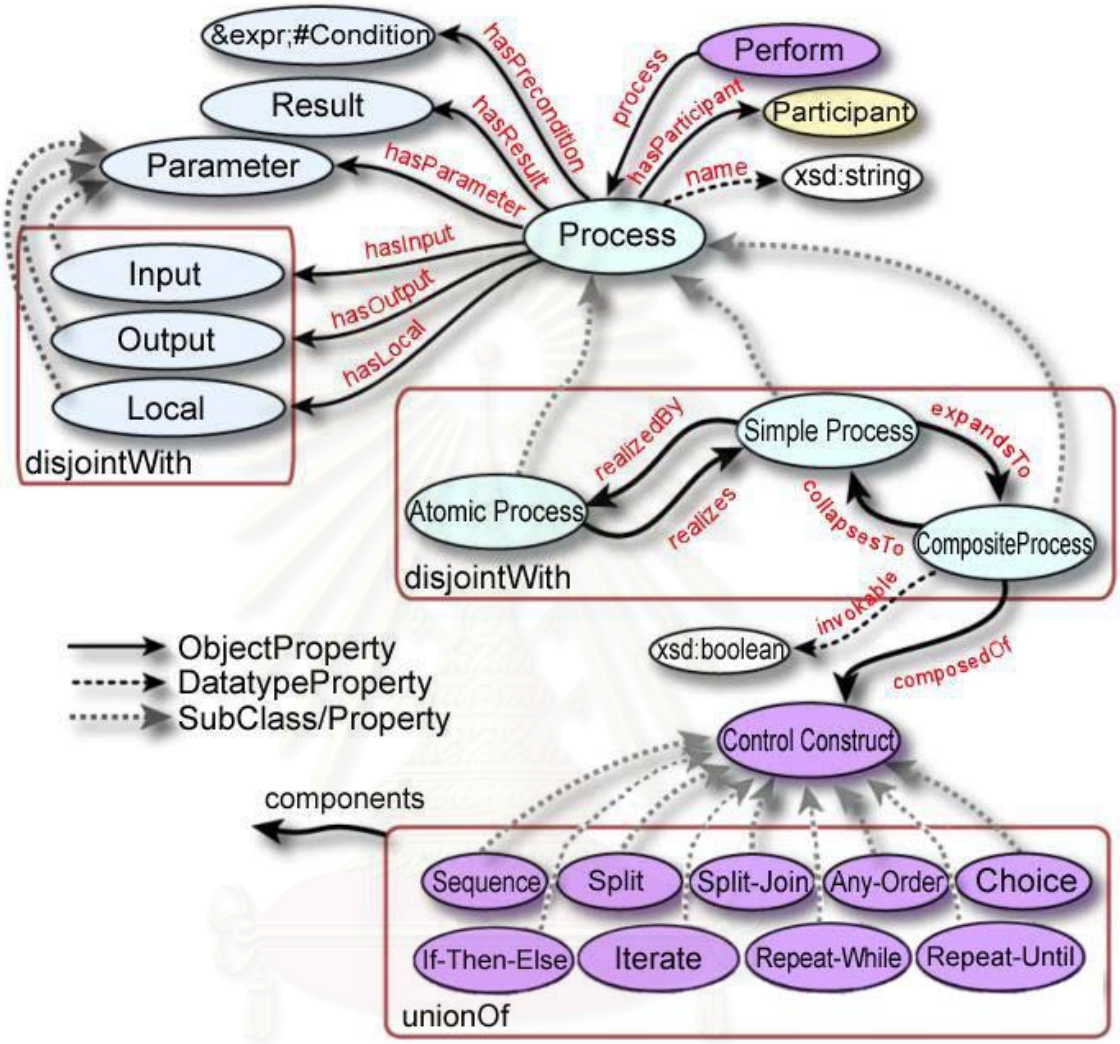
4. ทีโมเดล (tModel) เป็นคำอธิบายที่ใช้ในการกำหนดชนิดของบริการ (Service Type) และใช้อ้างอิงสำหรับการใช้บริการ ประกอบด้วย ชื่อของทีโมเดล (Name) คำอธิบายสำหรับทีโมเดล (Description)

2.1.2 อาวล์-เอสโพรเซสโมเดล (OWL-S Process Model)

อาวล์-เอสโพรเซสโมเดลเป็นโพรไฟล์หนึ่งของอาวล์-เอส [5] ทำหน้าที่อธิบายการดำเนินการของบริการในรูปแบบของพฤติกรรมเชิงหน้าที่ โครงสร้างควบคุม และโครงสร้างการไหลของข้อมูล โดยบริการที่ประกาศไว้จะอธิบายด้วยเซอร์วิสโมเดล (Service Model) ซึ่งมีกระบวนการ (Process) เป็นซับคลาส (Subclass) ของเซอร์วิส รูปที่ 2.2 แสดงอัปเปอร์ออนโทโลจี (Upper Ontology) ของอาวล์-เอสโพรเซสโมเดล ซึ่งประกอบด้วยคลาส (Class) และพร็อพเพอร์ตี (Property) ซึ่งร่วมกันอธิบายการทำงานของบริการ กระบวนการจะอธิบายพฤติกรรมเชิงหน้าที่โดยระบุอินพุท เอาท์พุท เงื่อนไขก่อนการทำงาน และเอฟเฟ็กต์ เงื่อนไขก่อนการทำงานคือ นิพจน์ (Expression) ซึ่งต้องเก็บค่าไว้เพื่อกระบวนการจะได้ถูกเรียกใช้งาน โลคัล (Local) เป็นพารามิเตอร์เสริมซึ่งผูกติดกับเงื่อนไขก่อนการทำงาน และมีประโยชน์ในการพิจารณาค่าเชิงตรรกะของเงื่อนไขก่อนการทำงาน ผลลัพธ์ (Result) เกี่ยวข้องกับเอาท์พุทและเอฟเฟ็กต์ ซึ่งอาจถูกกำหนดเงื่อนไขความสัมพันธ์โดยพร็อพเพอร์ตีของเงื่อนไขในการทำงาน (Incondition) ซึ่งระบุเงื่อนไขทางตรรกะของผลลัพธ์ที่จะเกิด ดังนั้นเอาท์พุทและเอฟเฟ็กต์ จะกลายเป็นเอาท์พุทเชิงเงื่อนไข (Conditional Output) และเอฟเฟ็กต์เชิงเงื่อนไข (Conditional Effect)

กระบวนการยังสามารถอธิบายด้วยการประกอบกันของกระบวนการย่อย กระบวนการย่อยอาจเป็นกระบวนการอะตอมมิก (Atomic Process) กระบวนการประกอบ (Composite Process) หรือกระบวนการอย่างง่าย (Simple Process) กระบวนการอะตอมมิกเป็นกระบวนการที่ไม่สามารถมีกระบวนการย่อยได้อีก สามารถเรียกใช้ได้โดยตรง และกระทำการ (Execute) ภายในขั้นตอนเดียว กระบวนการประกอบสามารถแยกได้เป็นกระบวนการที่ไม่ใช่กระบวนการประกอบหรือเป็นกระบวนการประกอบ การแยกกระบวนการสามารถระบุโดยใช้โครงสร้างควบคุม (Control Structure) เช่น ลำดับ (Sequence) ตัวแบ่ง (Split) ตัวแบ่ง-รวม (Split-Join) ลำดับอิสระ (Any-Order) ตัวเลือก (Choice) ทางเลือก (If-Then-Else) อิเทอเรท (Iterate) รีพีท-ไวล์ (Repeat-While) รีพีท-อันทิล (Repeat-Until) กระบวนการอย่างง่ายไม่สามารถเรียกใช้ได้โดยตรง และไม่ได้เชื่อมโยงกับกราวน์ดิง แต่กระทำการภายในขั้นตอนเดียว เช่นเดียวกับกระบวนการอะตอมมิก กระบวนการอย่างง่ายเป็นเพียงการกำหนดนามธรรม (Abstraction) ของการใช้งานกระบวนการอะตอมมิก หรือ

เตรียมการแสดงตัวอย่างอย่างง่ายของกระบวนการประกอบเพื่อการวางแผนหรือการคิดหาเหตุผล (Reasoning)



รูปที่ 2.2 อัปเปอร์ออนโทโลจีของอาวล์-เอสโพรเซสโมเดล

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.2.1 การเข้าคู่และการจัดลำดับเว็บเซอร์วิสเชิงความหมายโดยใช้เซอร์วิสโพรไฟล์รวม (Matchmaking and Ranking of Semantic Web Services Using Integrated Service Profile) โดย Natenapa Sriharee, Twittie Senivongse

งานวิจัย [9] นำเสนอโมเดลการอธิบายเว็บเซอร์วิสที่เรียกว่า เซอร์วิสโพรไฟล์รวม (Integrated Service Profile) ซึ่งอธิบายความสามารถของเว็บเซอร์วิสในหลายแง่มุม และนำไปใช้ใน

การค้นหาเว็บเซอร์วิสได้ ผลการค้นหาเว็บเซอร์วิสยังได้นำมาเรียงลำดับความถูกต้องของการค้นหาอีกด้วย เซอร์วิสโพรไฟล์รวมประกอบด้วยโพรไฟล์ย่อย 2 โพรไฟล์ คือ

1. โพรไฟล์แอททริบิวท์อย่างง่าย (Simple Attribute Profile) เป็นโพรไฟล์ที่ใช้สำหรับประกาศเว็บเซอร์วิสในยูดีดีไอแบบปกติ

2. โพรไฟล์ความสามารถ (Capability Profile) ถูกจัดการโดยระบบลงทะเบียนเชิงความหมาย (Semantic Registry) ตัวโพรไฟล์ความสามารถเองไม่ได้แสดงถึงความสามารถของเว็บเซอร์วิสแต่อย่างใด แต่จะใช้โพรไฟล์ย่อยในการแสดงความสามารถ ซึ่งประกอบด้วยโพรไฟล์ย่อย 4 โพรไฟล์

2.1 โพรไฟล์แอททริบิวท์เชิงความหมาย (Semantic Attribute Profile) บรรจุแอททริบิวท์เชิงความหมาย ซึ่งมีค่าเชิงออนโทโลจีอยู่ เพื่อประโยชน์ในการคิดหาเหตุผล

2.2 โพรไฟล์โครงสร้าง (Structural Profile) เป็นโครงสร้างของความรู้เชิงความหมาย (Semantic Knowledge) เกี่ยวกับบริการ ซึ่งผู้ใช้บริการคาดหวังที่จะรู้ก่อนการตัดสินใจใช้บริการ เช่น ผลิตภัณฑ์ของบริการ รายละเอียดการขาย การส่งของ

2.3 โพรไฟล์พฤติกรรม (Behavioural Profile) เป็นความสามารถทางหน้าที่ของบริการด้านการปฏิบัติงาน อินพุท เอาท์พุท เอฟเฟ็กต์ ซึ่งบางครั้งจะต้องมีเงื่อนไขเข้ามาเกี่ยวข้อง เงื่อนไขนี้จะเกี่ยวข้องโดยตรงกับโพรไฟล์กฎ

2.4 โพรไฟล์กฎ (Rule Profile) แบ่งเป็น 2 ชนิดคือ เงื่อนไขเชิงพฤติกรรม (Behavioural Constraint) ซึ่งจะเกี่ยวข้องกับโพรไฟล์พฤติกรรม และเงื่อนไขในการปฏิบัติงาน (Operational Constraint) ซึ่งไม่เกี่ยวข้องกับโพรไฟล์พฤติกรรม

ทั้งโพรไฟล์ความสามารถ และโพรไฟล์ย่อยทั้ง 4 โพรไฟล์สืบทอดมาจากอัปเปอร์ออนโทโลจีของแต่ละโพรไฟล์

งานวิจัยนี้ได้นำเสนอหลักการในการค้นหาเว็บเซอร์วิสโดยพิจารณาจากทุกโพรไฟล์ข้างต้น โดยแต่ละโพรไฟล์จะมีเงื่อนไขในการค้นหาที่แตกต่างกัน ซึ่งเงื่อนไขเหล่านี้ได้นำมาใช้ในการจัดลำดับความถูกต้องของการค้นหา งานวิจัยนี้ได้ยกตัวอย่างการค้นหาเว็บเซอร์วิส และยังได้นำเสนอเฟรมเวิร์กสำหรับการประกาศและการค้นหาเว็บเซอร์วิส แต่งานวิจัยนี้ไม่ได้พิจารณาการค้นหาเว็บเซอร์วิสจากกระแสงานของกระบวนการทำงานของเซอร์วิส

2.2.2 การค้นหาบริการเชิงความหมายด้วยออนโทโลจีกระบวนการ (Searching for Services on the Semantic Web Using Process Ontologies) โดย Mark Klien, Abraham Bernstein

งานวิจัย [10] นำออนโทโลจีกระบวนการ (Process Ontology) มาเป็นพื้นฐานในการค้นหาบริการ ออนโทโลจีกระบวนการอธิบายโดยกระบวนการบริการ (Service Process) งานย่อยที่เป็น

ส่วนประกอบ (Constituent Subtask) พอร์ตการเชื่อมต่อ (Connection Port) ระหว่างงานย่อย (Subtask) กลไกการเชื่อมต่อ (Connection Mechanisms) และเอ็กซ์เซ็พชัน (Exception) ภายในกระบวนการ การค้นหาทำได้โดยใช้ภาษาพีคิวแอล (PQL)

ข้อจำกัดของงานวิจัยนี้ คือ

1. ออนโทโลจีกระบวนการจะอยู่ในรูปแบบของเป้าหมาย-เป้าหมายย่อย (Goal-Subgoal) ของการประกอบบริการ แต่ไม่ได้อยู่ในรูปแบบของกระแสงาน
2. ออนโทโลจีของงานวิจัยนี้ไม่สนับสนุนเงื่อนไขทางกระบวนการ

2.2.3 การเข้าคู่เว็บเซอร์วิสเชิงความหมายด้วยข้อกำหนดบนโพรเซสโมเดล (Matchmaking for Semantic Web Services with Constraints on Process Models) โดย Natenapa Sriharee, Twittie Senivongse

งานวิจัย [11] นำอาวล์-เอสโพรเซสโมเดลมาอธิบายพฤติกรรมของเว็บเซอร์วิส ได้แก่ อินพุท เงื่อนไขก่อนการทำงาน เอาท์พุท เอาท์พุทที่มีเงื่อนไข (Conditional Output) เอฟเฟ็กต์ และเอฟเฟ็กต์ที่มีเงื่อนไข (Conditional Effect) และได้นำอาวล์-เอสโพรเซสโมเดลที่ได้อธิบายพฤติกรรมนี้มาใช้ในเฟรมเวิร์กสำหรับการค้นหาเว็บเซอร์วิส โดยมีการยกตัวอย่างประกอบอย่างชัดเจน การค้นหาเว็บเซอร์วิสคำนึงถึง 3 ส่วนคือ การคิดหาเหตุผลทางออนโทโลจี การคำนวณทางคณิตศาสตร์ และการคิดหาเหตุผลทางกฎเกณฑ์

งานวิจัยนี้พิจารณาอาวล์-เอสโพรเซสโมเดลในการค้นหาเว็บเซอร์วิส ซึ่งเป็นการนำเสนอแนวทางใหม่ในการค้นหาเว็บเซอร์วิสจากพฤติกรรม เนื่องจากงานวิจัยส่วนใหญ่จะใช้อาวล์-เอสเซอร์วิสโพรไฟล์ในการค้นหา อย่างไรก็ตามงานวิจัยนี้ไม่ได้ทำการค้นหาเว็บเซอร์วิสโดยพิจารณาถึงกระแสงานของกระบวนการภายในของเว็บเซอร์วิส

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3
การค้นหาเว็บเซอร์วิสด้วยอาวล์-เอสโพรเซสโมเดล

งานวิจัยนี้เสนอแนวทางใหม่ในการค้นหาเว็บเซอร์วิสด้วยอาวล์-เอสโพรเซสโมเดล แนวคิดของงานวิจัยนี้จะใช้ตัวอย่างของกระแสงานที่มีโครงสร้างแบบการตัดสินใจ ซึ่งเป็นบริการด้านการกู้เงินของธนาคารในการอธิบาย โดยในตัวอย่างมีธนาคาร 3 ธนาคารที่ให้บริการเงินกู้

3.1 ข้อกำหนดกระบวนการ

กระบวนการของบริการกู้เงินนั้นกำหนดโดยผู้ให้บริการเว็บเซอร์วิสโดยใช้อาวล์-เอสโพรเซสโมเดล ซึ่งจะมี 2 ส่วน คือ ส่วนที่แสดงพฤติกรรมเชิงหน้าที่ของบริการ และส่วนที่แสดงถึงกระแสงานของกระบวนการภายใน ผู้ให้บริการสามารถสร้างกระบวนการของบริการจากเครื่องมือสร้างออนโทโลจี เช่น โพรทีเจ (Protégé) [13] รูปที่ 3.1 แสดงส่วนพฤติกรรมเชิงหน้าที่ของบริการของธนาคารแรก

```
1.  <?xml version="1.0"?>
2.  <rdf:RDF xmlns:process="http://www.daml.org/services/owl-s/1.1/Process.owl#"
    xmlns:list="http://www.daml.org/services/owl-s/1.1/generic/ObjectList.owl#" xmlns:swrl="http://www.w3.org/2003/11/swrl#"
    xmlns:time="http://www.isi.edu/~pan/damltime/time-entry.owl#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns="http://localhost:/ontology/local/LoanService1.owl#"
    xmlns:expr="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#" xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
    xmlns:service="http://www.daml.org/services/owl-s/1.1/Service.owl#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:grounding="http://www.daml.org/services/owl-s/1.1/Grounding.owl#" xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:profile="http://www.daml.org/services/owl-s/1.1/Profile.owl#"
    xml:base="http://localhost/ontology/local/LoanService1.owl">
3.    <owl:Ontology rdf:about="">
4.      <owl:imports rdf:resource="http://www.daml.org/rules/proposal/swrl.owl"/>
5.      <owl:imports rdf:resource="http://www.daml.org/rules/proposal/swrlb.owl"/>
6.      <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.1/Profile.owl"/>
7.      <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.1/Grounding.owl"/>
8.    </owl:Ontology>
9.  <process:CompositeProcess rdf:ID="LoanService_S1">
10.   <process:hasInput>
11.     <process:Input rdf:ID="CustomerInfo_S1"/>   <-- อินพุท
12.   </process:hasInput>
13.   <process:hasOutput>
14.     <process:Output rdf:ID="LoanInterestRate_S1"/>   <-- เอาท์พุท
15.   </process:hasOutput>
16.   <process:hasLocal>
17.     <process:Local rdf:ID="IncomePerMonth_S1"/>
18.   </process:hasLocal>
19.   <process:hasLocal>
20.     <process:Local rdf:ID="LoanPurpose_S1"/>   <-- เงื่อนไขก่อนการทำงาน
21.   </process:hasLocal>
22.   <process:hasPrecondition>
23.     <expr:Condition rdf:ID="IncomeCondition_S1">
24.       <expr:expressionLanguage rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#SWRL"/>
25.       <expr:expressionBody rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">
26.         <swrl:AtomList>
27.           <rdf:first>
28.             <swrl:BuiltinAtom>
29.               <swrl:builtin rdf:resource="http://www.w3.org/2003/11/swrlb#greaterThanOrEqual"/>
30.               <swrl:arguments>
31.                 <rdf:List>
32.                   <rdf:first rdf:resource="#IncomePerMonth_S1"/>
33.                   <rdf:rest>
34.                     <rdf:List>
```

รูปที่ 3.1 อาวล์-เอสโพรเซสโมเดลของส่วนแสดงพฤติกรรมเชิงหน้าที่ของบริการของธนาคารแรก

```
                    <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#int">10000</rdf:first>
                    <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
                  </rdf:List>
                </rdf:rest>
              </rdf:List>
            </swrl:arguments>
          </swrl:BuiltinAtom>
        </rdf:first>
        <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
      </swrl:AtomList>
    </expr:expressionBody>
  </expr:Condition>
</process:hasPrecondition>
<process:hasResult>
  <process:Result rdf:ID="PremiumLoanResult_S1">
    <process:hasResultVar>
      <process:ResultVar rdf:ID="LoanAmount_S1"/>
    </process:hasResultVar>
    <process:hasResultVar>
      <process:ResultVar rdf:ID="PremiumCreditCardFee_S1"/>
    </process:hasResultVar>
    <process:inCondition>
      <expr:Condition rdf:ID="PremiumLoanCondition_S1">
        <expr:expressionLanguage rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#SWRL"/>
        <expr:expressionBody rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">
          <swrl:AtomList>
            <rdf:first>
              <swrl:BuiltinAtom>
                <swrl:builtin rdf:resource="http://www.w3.org/2003/11/swrlb#greaterThan"/>
                <swrl:arguments>
                  <rdf:List>
                    <rdf:first rdf:resource="#LoanAmount_S1"/>
                    <rdf:rest>
                      <rdf:List>
                        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#int">300000</rdf:first>
                        <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
                      </rdf:List>
                    </rdf:rest>
                  </rdf:List>
                </swrl:arguments>
              </swrl:BuiltinAtom>
            </rdf:first>
            <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
          </swrl:AtomList>
        </expr:expressionBody>
      </expr:Condition>
    </process:inCondition>
    <process:hasEffect>
      <expr: Expression rdf:ID="PremiumCreditCardExpression_S1">
        <expr:expressionLanguage rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#SWRL"/>
        <expr:expressionBody rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">
          <swrl:AtomList>
            <rdf:first>
              <swrl:BuiltinAtom>
                <swrl:builtin rdf:resource="http://www.w3.org/2003/11/swrlb#equal"/>
                <swrl:arguments>
                  <rdf:List>
                    <rdf:first rdf:resource="#PremiumCreditCardFee_S1"/>
                    <rdf:rest>
                      <rdf:List>
                        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#int">0</rdf:first>
                        <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
                      </rdf:List>
                    </rdf:rest>
                  </rdf:List>
                </swrl:arguments>
              </swrl:BuiltinAtom>
            </rdf:first>
            <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
          </swrl:AtomList>
        </expr:expressionBody>
      </expr:Expression>
    </process:hasEffect>
  </process:Result>
</process:hasResult>
```

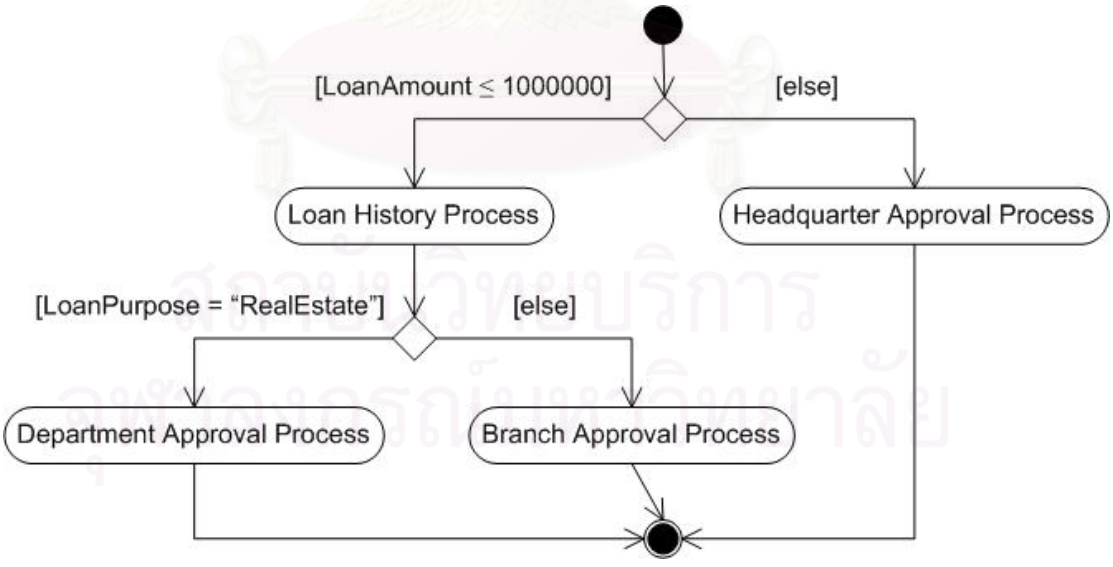
เงื่อนไขก่อนการทำงาน (ต่อ)

เอฟเฟ็กต์

รูปที่ 3.1 อาวล์-เอสโพรเซสโมเดลของส่วนแสดงพฤติกรรมเชิงหน้าที่ของบริการของธนาคารแรก (ต่อ)

จากรูป บริการนี้ต้องการข้อมูลของลูกค้า (Customer Information) เพื่อเป็นอินพุท (บรรทัดที่ 10-12) และให้อัตราดอกเบี้ยเงินกู้เป็นเอาท์พุท (บรรทัดที่ 13-15) บริการมีเงื่อนไขก่อนการทำงาน คือ ลูกค้าต้องมีรายได้อย่างน้อย 10,000 บาทต่อเดือน (บรรทัดที่ 22-47) เอฟเฟ็กต์ของบริการนี้เป็นเอฟเฟ็กต์ที่มีเงื่อนไข ซึ่งขึ้นอยู่กับจำนวนเงินกู้ ถ้าเงินกู้มากกว่า 300,000 บาท ถือเป็นการกู้แบบพิเศษ (Premium Loan) (บรรทัดที่ 56-81) ซึ่งลูกค้าจะได้รับบัตรเครดิตพิเศษ (Premium Credit Card) และเอฟเฟ็กต์มีผลคือไม่ต้องเสียค่าธรรมเนียม (บรรทัดที่ 82-107) โดยเงื่อนไขทั้งหมดอธิบายด้วยภาษากฎสเวิร์ล (SWRL) สำหรับอีกสองธนาคารสมมติให้มีรายละเอียดส่วนแสดงพฤติกรรมเชิงหน้าที่เช่นเดียวกับธนาคารแรก

ส่วนแสดงกระแสงานของกระบวนการภายในของธนาคารแรก แสดงได้ด้วยแผนภาพดังรูปที่ 3.2 โดยการให้บริการกู้เงินประกอบด้วยคลาสของการพิจารณาเงินกู้ (Loan Approval) เช่น การพิจารณาเงินกู้โดยแผนก (Department Approval) จะเกิดขึ้นเมื่อเงินกู้มีจำนวนน้อยหรือการกู้นั้นไม่ได้มีความสำคัญมากนัก โดยการพิจารณาจะทำโดยผู้จัดการแผนก การพิจารณาเงินกู้โดยสาขา (Branch Approval) เกิดขึ้นเมื่อการกู้มีความสำคัญมากขึ้นแต่ยังสามารถตัดสินใจภายในสาขาโดยผู้จัดการสาขาได้ กรณีอื่นซึ่งการกู้มีความสำคัญมากจะต้องผ่านการพิจารณาโดยสำนักงานใหญ่ (Headquarter Approval) โดยระยะเวลาในการพิจารณาจะเพิ่มขึ้นตามลำดับ กระบวนการภายในยังประกอบด้วยการจัดทำประวัติการกู้เงิน (Loan History Process) เพื่อใช้เป็นข้อมูลอ้างอิงของลูกค้าในอนาคต



รูปที่ 3.2 แผนภาพแสดงกระบวนการภายในบริการเงินกู้ของธนาคารแรก

รูปที่ 3.3 แสดงกระแสงานของกระบวนการภายในดังรูปที่ 3.2 ในรูปของอาวล์-เอสโพรเซสโมเดล เงื่อนไขควบคุมกระแสงานของกระบวนการเงื่อนไขแรกทำการตรวจสอบว่า จำนวน

เงินกู้น้อยกว่าหรือเท่ากับหนึ่งล้านหรือไม่ (บรรทัดที่ 111-137) หากมากกว่าหนึ่งล้านจะเข้าสู่กระบวนการพิจารณาโดยสำนักงานใหญ่ (บรรทัดที่ 202-208) แต่หากน้อยกว่าหรือเท่ากับหนึ่งล้านจะทำประวัติการกู้ (บรรทัดที่ 139-147) และเข้าสู่เงื่อนไขในการพิจารณาต่อไป เงื่อนไขควบคุมกระแสงานของกระบวนการเงื่อนไขที่สองพิจารณาว่า หากการกู้นี้กู้ไปเพื่อทำธุรกรรมด้านอสังหาริมทรัพย์ (บรรทัดที่ 152-178) ก็จะเข้าสู่กระบวนการพิจารณาโดยแผนก (บรรทัดที่ 179-185) มิฉะนั้นจะพิจารณาโดยสาขา (บรรทัดที่ 186-192)

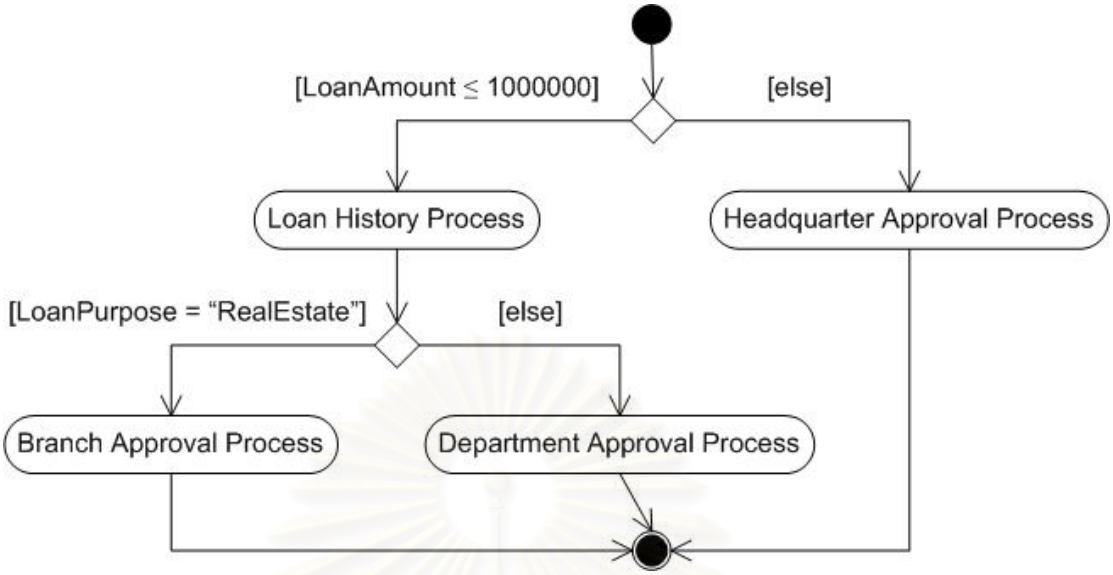
```
110.  <process:composedOf>
111.    <process:If-Then-Else rdf:ID="LoanAmount_If-Then-Else_S1">
112.      <process:ifCondition>
113.        <expr:Condition rdf:ID="LoanAmountCondition_S1">
114.          <expr:expressionLanguage rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#SWRL"/>
115.          <expr:expressionBody rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">
116.            <swrl:AtomList>
117.              <rdf:first>
118.                <swrl:BuiltinAtom>
119.                  <swrl:builtin rdf:resource="http://www.w3.org/2003/11/swrlb#lessThanOrEqual"/>
120.                  <swrl:arguments>
121.                    <rdf:List>
122.                      <rdf:first rdf:resource="#LoanAmount_S1"/>
123.                      <rdf:rest>
124.                        <rdf:List>
125.                          <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1000000</rdf:first>
126.                          <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
127.                        </rdf:List>
128.                      </rdf:rest>
129.                    </rdf:List>
130.                  </swrl:arguments>
131.                </swrl:BuiltinAtom>
132.              </rdf:first>
133.              <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
134.            </swrl:AtomList>
135.          </expr:expressionBody>
136.        </expr:Condition>
137.      </process:ifCondition>
138.      <process:then>
139.        <process:Sequence rdf:ID="Bank_Sequence_S1">
140.          <process:components>
141.            <process:ControlConstructList rdf:ID="LoanHistory_ControlConstructList_S1">
142.              <list:first>
143.                <process:Perform rdf:ID="LoanHistoryPerform_S1">
144.                  <process:process>
145.                    <process:AtomicProcess rdf:ID="LoanHistoryProcess_S1"/>
146.                  </process:process>
147.                </process:Perform>
148.              </list:first>
149.              <list:rest>
150.                <process:ControlConstructList rdf:ID="Bank_ControlConstructList_S1">
151.                  <list:first>
152.                    <process:If-Then-Else rdf:ID="Purpose_If-Then-Else_S1">
153.                      <process:ifCondition>
154.                        <expr:Condition rdf:ID="PurposeCondition_S1">
155.                          <expr:expressionLanguage
                                rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#SWRL"/>
156.                          <expr:expressionBody rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">
157.                            <swrl:AtomList>
158.                              <rdf:first>
159.                                <swrl:BuiltinAtom>
160.                                  <swrl:builtin rdf:resource="http://www.w3.org/2003/11/swrlb#equal"/>
```

รูปที่ 3.3 อาวล์-เอสโพรเซสโมเดลของส่วนแสดงกระบวนการภายในของบริการของธนาคารแรก

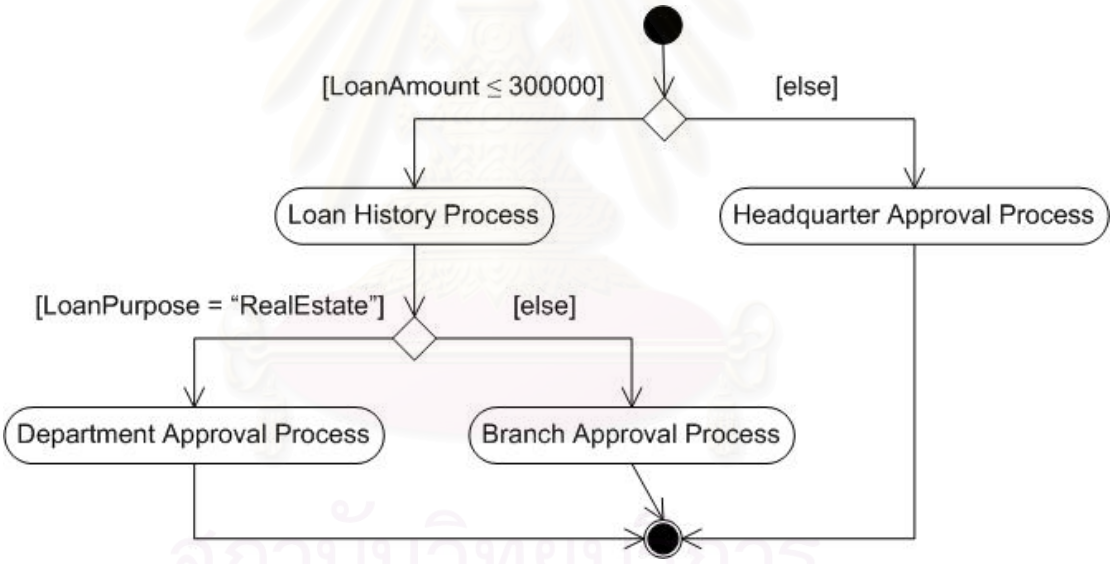
```
161.                        <swrl:arguments>
162.                          <rdf:List>
163.                            <rdf:first rdf:resource="#LoanPurpose_S1"/>
164.                            <rdf:rest>
165.                              <rdf:List>
166.                                <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
                                    http://localhost/ontology/domain/RealEstate.owl#RealEstate</rdf:first>
167.                                <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
168.                              </rdf:List>
169.                            </rdf:rest>
170.                          </rdf:List>
171.                        </swrl:arguments>
172.                      </swrl:BuiltinAtom>
173.                    </rdf:first>
174.                    <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
175.                  </swrl:AtomList>
176.                </expr:expressionBody>
177.              </expr:Condition>
178.            </process:ifCondition>
179.            <process:then>
180.              <process:Perform rdf:ID="DepartmentApprovalPerform_S1">
181.                <process:process>
182.                  <process:AtomicProcess rdf:ID="DepartmentApprovalProcess_S1"/>
183.                </process:process>
184.              </process:Perform>
185.            </process:then>
186.            <process:else>
187.              <process:Perform rdf:ID="BranchApprovalPerform_S1">
188.                <process:process>
189.                  <process:AtomicProcess rdf:ID="BranchApprovalProcess_S1"/>
190.                </process:process>
191.              </process:Perform>
192.            </process:else>
193.          </process:If-Then-Else>
194.        </list:first>
195.        <list:rest rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/ObjectList.owl#nil"/>
196.      </process:ControlConstructList>
197.    </list:rest>
198.  </process:ControlConstructList>
199.  </process:components>
200.  </process:Sequence>
201.  </process:then>
202.  <process:else>
203.    <process:Perform rdf:ID="HeadquarterApprovalPerform_S1">
204.      <process:process>
205.        <process:AtomicProcess rdf:ID="HeadquarterApprovalProcess_S1"/>
206.      </process:process>
207.    </process:Perform>
208.  </process:else>
209.  </process:If-Then-Else>
210.  </process:composedOf>
211.  </process:CompositeProcess>
212.</rdf:RDF>
```

รูปที่ 3.3 อาวล์-เอสโพรเซสโมเดลของส่วนแสดงกระบวนการภายในของบริการของธนาคารแรก (ต่อ)

สำหรับธนาคารที่สองและสาม สมมติให้มีส่วนแสดงกระบวนการภายในต่างจากบริการของธนาคารแรกดังแสดงในรูปที่ 3.4 และ 3.5 ตามลำดับ



รูปที่ 3.4 แผนภาพแสดงกระบวนการภายในบริการเงินกู้ของธนาคารที่สอง



รูปที่ 3.5 แผนภาพแสดงกระบวนการภายในบริการเงินกู้ของธนาคารที่สาม

ออนโทโลจีกระบวนการที่ใช้ร่วมกันภายในโดเมนการกู้เงินซึ่งสร้างจากผู้เชี่ยวชาญของโดเมนนั้น (รูปที่ 3.6) จะถูกอ้างอิงโดยบริการทั้งสาม และเป็นส่วนที่ลูกค้าจะทราบรายละเอียดของบริการและนำไปใช้ในการค้นหาบริการต่อไป แต่บริการทั้งสามจะไม่สามารถทำการอ้างอิงได้โดยตรงเนื่องจากในการประมวลผลข้อกำหนดกระบวนการอาวล์-เอสโพรเซสโมเดลโดยใช้ตัวแจงอาวล์-เอส (OWL-S Parser API) ซึ่งเป็นตัวแจงมาตรฐาน มีข้อจำกัดคือตัวแจงอาวล์-เอสสามารถอ่านได้เฉพาะคลาสที่กำหนดตามมาตรฐานอาวล์-เอสเท่านั้น

```
1.  <?xml version="1.0"?>
2.  <rdf:RDF xmlns:process="http://www.daml.org/services/owl-s/1.1/Process.owl#"
    xmlns:list="http://www.daml.org/services/owl-s/1.1/generic/ObjectList.owl#" xmlns:swrl="http://www.w3.org/2003/11/swrl#"
    xmlns:time="http://www.isi.edu/~pan/damltime/time-entry.owl#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:expr="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#"
    xmlns:swrlb="http://www.w3.org/2003/11/swrlb#" xmlns:service="http://www.daml.org/services/owl-s/1.1/Service.owl#"
    xmlns="http://localhost:8080/ontology/loanservice/loanservicedomain/loanservicedomain.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:grounding="http://www.daml.org/services/owl-s/1.1/Grounding.owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
    xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:profile="http://www.daml.org/services/owl-s/1.1/Profile.owl#"
    xml:base="http://localhost:8080/ontology/loanservice/loanservicedomain/loanservicedomain.owl">
3.    <owl:Ontology rdf:about="">
4.      <owl:imports rdf:resource="http://www.daml.org/rules/proposal/swrlb.owl"/>
5.      <owl:imports rdf:resource="http://www.daml.org/rules/proposal/swrl.owl"/>
6.      <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.1/Profile.owl"/>
7.      <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.1/Grounding.owl"/>
8.    </owl:Ontology>
9.    <owl:Class rdf:ID="CustomerInfo">
10.     <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/Process.owl#Input"/>
11.   </owl:Class>
12.   <owl:Class rdf:ID="LoanInterestRate">
13.     <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/Process.owl#Output"/>
14.   </owl:Class>
15.   <owl:Class rdf:ID="IncomeCondition">
16.     <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#Condition"/>
17.   </owl:Class>
18.   <owl:Class rdf:ID="PremiumLoanCondition">
19.     <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#Condition"/>
20.   </owl:Class>
21.   <owl:Class rdf:ID="LoanAmountCondition">
22.     <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#Condition"/>
23.   </owl:Class>
24.   <owl:Class rdf:ID="PurposeCondition">
25.     <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#Condition"/>
26.   </owl:Class>
27.   <owl:Class rdf:ID="PremiumCreditCardExpression">
28.     <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#Expression"/>
29.   </owl:Class>
30.   <owl:Class rdf:ID="PremiumLoanResult">
31.     <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/Process.owl#Result"/>
32.   </owl:Class>
33.   <owl:Class rdf:ID="IncomePerMonth">
34.     <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/Process.owl#Local"/>
35.   </owl:Class>
36.   <owl:Class rdf:ID="LoanPurpose">
37.     <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/Process.owl#Local"/>
38.   </owl:Class>
39.   <owl:Class rdf:ID="LoanAmount">
40.     <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/Process.owl#ResultVar"/>
41.   </owl:Class>
42.   <owl:Class rdf:ID="PremiumCreditCardFee">
43.     <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/Process.owl#ResultVar"/>
44.   </owl:Class>
45.   <owl:Class rdf:ID="LoanService">
46.     <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/Process.owl#CompositeProcess"/>
47.   </owl:Class>
48.   <owl:Class rdf:ID="HeadquarterApprovalProcess">
49.     <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/Process.owl#AtomicProcess"/>
50.   </owl:Class>
51.   <owl:Class rdf:ID="LoanHistoryProcess">
52.     <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/Process.owl#AtomicProcess"/>
53.   </owl:Class>
54.   <owl:Class rdf:ID="BranchApprovalProcess">
55.     <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/Process.owl#AtomicProcess"/>
56.   </owl:Class>
57.   <owl:Class rdf:ID="DepartmentApprovalProcess">
58.     <rdfs:subClassOf rdf:resource="http://www.daml.org/services/owl-s/1.1/Process.owl#AtomicProcess"/>
59.   </owl:Class>
60. </rdf:RDF>
```

รูปที่ 3.6 ออนโทโลจีกระบวนการที่ใช้ร่วมกันภายในโดเมนการกู้เงิน

ตัวอย่างเช่น <process:AtomicProcess rdf:ID="DepartmentApprovalProcess_S1"/> ในรูปที่ 3.3 บรรทัดที่ 182 ในความเป็นจริงควรจะมีการอ้างอิงออนโทโลจีกระบวนการที่ใช้ร่วมกันภายในโดเมนการกู้เงินโดย DepartmentApprovalProcess_S1 ควรจะเป็นอินสแตนซ์ของคลาส

DepartmentApprovalProcess ในรูปที่ 3.6 บรรทัดที่ 57 แต่จากข้อจำกัดของตัวแจงอาวล์-เอส จึงต้องสร้างแฟ้มข้อมูลอีกแฟ้มหนึ่งเพื่อเชื่อมโยงความสัมพันธ์ระหว่างคลาส DepartmentApprovalProcess ที่กำหนดขึ้นสำหรับโดเมนการกู้เงินกับอินสแตนซ์ DepartmentApprovalProcess_S1 ซึ่งจำเป็นต้องถูกกำหนดเป็นอินสแตนซ์ของคลาสมาตรฐาน <process:AtomicProcess> โดยในแฟ้มข้อมูลนี้จะใช้จีนา (Jena) [14] เพื่อประมวลผลความสัมพันธ์แทน แล้วนำข้อมูลที่ได้ไปรวมเข้ากับข้อมูลที่ประมวลผลได้ในแฟ้มข้อมูลอาวล์-เอสแฟ้มหลัก โดยแฟ้มข้อมูลอาวล์-เอสที่แสดงความสัมพันธ์ระหว่างคลาสในออนโทโลจีกระบวนการของโดเมนการกู้เงินกับกระบวนการภายในบริการเงินกู้ของธนาคารแรกแสดงได้ดังรูปที่ 3.7

```
1.  <?xml version="1.0"?>
2.  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:sls="http://localhost/ontology/shared/SharedLoanService.owl#"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
        xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns="http://localhost/ontology/local/LoanService1Relation.owl#"
        xml:base="http://localhost/ontology/local/LoanService1Relation.owl">
3.    <owl:Ontology rdf:about="">
4.      <owl:imports rdf:resource="http://localhost/ontology/shared/SharedLoanService.owl"/>
5.    </owl:Ontology>
6.    <sls:HeadquarterApprovalProcess rdf:ID="HeadquarterApprovalProcess_S1"/>
7.    <sls:DepartmentApprovalProcess rdf:ID="DepartmentApprovalProcess_S1"/>
8.    <sls:PremiumCreditCardFee rdf:ID="PremiumCreditCardFee_S1"/>
9.    <sls:LoanPurpose rdf:ID="LoanPurpose_S1"/>
10.   <sls:LoanHistoryProcess rdf:ID="LoanHistoryProcess_S1"/>
11.   <sls:BranchApprovalProcess rdf:ID="BranchApprovalProcess_S1"/>
12.   <sls:LoanService rdf:ID="LoanService_S1"/>
13.   <sls:IncomePerMonth rdf:ID="IncomePerMonth_S1"/>
14.   <sls:CustomerInfo rdf:ID="CustomerInfo_S1"/>
15.   <sls:LoanInterestRate rdf:ID="LoanInterestRate_S1"/>
16.   <sls:LoanAmount rdf:ID="LoanAmount_S1"/>
17.  </rdf:RDF>
```

รูปที่ 3.7 แฟ้มข้อมูลอาวล์-เอสที่แสดงความสัมพันธ์ระหว่างคลาสในออนโทโลจีกระบวนการของโดเมนการกู้เงินกับกระบวนการภายในบริการเงินกู้ของธนาคารแรก

นอกจากนี้บริการยังอาจเกี่ยวข้องกับโดเมนออนโทโลจีอื่นๆ เช่น จากบรรทัดที่ 166 ในรูปที่ 3.1 มีการอ้างอิงโดเมนออนโทโลจีเกี่ยวกับอสังหาริมทรัพย์ โดเมนออนโทโลจีนี้แสดงในรูปที่ 3.8

```
1.  <?xml version="1.0"?>
2.  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:owl="http://www.w3.org/2002/07/owl#"
        xmlns="http://localhost/ontology/domain/RealEstate.owl#"
        xml:base="http://localhost/ontology/domain/RealEstate.owl">
3.    <owl:Ontology rdf:about=""/>
4.    <owl:Class rdf:ID="RealEstate"/>
5.    <owl:Class rdf:ID="Land">
6.    <rdfs:subClassOf rdf:resource="#RealEstate"/>
7.    </owl:Class>
8.    <owl:Class rdf:ID="Housing">
9.      <rdfs:subClassOf rdf:resource="#RealEstate"/>
10.   </owl:Class>
11.   …
12. </rdf:RDF>
```

รูปที่ 3.8 บางส่วนของโดเมนออนโทโลจีของอสังหาริมทรัพย์

3.2 เงื่อนไขในการเข้าคู่ (Matching Criteria)

การค้นหาเว็บเซอร์วิสที่มีกระบวนการเป็นไปตามที่ต้องการ จะพิจารณาองค์ประกอบต่างๆ ภายในข้อกำหนดอาวล์-เอสโพรเซสโมเดลว่าเข้าคู่กับข้อความค้นหาหรือไม่ (หัวข้อที่ 3.2.4) โดยการพิจารณาจะอาศัยเงื่อนไขการเข้าคู่ต่างๆ ที่ประยุกต์ใช้จากงานวิจัย [9] ได้แก่ เงื่อนไขทางออนโทโลจี (หัวข้อที่ 3.2.1) เงื่อนไขแบบช่วงตัวเลข (หัวข้อที่ 3.2.2) และเงื่อนไขทางตรรกะ (หัวข้อที่ 3.2.3)

3.2.1 เงื่อนไขทางออนโทโลจี

เงื่อนไขนี้ใช้กับออนโทโลจีกระบวนการที่ใช้ร่วมกันภายในโดเมน รวมทั้งโดเมนออนโทโลจีอื่นๆ ที่เกี่ยวข้อง โดยมีพื้นฐานจากความสัมพันธ์แบบคลาส-ซับคลาสของออนโทโลจี ซึ่งมีใช้ในงานวิจัย [9], [15] กำหนดให้ C_Q เป็นคอนเซ็ปต์ (Concept) ซึ่งระบุในการค้นหา และ C_P เป็นคอนเซ็ปต์ในข้อกำหนดกระบวนการ

1) ถ้า $C_Q \equiv C_P$ แล้ว C_P จะเข้าคู่อย่างถูกต้อง (Exact Match) สำหรับ C_Q เมื่อ $\equiv$ หมายถึง เท่าเทียมกับ
2) ถ้า $C_P \sqsubseteq C_Q$ แล้ว C_P จะเข้าคู่อย่างเจาะจง (Specialised Match) สำหรับ C_Q เมื่อ $\sqsubseteq$ หมายถึง ถูกครอบคลุมโดย (Is Subsumed By)
3) ถ้า $C_Q \sqsubseteq C_P$ แล้ว C_P จะเข้าคู่อย่างกว้าง (Generalised Match) สำหรับ C_Q ซึ่งหมายความว่าคอนเซ็ปต์ของการค้นหาเจาะจงมากกว่าคอนเซ็ปต์ของข้อกำหนดกระบวนการ
4) ถ้า $(C_Q \not\sqsubseteq C_P) \wedge (C_P \not\sqsubseteq C_Q) \wedge (C_Q \sqsubseteq C_C) \wedge (C_P \sqsubseteq C_C)$ แล้ว C_P จะเข้าคู่บางส่วน (Partial Match) สำหรับ C_Q เมื่อ $\not\sqsubseteq$ หมายถึง ไม่ถูกครอบคลุมโดย และ C_C เป็นโหนด (Node) ในออนโทโลจีเดียวกัน
5) กรณีอื่นนอกเหนือจากสี่กรณีข้างต้น C_P จะไม่เข้าคู่ (Fail Match) สำหรับ C_Q

3.2.2 เงื่อนไขแบบช่วงตัวเลข (Numerical Range)

ให้ N_Q เป็นเซตไม่ว่าง (Nonempty Set) ของค่าช่วงตัวเลขของนิพจน์ในการค้นหา (E_Q) และ N_P เป็นเซตไม่ว่างของค่าช่วงตัวเลขของนิพจน์ในข้อกำหนดกระบวนการ (E_P)

1) ถ้า $N_P \subseteq N_Q$ แล้ว E_P จะเข้าคู่อย่างถูกต้องสำหรับ E_Q
2) ถ้า $N_Q \subseteq N_P$ แล้ว E_P จะเข้าคู่แบบเสียบเข้า (Plug-In Match) สำหรับ E_Q
3) ถ้า $(N_P \cap N_Q \neq \phi) \wedge (N_P \not\subseteq N_Q) \wedge (N_Q \not\subseteq N_P)$ แล้ว E_P จะเข้าคู่อย่างอ่อน (Weak Match) สำหรับ E_Q
4) ถ้า $N_P \cap N_Q = \phi$ แล้ว E_P จะไม่เข้าคู่สำหรับ E_Q

3.2.3 เงื่อนไขทางตรรกะ

บริการจะเข้าคู่กับการค้นหาเมื่อเซตของค่าการค้นหาเป็นค่าที่เหมาะสมถูกต้องตามนิพจน์ในกฎ โดยนิพจน์ที่แสดงอาจเป็นเงื่อนไขแบบช่วงตัวเลข หรือเป็นเงื่อนไขทางออนโทโลจี ซึ่งต้องพิจารณาเงื่อนไขตามหัวข้อที่ 3.2.1 และหัวข้อที่ 3.2.2

3.2.4 เงื่อนไขทางรูปแบบกระบวนการ

เงื่อนไขนี้พิจารณาทุกแง่มุมของพฤติกรรมเชิงหน้าที่และกระบวนการภายในกระแสงาน โดยในแต่ละแง่มุมอาจเกี่ยวข้องกับเงื่อนไขทั้งสามข้างต้น ข้อกำหนดกระบวนการจะเข้าคู่กับการค้นหาเมื่อ

1) อินพุท เอาท์พุท เอฟเฟ็กต์ และกระบวนการซึ่งไม่มีเงื่อนไขควบคุมเป็นไปตามการเข้าคู่โดยเงื่อนไขทางออนโทโลจีในหัวข้อที่ 3.2.1 และ
2) เงื่อนไขก่อนการทำงาน เอาท์พุทที่มีเงื่อนไข เอฟเฟ็กต์ที่มีเงื่อนไข และกระบวนการซึ่งมีเงื่อนไขควบคุมเป็นไปตามการเข้าคู่โดยหัวข้อที่ 3.2.2 และหัวข้อที่ 3.2.3

ให้ $\mathbb{R}_Q$ และ $\mathbb{R}_P$ เป็นเซตของพฤติกรรมเชิงหน้าที่และกระบวนการภายในกระแสงาน (ทั้งมีเงื่อนไขและไม่มีเงื่อนไข) ของการค้นหา และของข้อกำหนดกระบวนการตามลำดับ

$$\text{ProcessModelMatch}(\mathbb{R}_Q, \mathbb{R}_P) = \text{true} \Leftrightarrow$$

$$(\mathbb{R}_Q \subseteq \mathbb{R}_P) \wedge (\forall i, \exists j : (i \in \mathbb{R}_Q) \wedge (j \in \mathbb{R}_P) \wedge (i \; \Theta \; j))$$

เมื่อ Θ หมายถึงการเข้าคู่กันในหัวข้อที่ 3.2.1-3.2.3

3.3 ตัวอย่างการค้นหาบริการจากข้อกำหนดกระบวนการ

สมมติมีลูกค้าต้องการกู้เงินจากธนาคารเป็นจำนวน 400,000 บาท เพื่อไปซื้อบ้าน โดยต้องการให้การพิจารณาอนุมัติเงินกู้ทำโดยแผนกเงินกู้ ซึ่งจะทำให้ใช้เวลาน้อยที่สุดในการทราบผล และลูกค้าต้องการให้ธนาคารอนุมัติการเป็นสมาชิกบัตรเครดิตโดยไม่เสียค่าธรรมเนียมด้วย โดยลูกค้ามีรายได้ 20,000 บาทต่อเดือน

ให้การค้นหา ($\mathbb{Q}$) เป็นเซตของนิพจน์โดยอยู่ในรูปแบบ property(subject, object) ซึ่งสอดคล้องกับรูปแบบของอาร์ดีเอฟ (RDF) <subject, property, object> สำหรับเงื่อนไขแบบช่วงตัวเลขจะอยู่ในรูปแบบ property(argument, relationaloperator, literalvalue1 [, literalvalue2]) แต่ละนิพจน์จะมีสัญลักษณ์กำกับว่าเป็นเงื่อนไขเกี่ยวกับอะไร เช่น C, O, G และ P หมายถึงเงื่อนไขก่อนการทำงาน เอาท์พุท เงื่อนไขกระบวนการ และกระบวนการตามลำดับ ดังนั้นจากตัวอย่างจะเขียนเซตของการค้นหาได้เป็น

$$\mathbb{Q} = \{\text{hasIncomePerMonth(IncomePerMonth, Equal, 20000)}^{C},$$
$$\text{hasPremiumCreditCardFee(PremiumCreditCardFee, Equal, 0)}^{O},$$
$$\text{hasLoanAmount(LoanAmount, Equal, 400000)}^{G},$$
$$\text{hasLoanPurpose(LoanPurpose, Housing)}^{G},$$
$$\text{hasProcess(Process, DepartmentApprovalProcess)}^{P}\}$$

ในการพิจารณาว่าบริการหนึ่งๆ ตรงกับความต้องการหรือไม่ ข้อกำหนดกระบวนการของบริการจะถูกแปลงไปเป็นนิพจน์ในทำนองเดียวกับการค้นหา การแปลงข้อกำหนดกระบวนการไปเป็นนิพจน์ทำได้โดยการใช้ตัวแจง (Parser) ซึ่งแบ่งได้เป็นสามส่วนที่แตกต่างกัน ส่วนแรกคือส่วนกระบวนการภายในได้รับการแปลงโดยใช้เอพีไอสำหรับตัวแจงของอาวล์-เอส (OWL-S Parser API) ส่วนถัดมาคือส่วนที่เกี่ยวข้องกับออนโทโลจีของโดเมนต่างๆ ส่วนนี้ได้รับการแปลงโดยตัวแจงสำหรับออนโทโลจีของจีนา และส่วนสุดท้ายคือส่วนที่เป็นภาษากฎสเวิร์ล จะได้รับการแปลงโดยตัวแจงภาษาสเวิร์ล จากนั้นนำนิพจน์เหล่านี้มาคิดหาเหตุผลการเข้าคู่กันโดยใช้เครื่องมือคิดหาเหตุผล บอสซาม (Bossam) [16]

จากตัวอย่างข้างต้น เมื่อพิจารณาจากเงื่อนไขก่อนการทำงานของบริการของธนาคารแรกเกี่ยวกับรายได้ของลูกค้า ซึ่งต้องพิจารณาเงื่อนไขแบบช่วงตัวเลขตามหัวข้อที่ 3.2.2 จะเห็นว่าเงื่อนไขนี้มีการเข้าคู่กันอย่างถูกต้อง ถัดมาจึงพิจารณาเอฟเฟ็กต์ แต่เนื่องจากเป็นเอฟเฟ็กต์แบบมีเงื่อนไข จึงต้องพิจารณาเงื่อนไขในการทำงานเกี่ยวกับจำนวนเงินกู้ก่อน และพิจารณานิพจน์ในเอฟเฟ็กต์เกี่ยวกับค่าธรรมเนียมโดยพิจารณาเงื่อนไขแบบช่วงตัวเลข ซึ่งเงื่อนไขนี้มีการเข้าคู่กันอย่างถูกต้อง เมื่อแง่มุมทั้งหมดของส่วนพฤติกรรมเชิงหน้าที่ผ่านการพิจารณาเงื่อนไข บริการของธนาคารแรกจะถูกนำไปพิจารณาต่อในส่วนกระบวนการภายใน (ในที่นี้เนื่องจากบริการในส่วนพฤติกรรมเชิงหน้าที่ของธนาคารที่สองและสามมีบริการเช่นเดียวกับธนาคารแรก ธนาคารทั้งสามจึงผ่านการพิจารณาในส่วนนี้ทั้งหมด) การพิจารณาพฤติกรรมเชิงหน้าที่สามารถแสดงได้ดังตารางที่ 3.1

ตารางที่ 3.1 ตารางเปรียบเทียบพฤติกรรมเชิงหน้าที่ของบริการกู้เงินของธนาคารทั้งสามกับข้อความค้นหา

ส่วนของข้อมูล	ข้อมูลการค้นหา	ข้อมูลบริการทั้งสาม	ลักษณะการเข้าคู่
เงื่อนไขก่อนการทำงาน	IncomePerMonth = 20000	IncomePerMonth $\geq$ 10000	เข้าคู่อย่างถูกต้อง
เงื่อนไขเอฟเฟ็กต์	PremiumCreditCardFee = 0	PremiumCreditCardFee = 0	เข้าคู่อย่างถูกต้อง

การพิจารณาส่วนของกระบวนการภายใน จะมีการเชื่อมโยงกระบวนการย่อยเข้ากับเงื่อนไขทางกระบวนการ จากบริการเงินกู้ของธนาคารแรก จะได้กฎสำหรับทุกกระบวนการพิจารณาเงินกู้ดังนี้

!hasLoanAmount(LoanAmount, LessThanOrEqual, 1000000) →
hasProcess(Process, HeadquarterApprovalProcess);

hasLoanAmount(LoanAmount, LessThanOrEqual, 1000000) →
hasProcess(Process, LoanHistoryProcess);

hasLoanAmount(LoanAmount, LessThanOrEqual, 1000000),
hasLoanPurpose(LoanPurpose, RealEstate) →
hasProcess(Process, DepartmentApprovalProcess);

hasLoanAmount(LoanAmount, LessThanOrEqual, 1000000),
!hasLoanPurpose(LoanPurpose, RealEstate) →
hasProcess(Process, BranchApprovalProcess);

การพิจารณาในส่วนกระบวนการภายใน เริ่มจากการพิจารณาบริการของธนาคารแรก เงื่อนไขกระบวนการแรกเป็นการพิจารณาเงื่อนไขแบบช่วงตัวเลขเกี่ยวกับจำนวนเงินกู้ ซึ่งผลที่ได้คือมีการเข้าคู่กันอย่างถูกต้อง เนื่องจากการค้นหาต้องการกู้เงิน 400,000 บาทและเงื่อนไขกระบวนการแรกประกาศไว้ว่าหากเงินกู้น้อยกว่า 1,000,000 บาท จะเข้าสู่กระบวนการจัดทำประวัติการกู้เงินและเข้าสู่เงื่อนไขกระบวนการที่สอง เงื่อนไขกระบวนการที่สองจะเกี่ยวกับวัตถุประสงค์ของการกู้ โดยขั้นแรกต้องพิจารณาเงื่อนไขทางออนโทโลจี (หัวข้อที่ 3.2.1) เกี่ยวกับโดเมนออนโทโลจีของอสังหาริมทรัพย์ (รูปที่ 3.8) ก่อน ซึ่ง RealEstate มีซับคลาสเป็น Housing จึงส่งผลให้เงื่อนไขนี้มีการเข้าคู่กันอย่างเจาะจง ดังนั้นบริการของธนาคารแรกจะทำการพิจารณาเงินกู้โดยแผนกซึ่งตรงกับความต้องการของลูกค้า เมื่อทุกแง่มุมของพฤติกรรมเชิงหน้าที่และกระบวนการภายในผ่านการพิจารณาเงื่อนไข บริการของธนาคารแรกจะถูกส่งเป็นผลลัพธ์ไปยังลูกค้า ด้วยวิธีการเดียวกันนี้บริการของธนาคารที่สองจะไม่เข้าคู่ เนื่องจากเงื่อนไขกระบวนการที่สองเกี่ยวกับวัตถุประสงค์การกู้ส่งผลให้กระบวนการพิจารณาเงินกู้จะทำการพิจารณาโดยสาขา ในทำนองเดียวกันบริการของธนาคารที่สามจะไม่เข้าคู่เนื่องจากกระบวนการพิจารณาเงินกู้จะทำการพิจารณาโดยสำนักงานใหญ่จากเงื่อนไขกระบวนการแรก

การค้นหาบริการจากข้อกำหนดกระบวนการจะมีประสิทธิภาพ เมื่อนำออนโทโลจีกระบวนการที่ใช้ร่วมกันภายในโดเมนมาเป็นพื้นฐานในการค้นหา ออนโทโลจีกระบวนการที่ใช้ร่วมกันภายในโดเมนจะมีข้อมูลเกี่ยวกับกระบวนการภายในและเงื่อนไขที่เกี่ยวข้อง ซึ่งสอดคล้องกับแนวคิดของแพทเทิร์นของกระบวนการทางธุรกิจ (Business Process Pattern) [17], [18] ผู้

ให้บริการจะประกาศบริการโดยสืบทอดมาจากออนโทโลจีกระบวนการซึ่งมีความเหมาะสมกับธุรกิจ และลูกค้าจะมีความรู้เกี่ยวกับบริการนั้นจากออนโทโลจีกระบวนการที่ใช้ร่วมกันภายในโดเมน ทั้งในส่วนพฤติกรรมเชิงหน้าที่และกระบวนการย่อยเพื่อนำไปใช้ในการค้นหา จากตัวอย่างข้างต้นลูกค้าจะทราบว่าการอนุมัติกู้เงินมีได้หลายระดับ และทราบถึงองค์ประกอบของเงื่อนไขการกู้เช่น รายได้ของลูกค้า จำนวนเงินกู้ และวัตถุประสงค์การกู้เงิน แม้ว่าในความเป็นจริงผู้ให้บริการจะไม่ได้ให้รายละเอียดเกี่ยวกับเงื่อนไขการพิจารณาเงินกู้แก่ลูกค้าโดยตรง อาจด้วยเหตุผลด้านความมั่นคง แต่ลูกค้าสามารถสร้างการสอบถามข้อมูลได้จากออนโทโลจีกระบวนการที่ใช้ร่วมกันภายในโดเมน อย่างไรก็ตามจากตัวอย่างข้างต้น ลูกค้าจะไม่ต้องทราบรายละเอียดเช่นบริการของธนาคารแรกตั้งเกณฑ์การพิจารณาโดยสำนักงานใหญ่ไว้สำหรับวงเงินกู้มากกว่า 1,000,000 บาท ซึ่งข้อมูลนี้ไม่จำเป็นสำหรับสร้างการสอบถามข้อมูลโดยลูกค้า

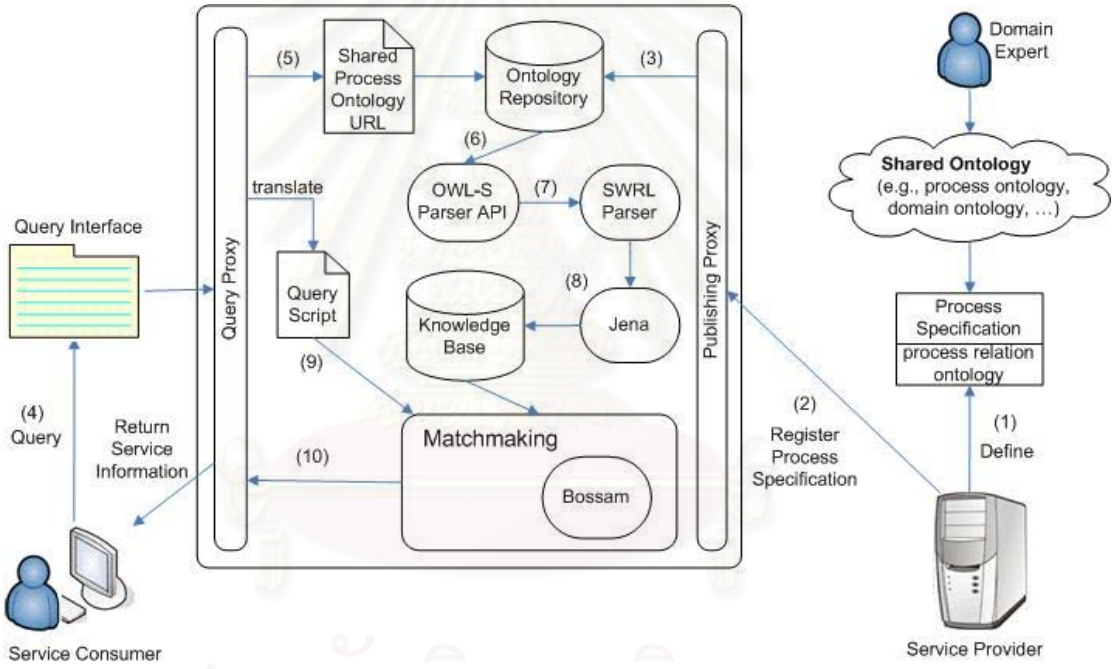
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4
สถาปัตยกรรมการค้นหาเว็บเซอร์วิสด้วยอาวล์-เอสโพรเซสโมเดล

ในบทนี้จะกล่าวถึงการออกแบบและพัฒนาสถาปัตยกรรมการค้นหาเว็บเซอร์วิสด้วยอาวล์-เอสโพรเซสโมเดล โดยสถาปัตยกรรมประกอบด้วยตัวกลาง (Mediator) ที่ทำหน้าที่รับการประกาศบริการและให้บริการค้นหาเว็บเซอร์วิส

4.1 การออกแบบการทำงานของตัวกลาง

การทำงานของตัวกลางสามารถแสดงเป็นเฟรมเวิร์กได้ดังรูปที่ 4.1



รูปที่ 4.1 เฟรมเวิร์กสำหรับการค้นหาบริการโดยข้อกำหนดอาวล์-เอสโพรเซสโมเดล

(1) ผู้ให้บริการสร้างข้อกำหนดกระบวนการอาวล์-เอสโพรเซสโมเดลสำหรับเว็บเซอร์วิสของตนเอง และออนโทโลจีความสัมพันธ์ของคลาสกระบวนการของบริการกับคลาสกระบวนการมาตรฐานโดยอ้างอิงจากออนโทโลจีกระบวนการที่ใช้งานร่วมกันภายในโดเมนซึ่งประกาศไว้โดยผู้เชี่ยวชาญของโดเมนนั้น (Domain Expert) การสร้างข้อกำหนดกระบวนการของผู้ให้บริการสามารถกระทำได้โดยใช้เครื่องมือสร้างออนโทโลจี โพรทีเจ (Protégé)

(2) ผู้ให้บริการประกาศข้อกำหนดกระบวนการผ่านตัวกลางที่สร้างขึ้น โดยกรอกข้อมูลยูอาร์แอล (URL) ของออนโทโลจีกระบวนการที่ใช้งานร่วมกัน ยูอาร์แอลของข้อกำหนด

กระบวนการ และยูอาร์แอลของข้อมูลแสดงความสัมพันธ์ของคลาสกระบวนการของบริการกับคลาสกระบวนการมาตรฐาน

(3) ส่วนของตัวกลางที่ทำหน้าที่รับการประกาศจากผู้ให้บริการ (Publishing Proxy) จะเก็บข้อมูลของผู้ให้บริการไว้ในฐานข้อมูลออนโทโลจี (Ontology Repository)

(4) เมื่อลูกค้าต้องการค้นหาบริการก็กระทำผ่านส่วนของตัวกลางที่ทำหน้าที่ค้นหาบริการ (Query Proxy) โดยกรอกข้อมูลยูอาร์แอลของออนโทโลจีกระบวนการที่ใช้งานร่วมกัน และข้อมูลการค้นหาบริการ

(5) ตัวกลางจะทำการค้นหาทุกบริการซึ่งอ้างอิงจากออนโทโลจีกระบวนการที่ใช้งานร่วมกันที่ลูกค้ากรอกข้อมูลค้นหา

(6) ทุกบริการที่ทำการค้นหาได้จะถูกประมวลผลโดยใช้ตัวแจงอาวล์-เอส ซึ่งจะได้ข้อมูลเงื่อนไขและกระบวนการต่างๆ

(7) ในส่วนของเงื่อนไขที่อ่านได้จะทำการประมวลผลโดยใช้ตัวแจงสเวิร์ล (SWRL Parser)

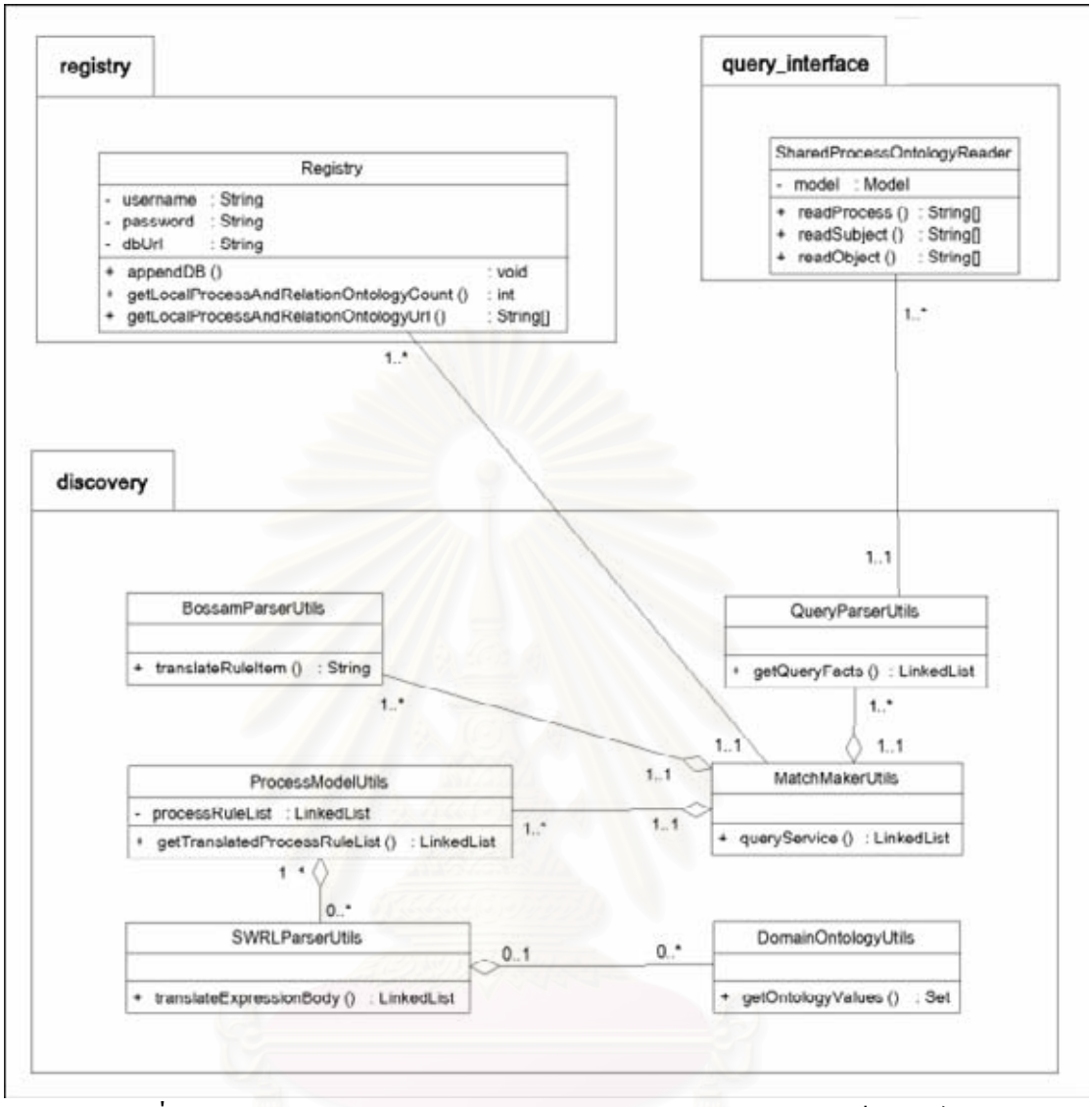
(8) ในกรณีที่เงื่อนไขมีความเกี่ยวข้องกับออนโทโลจีโดเมน ตัวกลางจะทำการประมวลผลโดยใช้เครื่องมืออนุมานออนโทโลจี ได้แก่ จีนา แล้วนำข้อมูลทุกอย่างที่อนุมานได้มาเก็บไว้เป็นองค์ความรู้ในฐานข้อมูล (Knowledge Base) ก่อน รูปแบบขององค์ความรู้จะเก็บไว้ในรูปของนิพจน์ของเครื่องมือหาเหตุผลบอสซาม

(9) ส่วนของตัวกลางที่รับการค้นหาจากลูกค้าจะอนุมานข้อมูลการค้นหาให้อยู่ในรูปแบบนิพจน์ที่เครื่องมือหาเหตุผลบอสซามสามารถเข้าใจได้ (Query Script)

(10) นำข้อมูลการค้นหา และข้อมูลของเว็บเซอร์วิสที่ได้ทำการอนุมานไว้ล่วงหน้าในฐานข้อมูลมาพิจารณาหาบริการที่ตรงกับความต้องการของลูกค้าโดยใช้เครื่องมือหาเหตุผลบอสซาม แล้วส่งรายละเอียดของเว็บเซอร์วิสที่ค้นพบกลับไปยังลูกค้า

4.2 การออกแบบตัวกลาง

การออกแบบตัวกลางสามารถออกแบบได้เป็นแผนภาพคลาส โดยแบ่งแผนภาพคลาสออกเป็นสามแพ็คเกจคือ แพ็คเกจการลงทะเบียน แพ็คเกจการค้นหา และแพ็คเกจการเข้าคู่ ดังรูปที่ 4.2 โดยแพ็คเกจการลงทะเบียนมีหน้าที่รับข้อมูลเว็บเซอร์วิสจากผู้ให้บริการนำมาเก็บไว้ในฐานข้อมูลออนโทโลจี แพ็คเกจการค้นหามีหน้าที่รับข้อมูลการค้นหาจากลูกค้าเพื่อนำมาใช้ในขั้นตอนการเข้าคู่ และแพ็คเกจการเข้าคู่ทำหน้าที่ประมวลผลข้อมูลการค้นหา และข้อมูลเว็บเซอร์วิสเพื่อหาเว็บเซอร์วิสที่ตรงกับความต้องการของลูกค้า



รูปที่ 4.2 แผนภาพคลาสของตัวกลางสำหรับการประกาศและค้นหาเว็บเซอร์วิส

4.3 เครื่องมือที่ใช้ในการพัฒนา

ในการพัฒนาตัวกลางในการประกาศและค้นหาเว็บเซอร์วิสนั้น ผู้วิจัยได้ใช้เครื่องมือในการพัฒนาดังนี้

1. จาวาเอสดีเค (Java SDK) รุ่น 1.5 เพื่อใช้เป็นตัวแปลโปรแกรม
2. อิคลิพซ์ (Eclipse) รุ่น 3.2 สำหรับเป็นเครื่องมือและสภาพแวดล้อมสำหรับการพัฒนาตัวกลาง
3. เจเอสพี (JSP) รุ่น 2.4 สำหรับเป็นภาษาในการเขียนสคริปต์
4. อาพาเช ทอมแคท (Apache Tomcat) รุ่น 5.0 สำหรับเป็นเว็บคอนเทนเนอร์
5. มายเอสคิวแอล (Mysql) รุ่น 5.0.24 สำหรับเป็นฐานข้อมูล

6. โพรทีเจ รุ่น 3.1 สำหรับสร้างแฟ้มข้อมูลอาวล์ของผู้ให้บริการ และออนโทโลจีโดเมน
7. จีนา รุ่น 2.1 สำหรับประมวลผลออนโทโลจี
8. บอสซาม รุ่น 0.8 สำหรับประมวลผลการเข้าคู่ของบริการ
9. ตัวแจงอาวล์-เอส เอพีไอ รุ่น 1.0 สำหรับประมวลผลแฟ้มข้อมูลอาวล์-เอส

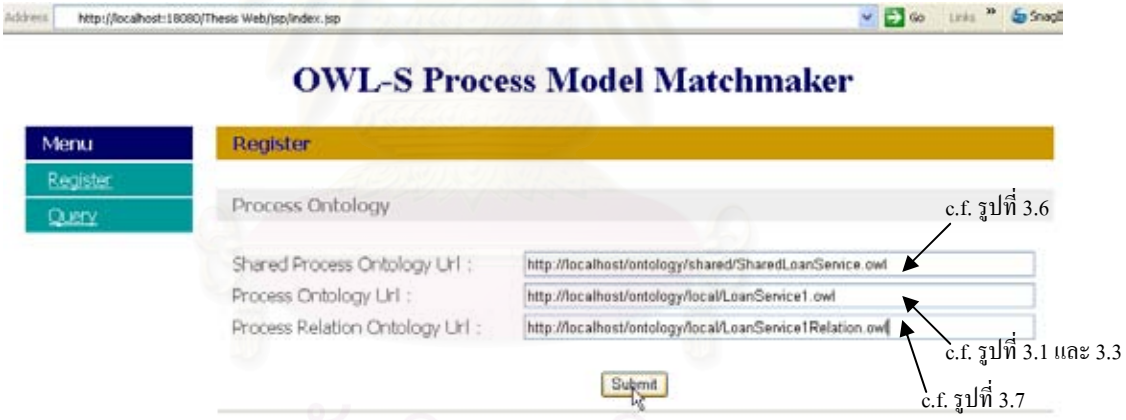
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5
การทดสอบตัวกลางในการค้นหาเว็บเซอร์วิสด้วยกรณีศึกษา

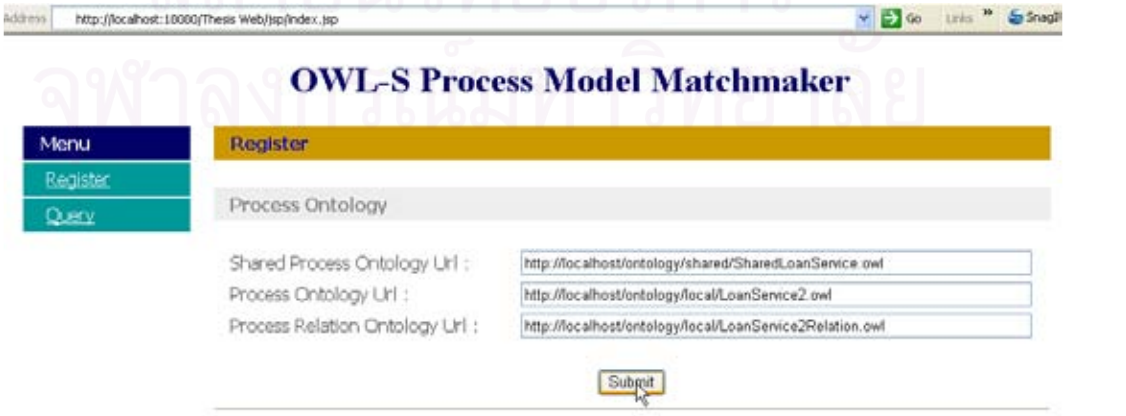
การทดสอบด้วยกรณีศึกษานั้นจะทดสอบด้วยกรณีศึกษา 3 กรณีคือ กรณีศึกษาที่มีโครงสร้างกระแสงานแบบตัดสินใจ กรณีศึกษาที่มีโครงสร้างแบบทำงานพร้อมกันและทำงานแบบลำดับ และกรณีศึกษาที่มีโครงสร้างแบบทำงานวนซ้ำและทำงานแบบลำดับ โดยแต่ละกรณีศึกษาจะแสดงการประกาศบริการ และการค้นหาโดยละเอียดด้วยตัวกลางที่พัฒนาขึ้น

5.1 กรณีศึกษาที่มีโครงสร้างกระแสงานแบบตัดสินใจ

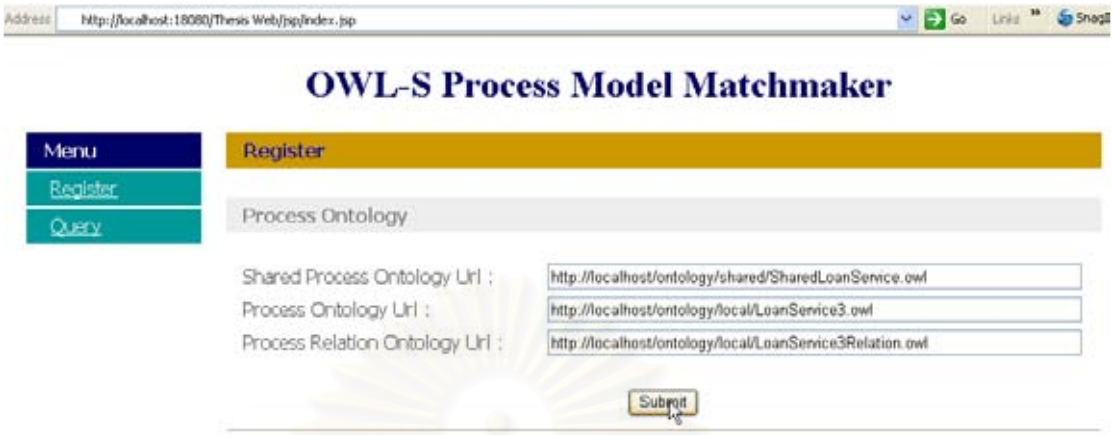
กรณีศึกษานี้จะใช้ตัวอย่างของบริการด้านการกู้เงินของธนาคาร 3 บริการเช่นเดียวกับในหัวข้อที่ 3.1 และลูกค้ามีความต้องการค้นหาบริการเช่นเดียวกับในหัวข้อที่ 3.3 โดยขั้นตอนแรกผู้ให้บริการทั้งสามต้องทำการลงทะเบียนบริการของตนผ่านตัวกลาง โดยทั้งสามบริการอ้างอิงออนโทโลจีกระบวนการที่ใช้ร่วมกันคือ http://localhost/ontology/shared/SharedLoanService.owl ดังรูปที่ 5.1 – 5.3



รูปที่ 5.1 การลงทะเบียนบริการกู้เงินของธนาคารแรก

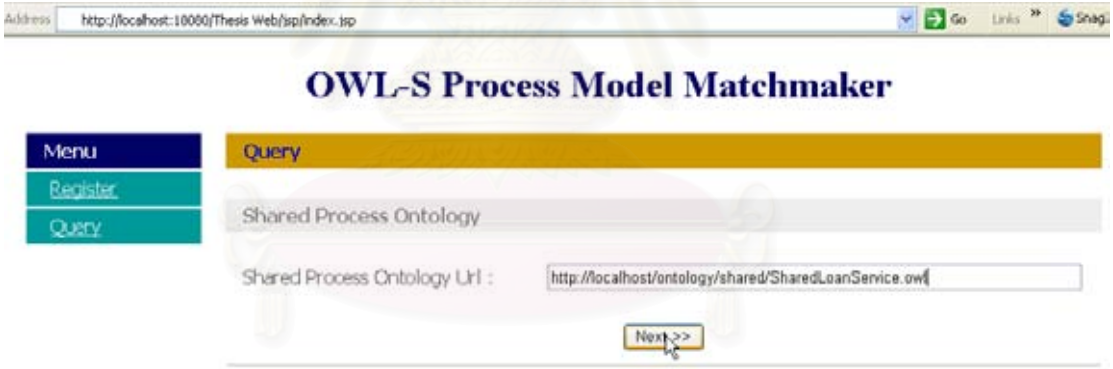


รูปที่ 5.2 การลงทะเบียนบริการกู้เงินของธนาคารที่สอง



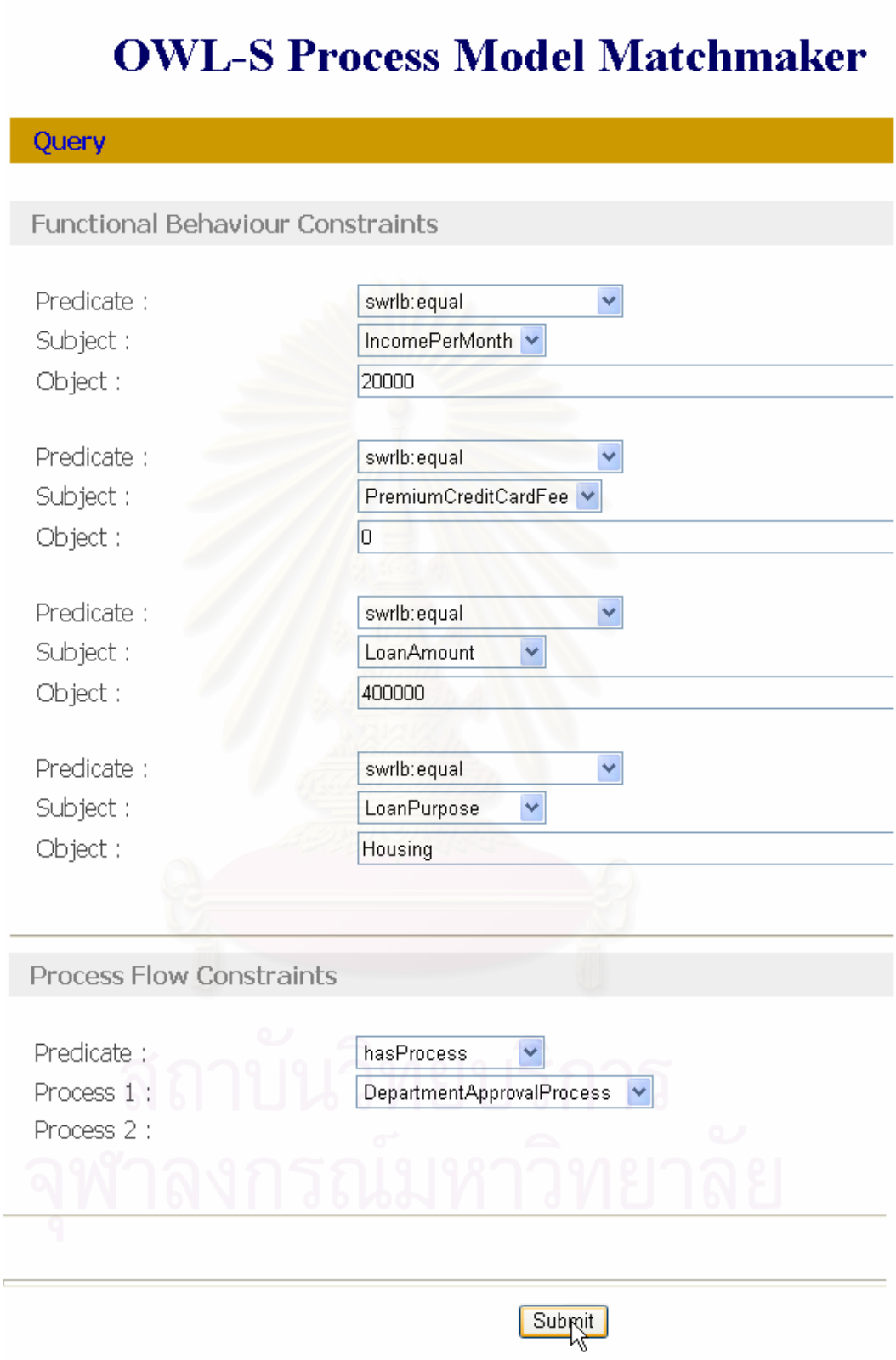
รูปที่ 5.3 การลงทะเบียนบริการกู้เงินของธนาคารที่สาม

จากนั้นในการค้นหาจะต้องทำการกรอกข้อมูลยูอาร์แอลของออนโทโลจีกระบวนการที่ใช้ร่วมกันภายในโดเมนกู้เงิน เพื่อระบุว่าต้องการค้นหาเว็บเซอร์วิสที่มีกระบวนการเป็นไปตามโดเมนนี้ ดังรูปที่ 5.4



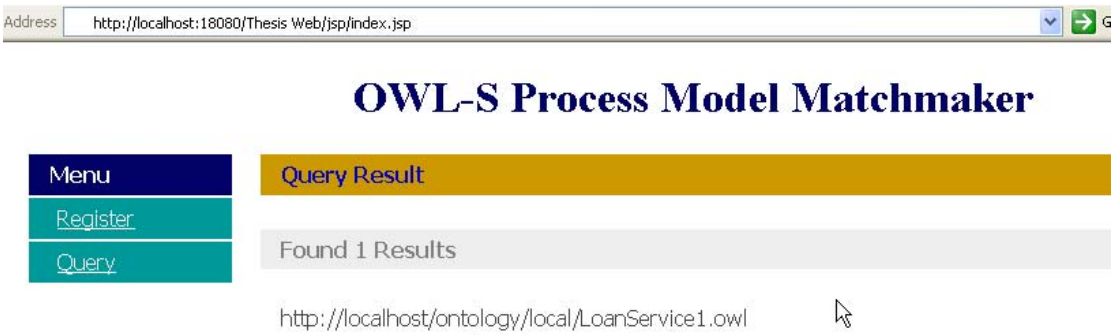
รูปที่ 5.4 การกรอกข้อมูลยูอาร์แอลของออนโทโลจีกระบวนการที่ใช้ร่วมกันของบริการกู้เงิน

จากนั้นจึงกรอกข้อมูลการค้นหาที่ต้องการตามตัวอย่างในหัวข้อที่ 3.3 ซึ่งลูกค้าต้องการกู้เงินจากธนาคารเป็นจำนวน 400,000 บาท เพื่อไปซื้อบ้าน โดยต้องการให้การพิจารณาอนุมัติเงินกู้ทำโดยแผนกเงินกู้ ซึ่งจะทำให้ใช้เวลาน้อยที่สุดในการทราบผล และลูกค้าต้องการให้ธนาคารอนุมัติการเป็นสมาชิกบัตรเครดิตโดยไม่เสียค่าธรรมเนียมด้วย โดยลูกค้ามีรายได้ 20,000 บาทต่อเดือน ดังรูปที่ 5.5



รูปที่ 5.5 การกรอกข้อมูลการค้นหาบริการกู้เงิน

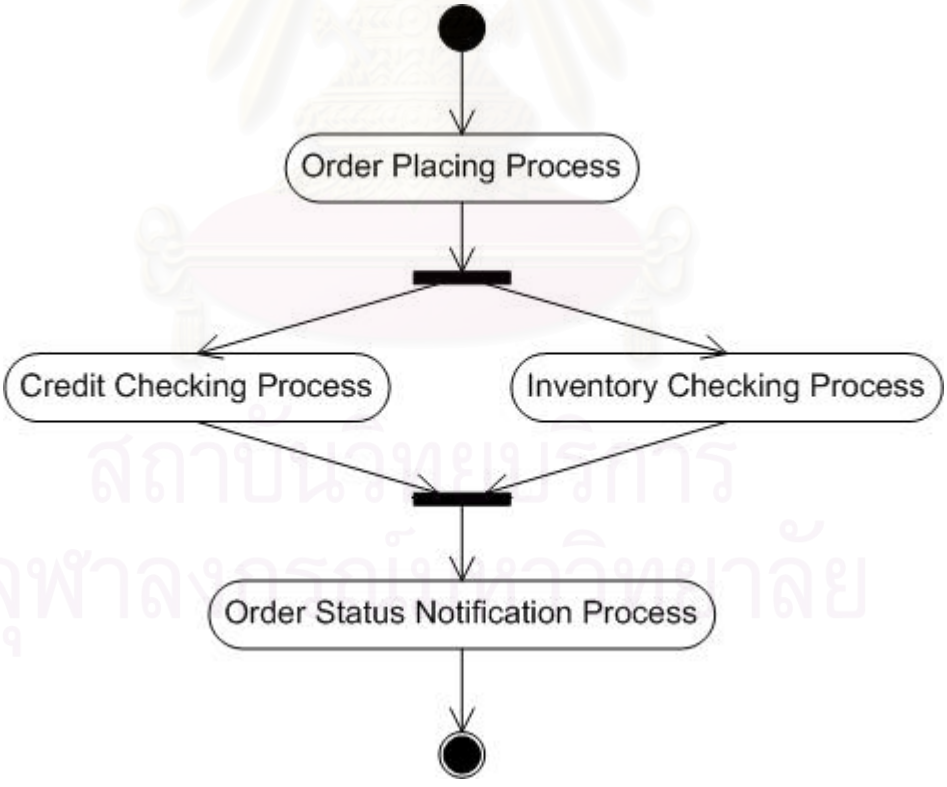
หลังจากกดปุ่มค้นหาแล้วจะได้ผลการค้นหาคือบริการของธนาคารแรกดังรูปที่ 5.6



รูปที่ 5.6 ผลลัพธ์ของการค้นหาบริการกู้เงิน

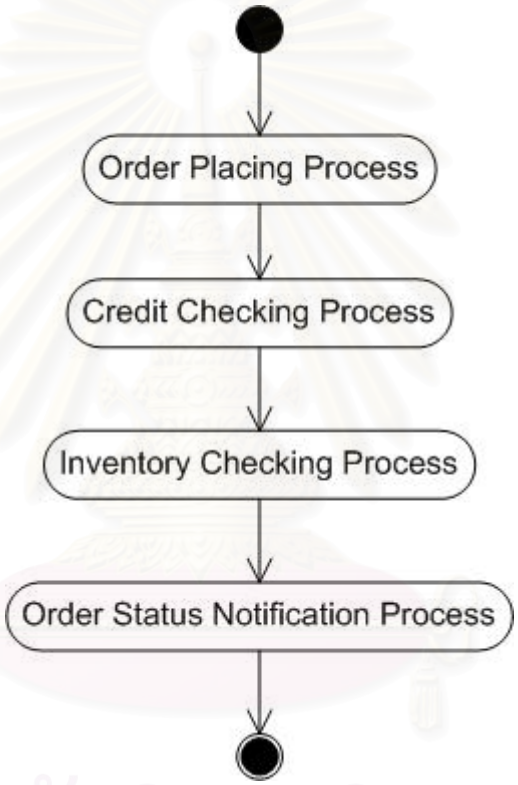
5.2 กรณีศึกษาที่มีโครงสร้างกระแสงานแบบทำงานพร้อมกันและทำงานแบบลำดับ

กรณีศึกษานี้จะใช้ตัวอย่างของบริการสั่งซื้อไวน์ 3 บริการ โดยบริการทั้งสามอ้างอิงออนโทโลจีกระบวนการที่ใช้ร่วมกันคือ http://localhost/ontology/shared/SharedOrderService.owl สำหรับบริการแรกจะมีเงื่อนไขก่อนการทำงานคือ ลูกค้าต้องมีอายุมากกว่าหรือเท่ากับ 18 ปี และส่วนกระแสงานสามารถแสดงได้ดังรูปที่ 5.7



รูปที่ 5.7 กระแสงานของบริการสั่งซื้อไวน์บริการแรก

จากรูปที่ 5.7 บริการแรกเมื่อผ่านการตรวจสอบเงื่อนไขก่อนการทำงาน จะเข้าสู่กระบวนการบันทึกการสั่งซื้อ (Order Placing) จากนั้นจะเข้าสู่กระบวนการตรวจสอบเครดิตของลูกค้า (Credit Checking) ซึ่งทำงานพร้อมกันกับกระบวนการตรวจสอบคลังสินค้า (Inventory Checking) และจากนั้นจะเข้าสู่กระบวนการแจ้งสถานะการสั่งซื้อ (Order Status Notification) บริการที่สองจะคล้ายกับบริการแรก ต่างกันตรงเงื่อนไขก่อนการทำงานซึ่งกำหนดว่าลูกค้าต้องมีอายุมากกว่าหรือเท่ากับ 20 ปี และบริการที่สามมีเงื่อนไขก่อนการทำงานเช่นเดียวกับบริการแรก แต่มีกระแสงานดังแสดงดังรูปที่ 5.8

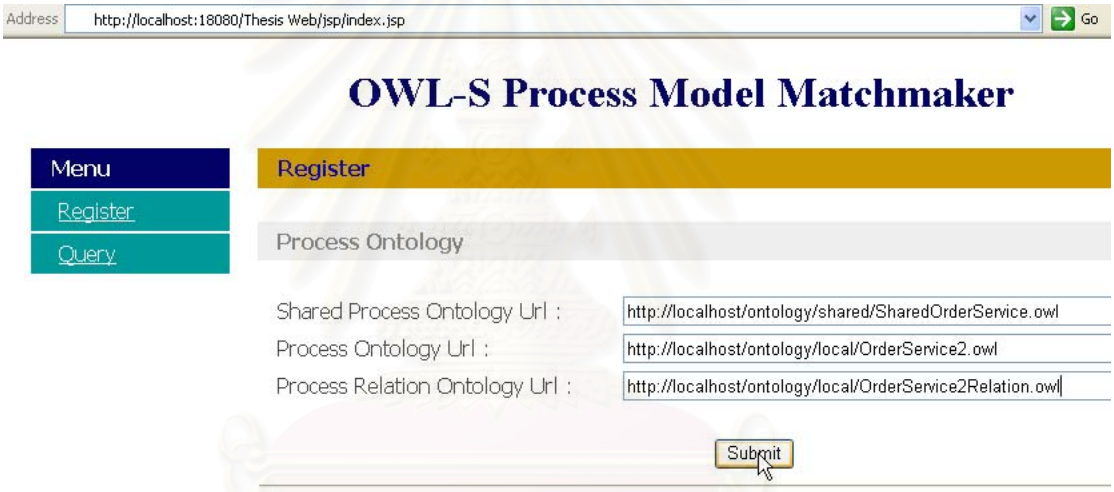


รูปที่ 5.8 กระแสงานของบริการสั่งซื้อไวน์บริการที่สาม

จากรูปที่ 5.8 บริการที่สามเมื่อผ่านการตรวจสอบเงื่อนไขก่อนการทำงาน จะเข้าสู่กระบวนการสั่งซื้อสินค้า กระบวนการตรวจสอบเครดิตลูกค้า กระบวนการตรวจสอบคลังสินค้า และกระบวนการแจ้งสถานะการสั่งซื้อ ตามลำดับ

ในการประกาศบริการ ผู้ให้บริการทั้งสามจะลงทะเบียนผ่านตัวกลางดังรูปที่ 5.9 – 5.11

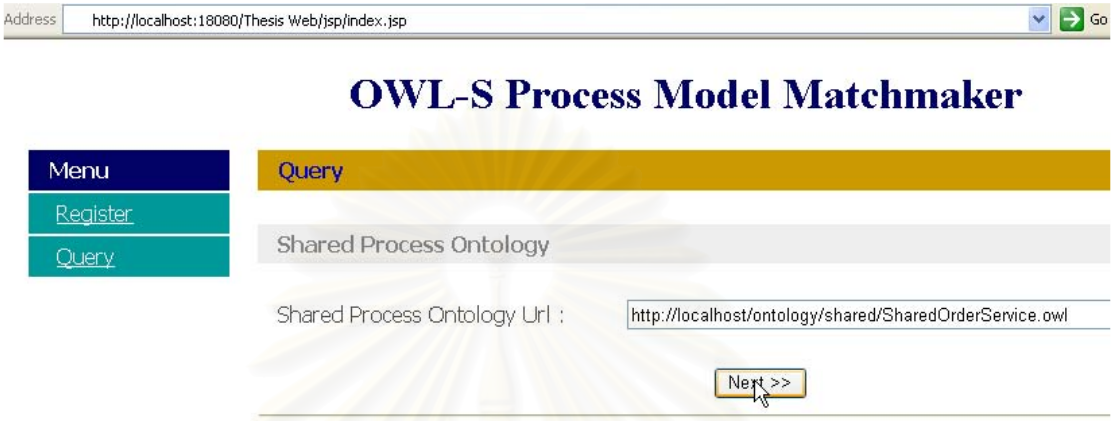
รูปที่ 5.9 การลงทะเบียนของบริการสั่งซื้อไวน์บริการแรก



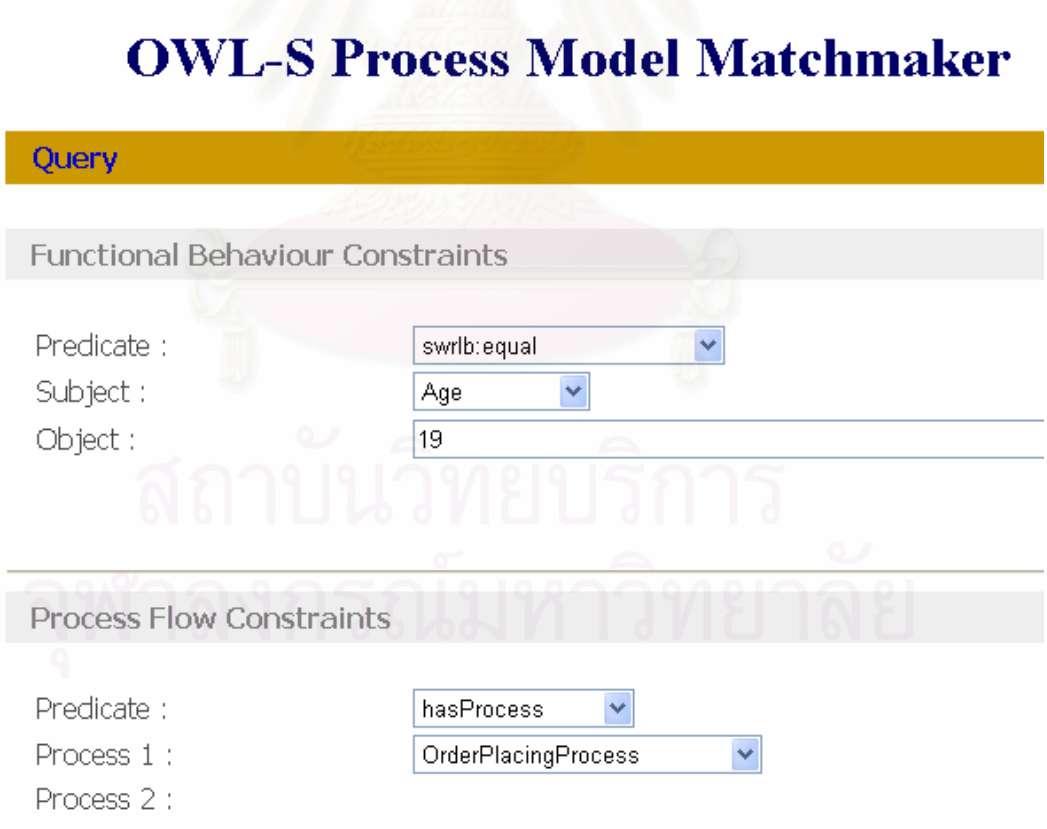
รูปที่ 5.10 การลงทะเบียนของบริการสั่งซื้อไวน์บริการที่สอง

รูปที่ 5.11 การลงทะเบียนของบริการสั่งซื้อไวน์บริการที่สาม

สำหรับการค้นหานั้นในกรณีศึกษาจะมีตัวอย่างการค้นหา 2 ตัวอย่าง ตัวอย่างแรกเป็นกรณีที่ลูกค้ามีอายุ 19 ปีและต้องการค้นหาบริการสั่งซื้อไวน์ การค้นหาต้องกรอกข้อมูลยูอาร์แอลของออนโทโลจีกระบวนการที่ใช้ร่วมกัน ดังรูปที่ 5.12 และกรอกข้อมูลค้นหาดังรูปที่ 5.13



รูปที่ 5.12 การกรอกข้อมูลยูอาร์แอลของออนโทโลจีกระบวนการที่ใช้ร่วมกันของบริการสั่งซื้อไวน์



รูปที่ 5.13 การกรอกข้อมูลค้นหาบริการสั่งซื้อไวน์ของกรณีแรก

จากข้อมูลการค้นหา สามารถเขียนเซตการค้นหาได้เป็น

$\mathbb{Q} = \{\text{hasAge(Age, Equal, 19)}^{G}$,

$\text{hasProcess(Process, OrderPlacingProcess)}^{P}\}$

การพิจารณาในส่วนของพฤติกรรมเชิงหน้าที่จะพิจารณาเงื่อนไขก่อนการทำงานซึ่งเป็นการพิจารณาแบบช่วงตัวเลข ซึ่งจะเห็นว่ามีบริการแรกและบริการที่สามเท่านั้นที่มีการเข้าคู่อย่างถูกต้อง การพิจารณาเงื่อนไขก่อนการทำงานแสดงได้ดังตารางที่ 5.1

ตารางที่ 5.1 ตารางเปรียบเทียบพฤติกรรมเชิงหน้าที่ของบริการสั่งซื้อไวน์ทั้งสามกับข้อความค้นหา

ส่วนของข้อมูล	ข้อมูลการค้นหา	บริการ	ข้อมูลบริการ	ลักษณะการเข้าคู่
เงื่อนไขก่อนการทำงาน	Age = 19	บริการแรก	Age $\geq$ 18	เข้าคู่อย่างถูกต้อง
		บริการที่สอง	Age $\geq$ 20	ไม่เข้าคู่
		บริการที่สาม	Age $\geq$ 18	เข้าคู่อย่างถูกต้อง

จากนั้นบริการแรกและบริการที่สามจะเข้าสู่การพิจารณาส่วนของกระบวนการภายใน จากบริการแรก จะได้กฎสำหรับกระบวนการภายในดังนี้

→ hasProcess(Process, OrderPlacingProcess);

→ hasProcess(Process, CreditCheckingProcess);

→ hasProcess(Process, InventoryCheckingProcess);

→ hasProcess(Process, OrderStatusNotificationProcess);

→ hasParallelProcess(CreditCheckingProcess, InventoryCheckingProcess);

→ hasParallelProcess(InventoryCheckingProcess, CreditCheckingProcess);

→ hasSequenceProcess(OrderPlacingProcess, CreditCheckingProcess);

→ hasSequenceProcess(OrderPlacingProcess, InventoryCheckingProcess);

→ hasSequenceProcess(CreditCheckingProcess, OrderStatusNotificationProcess);

→ hasSequenceProcess(InventoryCheckingProcess, OrderStatusNotificationProcess);

ส่วนบริการที่สามจะได้กฎสำหรับกระบวนการภายในดังนี้

→ hasProcess(Process, OrderPlacingProcess);

→ hasProcess(Process, CreditCheckingProcess);

→ hasProcess(Process, InventoryCheckingProcess);

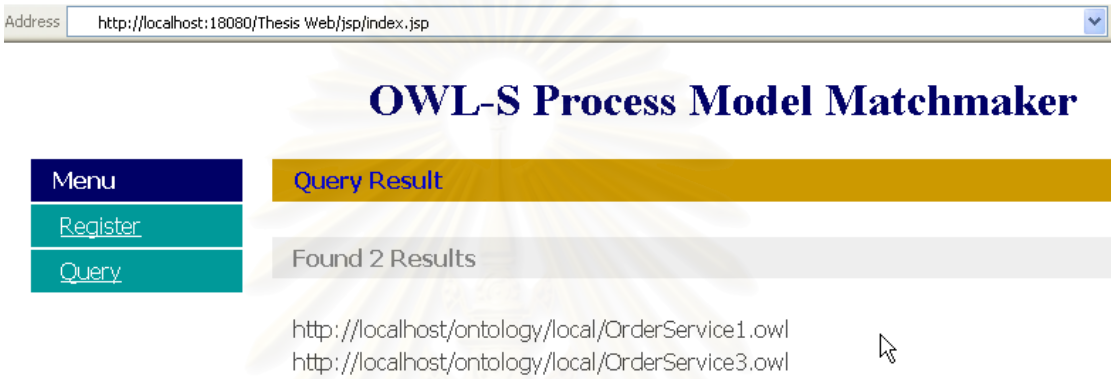
→ hasProcess(Process, OrderStatusNotificationProcess);

→ hasSequenceProcess(OrderPlacingProcess, CreditCheckingProcess);

→ hasSequenceProcess(CreditCheckingProcess, InventoryCheckingProcess);

→ hasSequenceProcess(InventoryCheckingProcess, OrderStatusNotificationProcess);

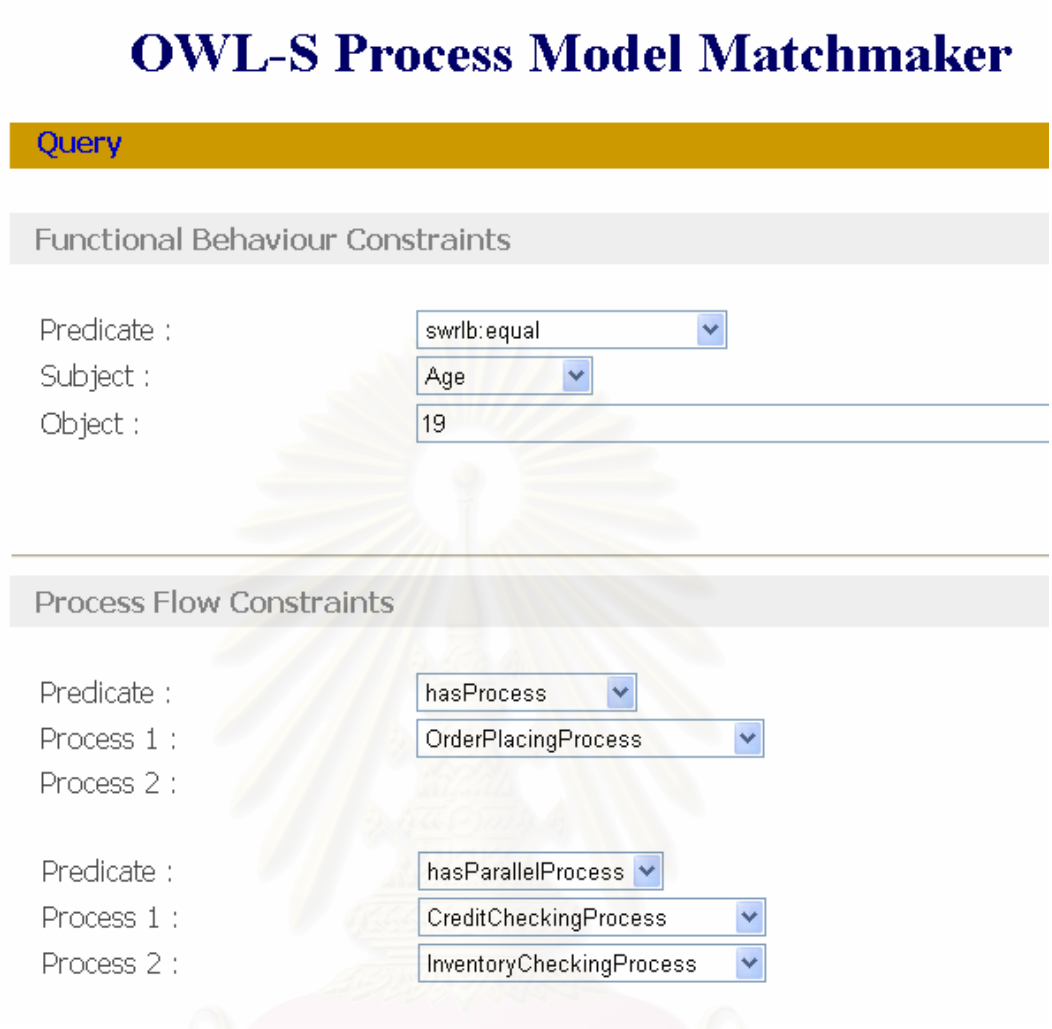
จะเห็นว่าทั้งบริการแรกและบริการที่สามมีกฎสำหรับกระบวนการภายใน → hasProcess(Process, OrderPlacingProcess); ซึ่งตรงกับการค้นหาทำให้มีการเข้าคู่กันอย่างถูกต้อง ส่งผลให้การค้นหาในกรณีศึกษานี้ ข้อมูลของบริการแรกและบริการที่สามจะถูกส่งไปเป็นผลลัพธ์ให้ลูกค้า ผลการค้นหาแสดงดังรูปที่ 5.14



รูปที่ 5.14 ผลการค้นหาบริการสั่งซื้อไวน์ของกรณีแรก

การค้นหากรณีที่สอง เป็นกรณีที่ลูกค้ามีอายุ 19 ปี ต้องการค้นหาบริการสั่งซื้อสินค้าไวน์ และต้องการบริการที่มีการตรวจสอบเครดิตลูกค้าไปพร้อมกับการตรวจสอบคลังสินค้าเลย ซึ่งจะทำให้การสั่งซื้อทำได้เร็ว ใช้เวลาน้อย ในการค้นหาผ่านตัวกลาง ลูกค้าต้องกรอกข้อมูลยูอาร์แอลของออนโทโลจีกระบวนการที่ใช้ร่วมกัน ดังรูปที่ 5.12 และกรอกข้อมูลการค้นหาดังรูปที่ 5.15

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.15 การกรอกข้อมูลค้นหาบริการสั่งซื้อไวน์ของกรณีที่สอง

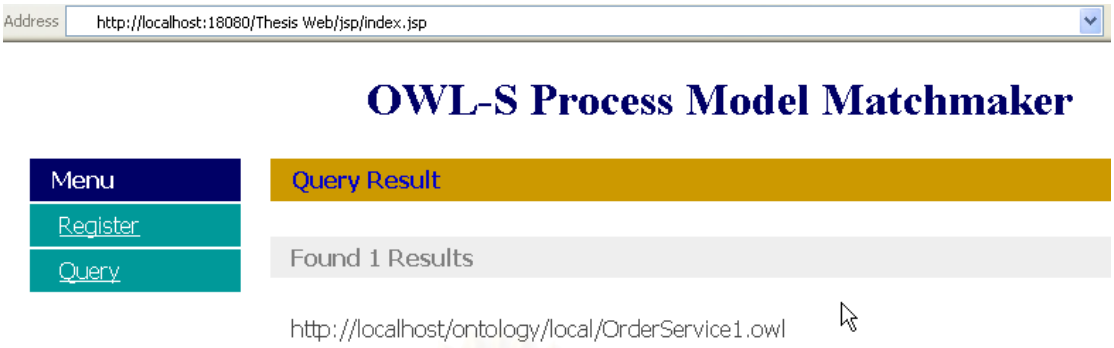
จากข้อมูลการค้นหา สามารถเขียนเซตการค้นหาได้เป็น

$\mathbb{Q} = \{\text{hasAge(Age, Equal, 19)}^{G}$,

$\text{hasProcess(Process, OrderPlacingProcess)}^{P}$,

$\text{hasParallelProcess(CreditCheckingProcess, InventoryCheckingProcess)}^{P}\}$

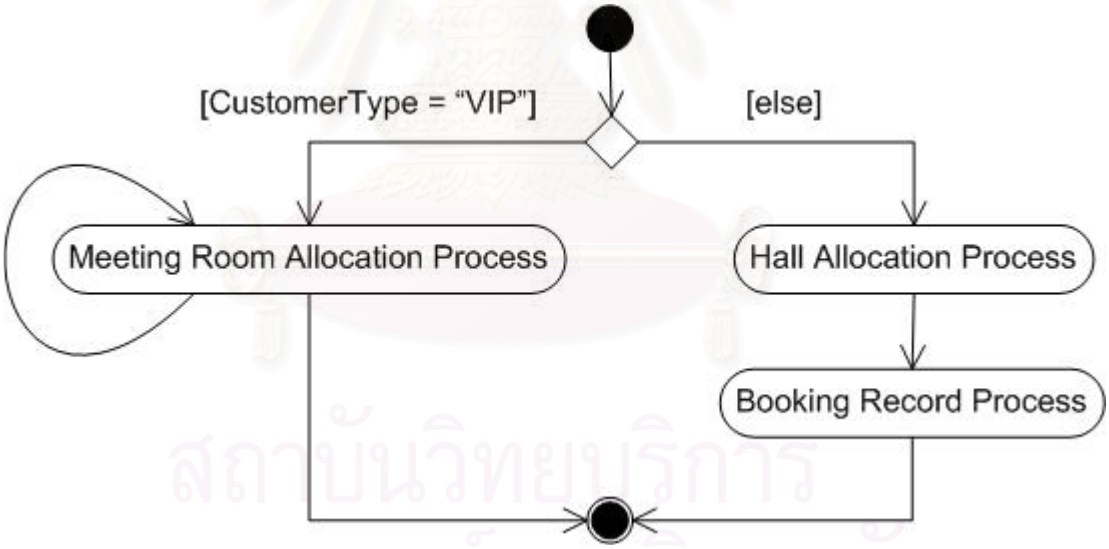
ในการพิจารณาพฤติกรรมเชิงหน้าที่นั้นจะได้ผลเช่นเดียวกับตารางที่ 5.1 นั่นคือบริการแรกและบริการที่สามมีการเข้าคู่อย่างถูกต้อง เนื่องจากลูกค้ามีอายุมากกว่าหรือเท่ากับเงื่อนไขก่อนการทำงานที่ทั้งสองบริการกำหนด จากนั้นจึงพิจารณาส่วนของกระบวนการภายในจะเห็นว่ามีเพียงบริการแรกเท่านั้นที่มีกฎ → hasParallelProcess(CreditCheckingProcess, InventoryChecking Process); ซึ่งตรงกับข้อมูลการค้นหา จึงทำให้ข้อมูลของบริการแรกนี้ถูกส่งไปเป็นผลลัพธ์ยังลูกค้า ผลการค้นหาแสดงได้ดังรูปที่ 5.16



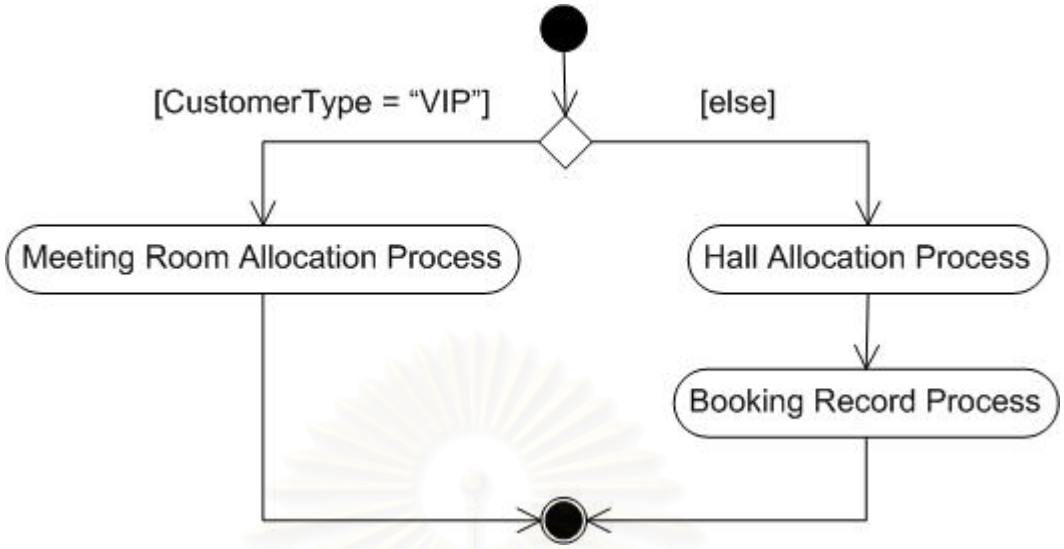
รูปที่ 5.16 ผลการค้นหาบริการสั่งซื้อไวน์ของกรณีที่สอง

5.3 กรณีศึกษาที่มีโครงสร้างกระแสงานแบบทำงานวนซ้ำและทำงานแบบลำดับ

กรณีศึกษานี้จะใช้ตัวอย่างของบริการจองโรงแรม 2 บริการ ดังรูปที่ 5.17 – 5.18 โดยกรณีศึกษานี้สนใจเฉพาะส่วนของโครงสร้างกระแสงานเท่านั้น บริการทั้งสองอ้างอิงออนโทโลจีกระบวนการที่ใช้ร่วมกันคือ http://localhost/ontology/shared/SharedMeetingService.owl



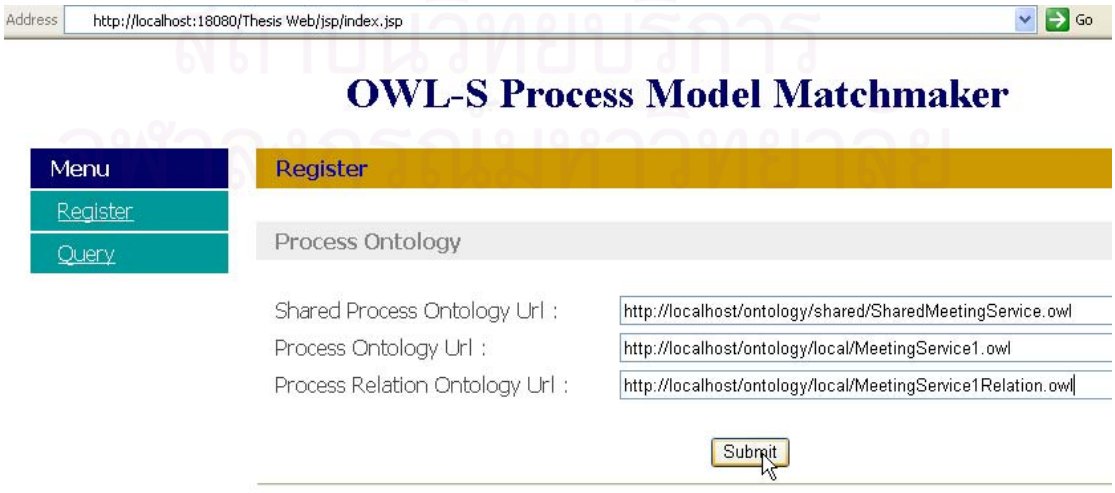
รูปที่ 5.17 กระแสงานของบริการจองโรงแรมบริการแรก



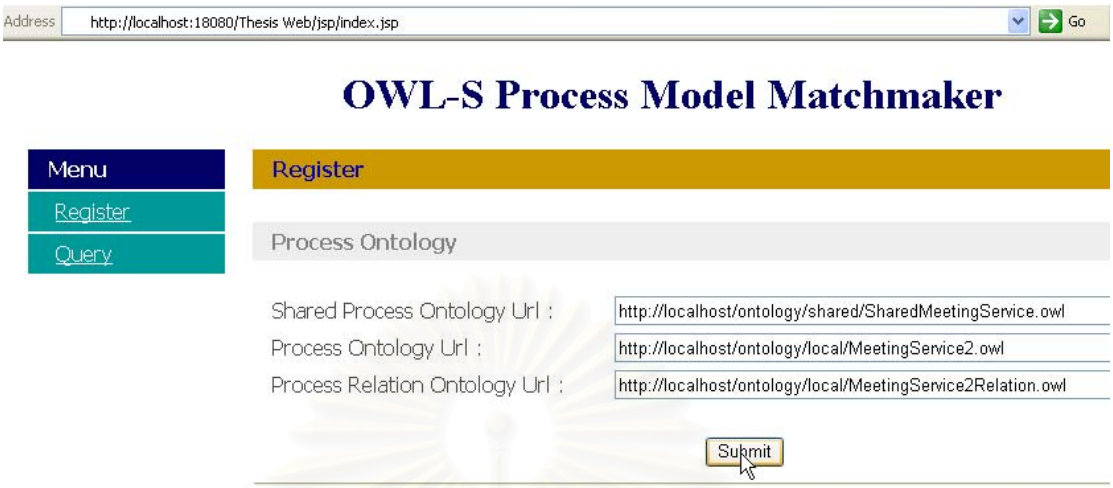
รูปที่ 5.18 กระแสงานของบริการจองโรงแรมบริการที่สอง

จากรูปที่ 5.17 บริการนี้ประกอบด้วยเงื่อนไขควบคุมกระแสงานซึ่งทำการตรวจสอบว่าลูกค้าเป็นลูกค้าวีไอพีหรือไม่ หากลูกค้าเป็นลูกค้าวีไอพีจะเข้าสู่กระบวนการจองห้องประชุม (Meeting Room Allocation) ซึ่งเป็นกระบวนการที่มีการทำงานแบบวนซ้ำ เนื่องจากต้องทำการหาห้องประชุมที่ว่างไปทีละห้อง หากลูกค้าเป็นลูกค้าประเภทอื่นจะเข้าสู่กระบวนการจองส่วนของห้องโถงเพื่อการประชุม (Hall Allocation) ซึ่งเป็นห้องประชุมรวมสำหรับลูกค้าทั่วไป และสามารถแบ่งห้องเป็นส่วนๆ เพื่อจัดการประชุมได้ จากนั้นจะเข้าสู่กระบวนการจัดเก็บข้อมูลการจองห้องประชุม (Booking Record) และจากรูปที่ 5.18 บริการที่สองจะต่างจากบริการแรกตรงกระบวนการจองห้องประชุมไม่ได้มีการทำงานแบบวนซ้ำเนื่องจากโรงแรมนี้มีห้องประชุมเพียงห้องเดียว

ในส่วนการประกาศบริการ แสดงได้ดังรูปที่ 5.19 – 5.20

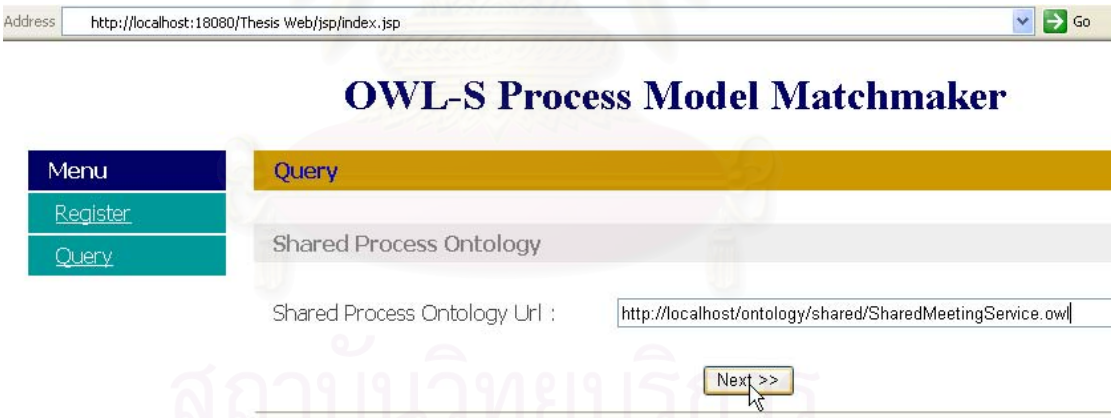


รูปที่ 5.19 การลงทะเบียนของบริการจองโรงแรมบริการแรก



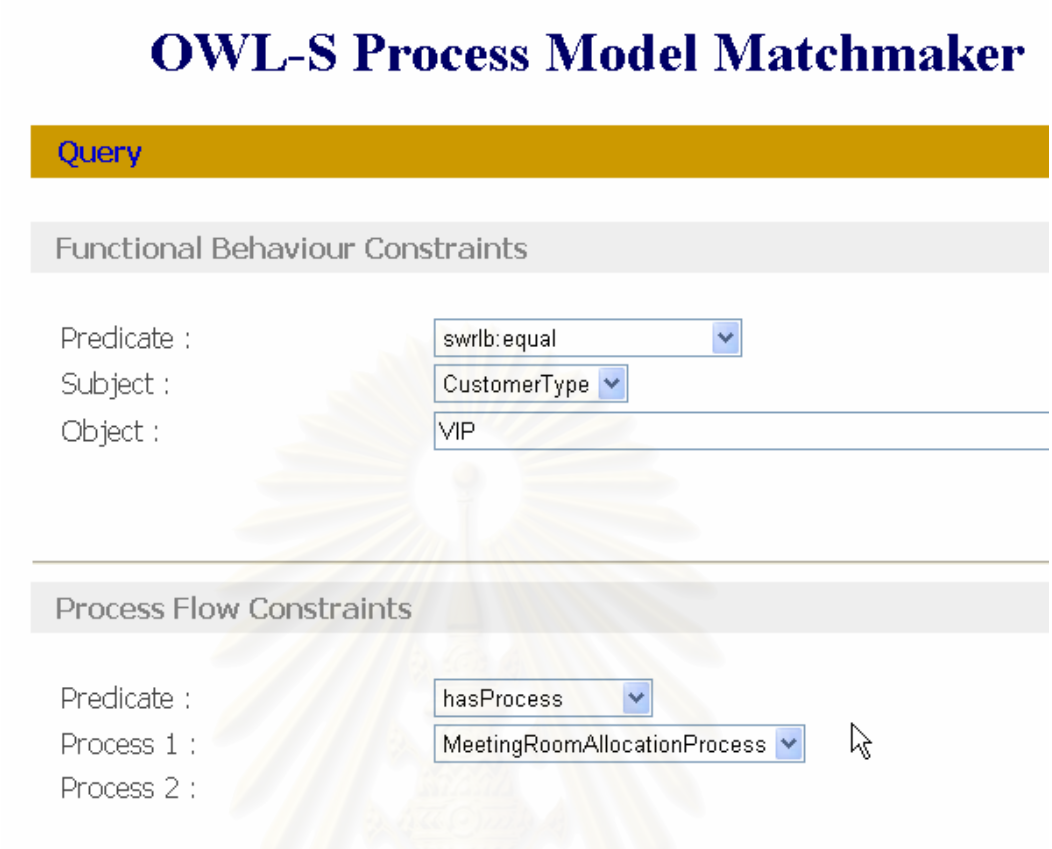
รูปที่ 5.20 การลงทะเบียนของบริการจองโรงแรมบริการที่สอง

สำหรับตัวอย่างการค้นหาตัวอย่างแรก จะเป็นกรณีที่ลูกค้าเป็นลูกค้าวีไอพี ต้องการค้นหาบริการจองโรงแรมซึ่งมีห้องประชุมส่วนตัวไว้บริการ การค้นหาต้องกรอกข้อมูลยูอาร์แอลของออนโทโลจีกระบวนการที่ใช้ร่วมกัน ดังรูปที่ 5.21 และกรอกข้อมูลค้นหาดังรูปที่ 5.22



รูปที่ 5.21 การกรอกข้อมูลยูอาร์แอลของออนโทโลจีกระบวนการที่ใช้ร่วมกัน
ของบริการจองโรงแรม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.22 การกรอกข้อมูลค้นหาบริการจองโรงแรมของกรณีแรก

จากข้อมูลการค้นหาจะสามารถเขียนเซตการค้นหาได้เป็น

$\mathbb{Q} = \{\text{hasCustomerType(CustomerType, VIP)}^{G},$

$\text{hasProcess(Process, MeetingRoomAllocationProcess)}^{P}\}$

เนื่องจากในกรณีศึกษานี้ไม่มีส่วนของพฤติกรรมเชิงหน้าที่จึงเข้าสู่การพิจารณากระบวนการภายใน จากบริการแรก จะได้กฎสำหรับกระบวนการภายในดังนี้

hasCustomerType(CustomerType, VIP) →
hasProcess(Process, MeetingRoomAllocationProcess);

hasCustomerType(CustomerType, VIP) →
hasRepetitiveProcess(Process, MeetingRoomAllocationProcess);

! hasCustomerType(CustomerType, VIP) →
hasProcess(Process, HallAllocationProcess);

! hasCustomerType(CustomerType, VIP) →
hasProcess(Process, BookingRecordProcess);

! hasCustomerType(CustomerType, VIP) →
hasSequenceProcess(HallAllocationProcess, BookingRecordProcess);

สำหรับบริการที่สองจะได้กฎสำหรับกระบวนการภายในดังนี้

hasCustomerType(CustomerType, VIP) →

hasProcess(Process, MeetingRoomAllocationProcess);

! hasCustomerType(CustomerType, VIP) →

hasProcess(Process, HallAllocationProcess);

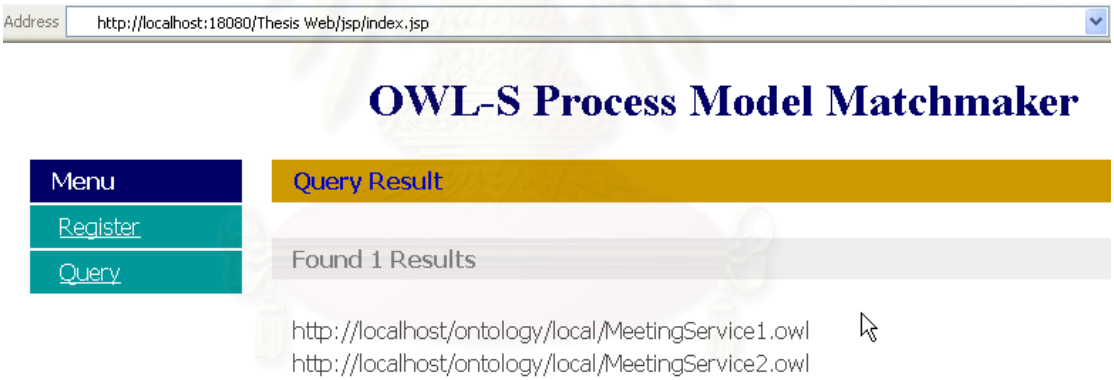
! hasCustomerType(CustomerType, VIP) →

hasProcess(Process, BookingRecordProcess);

! hasCustomerType(CustomerType, VIP) →

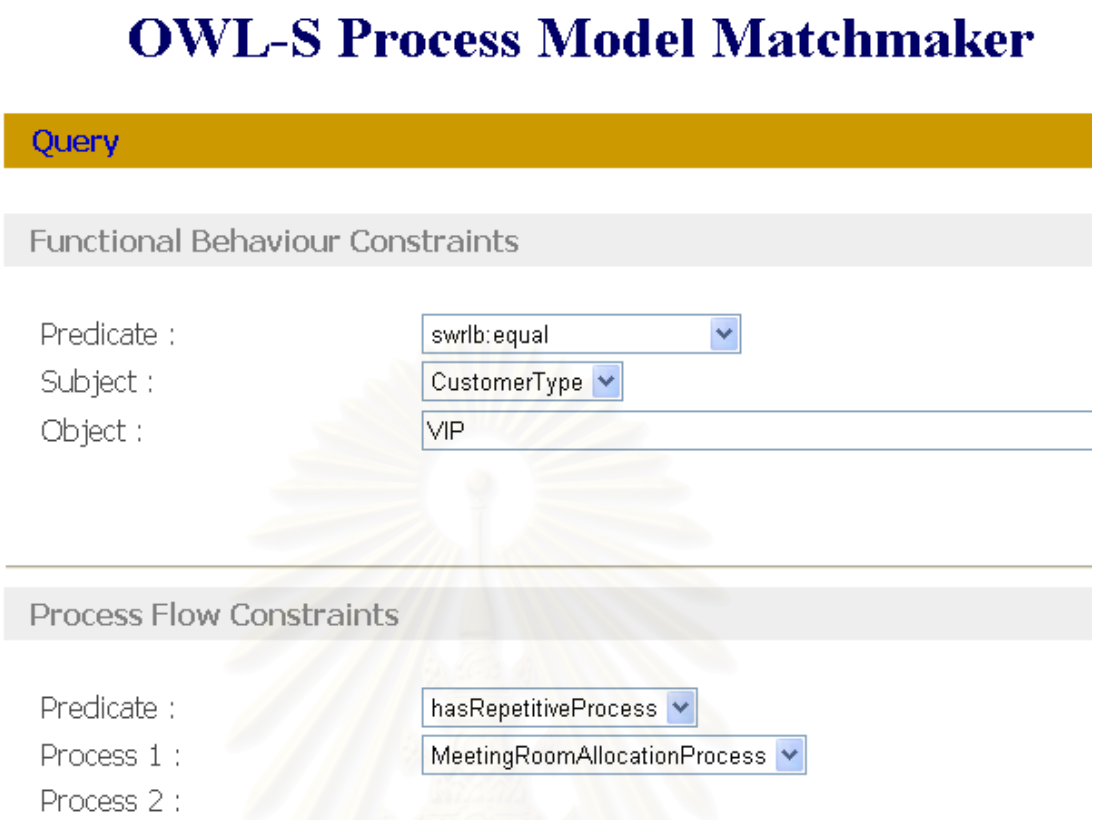
hasSequenceProcess(HallAllocationProcess, BookingRecordProcess);

จะเห็นว่าทั้งบริการแรกและบริการที่สองมีกฎสำหรับกระบวนการภายใน hasCustomer Type(CustomerType, VIP) → hasProcess(Process, MeetingRoomAllocationProcess); ซึ่งตรงกับการค้นหาทำให้มีการเข้าคู่กันอย่างถูกต้อง ส่งผลให้การค้นหาในกรณีศึกษานี้ ข้อมูลของบริการแรกและบริการที่สองจะถูกส่งไปเป็นผลลัพธ์ให้ลูกค้า ผลการค้นหาแสดงดังรูปที่ 5.23



รูปที่ 5.23 ผลการค้นหาบริการจองโรงแรมของกรณีแรก

การค้นหากรณีที่สอง เป็นกรณีที่ลูกค้าเป็นลูกค้าวีไอพี ต้องการค้นหาบริการจองโรงแรมซึ่งมีห้องประชุมส่วนตัวมากกว่าหนึ่งห้อง เพื่อจัดการประชุมได้เพียงพอกับหน่วยงานของตนเองที่มีมากกว่าหนึ่งหน่วยงาน ซึ่งก็คือบริการนั้นต้องมีกระบวนการจองห้องประชุมที่ทำงานแบบวนซ้ำ การค้นหาต้องกรอกข้อมูลยูอาร์แอลของออนโทโลจีกระบวนการที่ใช้ร่วมกัน ดังรูปที่ 5.21 และกรอกข้อมูลการค้นหาดังรูปที่ 5.24

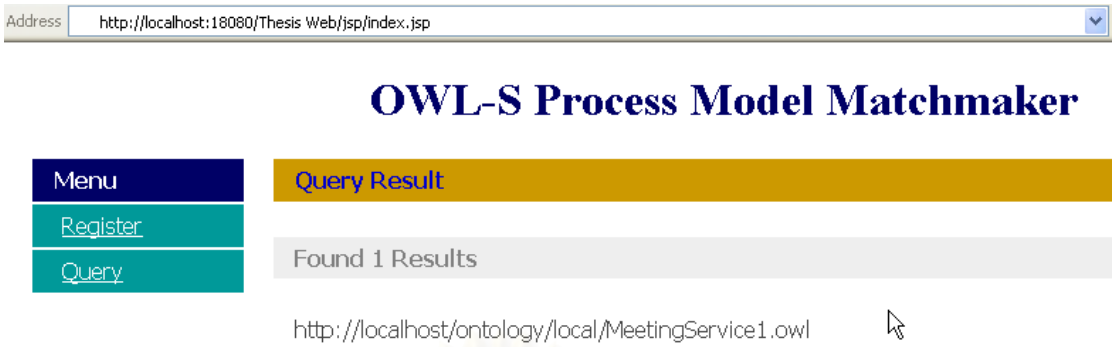


รูปที่ 5.24 การกรอกข้อมูลค้นหาบริการจองโรงแรมของกรณีที่สอง

จากข้อมูลการค้นหาจะสามารถเขียนเซตการค้นหาได้เป็น

$$\mathbb{Q} = \{\text{hasCustomerType(CustomerType, VIP)}^{G},$$
$$\text{hasRepetitiveProcess(Process, MeetingRoomAllocationProcess)}^{P}\}$$

เมื่อพิจารณาเงื่อนไขกระบวนการจะมีบริการที่มีการเข้าคู่กันอย่างถูกต้องเพียงบริการเดียว คือบริการแรก เนื่องจากมีกฎสำหรับกระบวนการภายใน hasCustomerType(CustomerType, VIP) → hasRepetitiveProcess(Process, MeetingRoomAllocationProcess); ซึ่งตรงตามข้อมูลการค้นหา โดยที่กฎสำหรับกระบวนการภายในนี้จะได้มาจาก <process:whileProcess> ในแฟ้มข้อมูลอาวล์-เอสของบริการแรก (รูปที่ ก.3 บรรทัดที่ 54-60) ผลการค้นหาแสดงดังรูปที่ 5.25



รูปที่ 5.25 ผลการค้นหาบริการจองโรงแรมของกรณีที่สอง

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6
สรุปผลการวิจัย

6.1 สรุปผลการวิจัย

ในงานวิจัยนี้ได้นำเสนอแนวทางในการค้นหาเว็บเซอร์วิสด้วยข้อกำหนดกระบวนการ โดยสนใจทั้งพฤติกรรมเชิงหน้าที่และกระแสงานภายใน ซึ่งงานวิจัยนี้อธิบายข้อกำหนดกระบวนการด้วยอาวล์-เอสโพรเซสโมเดลเนื่องจากมีการใช้กันอย่างแพร่หลาย นอกจากนี้ภายในกระแสงานของเว็บเซอร์วิสอาจมีส่วนเงื่อนไขกระบวนการ ซึ่งได้นำมาพิจารณาในการค้นหาด้วย โดยอธิบายเงื่อนไขกระบวนการด้วยภาษากฎสเวิร์ล รูปแบบกระแสงานที่งานวิจัยนี้สนใจได้แก่ กระแสงานที่มีโครงสร้างเป็นแบบการตัดสินใจ แบบลำดับ แบบทำงานพร้อมกัน และแบบทำงานวนซ้ำ

ผู้วิจัยได้พัฒนาตัวกลางเพื่อช่วยในการค้นหาเว็บเซอร์วิส ซึ่งช่วยเพิ่มประสิทธิภาพในการค้นหา โดยได้ใช้กรณีศึกษาสามกรณีศึกษาคือบริการกู้เงิน บริการสั่งซื้อไวน์ และบริการจองโรงแรม เพื่อทดสอบตัวกลาง โดยผลที่ได้นั้นเป็นไปตามความต้องการของผู้ใช้งานที่ต้องการค้นหาบริการ

6.2 ปัญหาและอุปสรรค

1. ถึงแม้ว่าแฟ้มข้อมูลอาวล์-เอสจะเป็นที่แพร่หลายในงานวิจัยเกี่ยวกับเว็บเซอร์ชิส และมีเครื่องมือตัวแจงอาวล์-เอสให้ใช้ แต่การใช้งานยังมีข้อจำกัดคือตัวแจงอาวล์-เอสสามารถอ่านได้เฉพาะคลาสที่กำหนดตามมาตรฐานอาวล์-เอสเท่านั้น แต่ไม่สามารถอ่านคลาสลูกของคลาสมาตรฐานเหล่านั้นได้ ในการพัฒนาตัวกลางจึงต้องสร้างแฟ้มข้อมูลอาวล์-เอสอีกแฟ้มหนึ่งเพื่อทำการประมวลผลความสัมพันธ์ระหว่างคลาสกระบวนการของบริการที่กำหนดขึ้นมาใหม่กับคลาสกระบวนการมาตรฐาน

2. ภาษากฎสเวิร์ลซึ่งเป็นภาษากฎที่แพร่หลายในงานวิจัยยังไม่มีตัวแจงที่ใช้งานได้ยืดหยุ่นเพียงพอ เนื่องจากตัวแจงภาษากฎสเวิร์ลที่พบได้ถูกออกแบบเพื่อการประมวลผลโปรแกรมเชิงตรรกะเท่านั้น ไม่ได้ถูกออกแบบให้ใช้ร่วมกับอาวล์-เอสโพรเซสโมเดล ทำให้ในการพัฒนาต้องทำการสร้างตัวแจงภาษากฎสเวิร์ลขึ้นเอง

6.3 แนวทางการวิจัยต่อไป

ประเด็นที่งานวิจัยนี้ยังไม่ได้ศึกษา และสามารถทำการวิจัยเพิ่มเติมได้ในอนาคตได้แก่

1. งานวิจัยนี้สามารถพัฒนาให้สมบูรณ์ต่อไปได้โดยพิจารณาการจัดลำดับผลลัพธ์จากการค้นหา ซึ่งต้องพิจารณาถึงตัวแปรในหลายแง่มุม เช่น ในด้านพฤติกรรมเชิงหน้าที่ หรือในด้านกระบวนการของบริการ

2. สามารถพัฒนาตัวกลางของงานวิจัยนี้ให้มีประสิทธิภาพยิ่งขึ้นในเรื่องของเวลาในการค้นหาได้ โดยเปลี่ยนแปลงอัลกอริทึมของการค้นหาจากเดิมที่นำออนโทโลจีกระบวนการที่ใช้งานร่วมกันที่ระบุร่วมกับข้อมูลการค้นหามาหาเว็บเซอร์วิสที่อ้างอิงออนโทโลจีกระบวนการที่ใช้งานร่วมกันนี้ทั้งหมด แล้วค่อยวนซ้ำเพื่อประมวลผลข้อกำหนดกระบวนการของเว็บเซอร์วิสเหล่านี้เพื่อหาเว็บเซอร์วิสที่ตรงตามการค้นหาโดยเปลี่ยนเป็น เมื่อผู้ให้บริการประกาศบริการผ่านตัวกลางแล้ว ตัวกลางจะทำการประมวลผลข้อกำหนดกระบวนการเว็บเซอร์วิสนั้นทันทีแล้วเก็บข้อมูลเว็บเซอร์วิสนั้นไว้ในโครงสร้างข้อมูลแบบต้นไม้ก่อน เพื่อที่เวลาค้นหาจะได้ไม่ต้องนำข้อกำหนดกระบวนการของเว็บเซอร์วิสเหล่านี้มาประมวลผล เพียงแต่อ่านค่าจากโครงสร้างข้อมูลต้นไม้ที่ได้เตรียมไว้แล้ว ซึ่งจะทำให้การค้นหานั้นทำได้อย่างรวดเร็วยิ่งขึ้น

3. ตัวกลางยังไม่รองรับข้อกำหนดอาวล์-เอสโพรเซสโมเดลที่ประกอบด้วยกระบวนการประกอบซึ่งมีการอ้างอิงมาจากภายนอก (Import) แต่จะรองรับเฉพาะกระบวนการประกอบที่มีการประกาศกระบวนการย่อยภายในทั้งหมดไว้ในแฟ้มข้อมูลอาวล์-เอสแฟ้มเดียวกัน ดังนั้นจึงสามารถพัฒนาตัวกลางให้สมบูรณ์มากขึ้นได้โดยพัฒนาให้สามารถรองรับกระบวนการประกอบที่มีรายละเอียดกระบวนการภายในอยู่ในแฟ้มข้อมูลอาวล์-เอสที่อ้างอิงมาได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

[1] Cerami, E. Web Services Essentials. First Edition. O'Reilly, 2002.

[2] Huhns, M. N., Singh, M. P. Service-Oriented Computing: Key Concepts and Principles. IEEE Internet Computing. (January-February 2005): 75-81.

[3] uddi.org. UDDI: Universal Description, Discovery, and Integration of Web Services (online). (2002). http://www.uddi.org.

[4] Burstein, M. Semantic Web Services Architecture. IEEE Internet Computing. (September-October 2005): 72-81.

[5] OWL-S Coalition. OWL-S 1.1 Release (online). http://www.daml.org/services/owl-s/1.1

[6] WSMO. Web Services Modeling Ontology (online). (2004). http://www.wsmo.org

[7] Bruijn, D.J., Lausen, H., Polleres, A., Fensel, D. The Web Service Modeling Language WSML: An Overview. DERI Technical Report. (June 2005).

[8] Paolucci, M., Sycara, K. UDDI Spec TC V4 Proposal Semantic Search (online). (2004). http://www.oasis-open.org/committees/uddi-spec/doc/req/uddi-spec-tc-req029-semanticsearch-20040308.doc

[9] Sriharee, N., Senivongse, T. Matchmaking and Ranking of Semantic Web Services Using Integrated Service Profile. International Journal of Metadata, Semantics and Ontologies Vol. 1, No. 2, Inderscience Publishers (2006).

[10] Klein, M., Bernstein, A. Searching for Services on the Semantic Web Using Process Ontologies. Semantic Web Working Symposium. (2002): 159-172.

[11] Sriharee, N., Senivongse, T. Matchmaking for Semantic Web Services with Constraints on Process Model. Proceedings. 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases. (February 2006).

[12] Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M. SWRL: A Semantic Web Rule Language combining OWL and RuleML. (online). (2003). http://daml.org/2003/11/swrl/

[13] Protégé. (online). http://protege.stanford.edu/

[14] Jena Semantic Web Framework: Jena. (online). http://jena.sourceforge.net/index.html

[15] Li, L., Horrocks, I. A Software Framework for Matchmaking Based on Semantic Web Technology. Proceedings of 12th International World Wide Web Conference. (2003).

[16] Bossam Rule/Owl Reasoner. (online). http://mknows.etri.re.kr/bossam/

[17] Havey, M. Essential Business Process Modeling. O'Rielly, 2005.

[18] Barros, O. H. Business Information System Design Based on Process Patterns and Frameworks. (online). (2004). http://www.bptrends.com

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

ตัวอย่างอาวล์-เอสโพรเซสโมเดลที่ใช้ในการทดสอบ

ในภาคผนวกนี้จะแสดงตัวอย่างอาวล์-เอสโพรเซสโมเดลที่ใช้ในการทดสอบ โดยจะแสดงตัวอย่างทั้งสามกรณีศึกษา แต่ละกรณีศึกษาจะแสดงเฉพาะอาวล์-เอสโพรเซสโมเดลของบริการแรกเท่านั้น

```
1.  <?xml version="1.0"?>
2.  <rdf:RDF xmlns:process="http://www.daml.org/services/owl-s/1.1/Process.owl#"
    xmlns:list="http://www.daml.org/services/owl-s/1.1/generic/ObjectList.owl#" xmlns:swrl="http://www.w3.org/2003/11/swrl#"
    xmlns:time="http://www.isi.edu/~pan/damltime/time-entry.owl#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns="http://localhost:/ontology/local/LoanService1.owl#"
    xmlns:expr="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#"
    xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
    xmlns:service="http://www.daml.org/services/owl-s/1.1/Service.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:grounding="http://www.daml.org/services/owl-s/1.1/Grounding.owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:profile="http://www.daml.org/services/owl-s/1.1/Profile.owl#"
    xml:base="http://localhost/ontology/local/LoanService1.owl">
3.    <owl:Ontology rdf:about="">
4.      <owl:imports rdf:resource="http://www.daml.org/rules/proposal/swrl.owl"/>
5.      <owl:imports rdf:resource="http://www.daml.org/rules/proposal/swrlb.owl"/>
6.      <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.1/Profile.owl"/>
7.      <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.1/Grounding.owl"/>
8.    </owl:Ontology>
9.  <process:CompositeProcess rdf:ID="LoanService_S1">
10.     <process:hasInput>
11.       <process:Input rdf:ID="CustomerInfo_S1"/>
12.     </process:hasInput>
13.     <process:hasOutput>
14.       <process:Output rdf:ID="LoanInterestRate_S1"/>
15.     </process:hasOutput>
16.     <process:hasLocal>
17.       <process:Local rdf:ID="IncomePerMonth_S1"/>
18.     </process:hasLocal>
19.     <process:hasLocal>
20.       <process:Local rdf:ID="LoanPurpose_S1"/>
21.     </process:hasLocal>
22.     <process:hasPrecondition>
23.       <expr:Condition rdf:ID="IncomeCondition_S1">
24.         <expr:expressionLanguage rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#SWRL"/>
25.         <expr:expressionBody rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">
26.           <swrl:AtomList>
27.             <rdf:first>
28.               <swrl:BuiltinAtom>
29.                 <swrl:builtin rdf:resource="http://www.w3.org/2003/11/swrlb#greaterThanOrEqual"/>
30.                 <swrl:arguments>
31.                   <rdf:List>
32.                     <rdf:first rdf:resource="#IncomePerMonth_S1"/>
33.                     <rdf:rest>
34.                       <rdf:List>
35.                         <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#int">10000</rdf:first>
36.                         <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
37.                       </rdf:List>
38.                     </rdf:rest>
39.                   </rdf:List>
40.                 </swrl:arguments>
41.               </swrl:BuiltinAtom>
42.             </rdf:first>
43.             <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
44.           </swrl:AtomList>
45.         </expr:expressionBody>
46.       </expr:Condition>
47.     </process:hasPrecondition>
48.     <process:hasResult>
49.       <process:Result rdf:ID="PremiumLoanResult_S1">
50.         <process:hasResultVar>
51.           <process:ResultVar rdf:ID="LoanAmount_S1"/>
```

รูปที่ ก.1 อาวล์-เอสโพรเซสโมเดลของบริการเงินกู้ของธนาคารแรก

```
52.       </process:hasResultVar>
53.       <process:hasResultVar>
54.         <process:ResultVar rdf:ID="PremiumCreditCardFee_S1"/>
55.       </process:hasResultVar>
56.       <process:inCondition>
57.         <expr:Condition rdf:ID="PremiumLoanCondition_S1">
58.           <expr:expressionLanguage rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#SWRL"/>
59.           <expr:expressionBody rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">
60.             <swrl:AtomList>
61.               <rdf:first>
62.                 <swrl:BuiltinAtom>
63.                   <swrl:builtin rdf:resource="http://www.w3.org/2003/11/swrlb#greaterThan"/>
64.                   <swrl:arguments>
65.                     <rdf:List>
66.                       <rdf:first rdf:resource="#LoanAmount_S1"/>
67.                       <rdf:rest>
68.                         <rdf:List>
69.                           <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#int">300000</rdf:first>
70.                           <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
71.                         </rdf:List>
72.                       </rdf:rest>
73.                     </rdf:List>
74.                   </swrl:arguments>
75.                 </swrl:BuiltinAtom>
76.               </rdf:first>
77.               <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
78.             </swrl:AtomList>
79.           </expr:expressionBody>
80.         </expr:Condition>
81.       </process:inCondition>
82.       <process:hasEffect>
83.         <expr: Expression rdf:ID="PremiumCreditCardExpression_S1">
84.           <expr:expressionLanguage rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#SWRL"/>
85.           <expr:expressionBody rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">
86.             <swrl:AtomList>
87.               <rdf:first>
88.                 <swrl:BuiltinAtom>
89.                   <swrl:builtin rdf:resource="http://www.w3.org/2003/11/swrlb#equal"/>
90.                   <swrl:arguments>
91.                     <rdf:List>
92.                       <rdf:first rdf:resource="#PremiumCreditCardFee_S1"/>
93.                       <rdf:rest>
94.                         <rdf:List>
95.                           <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#int">0</rdf:first>
96.                           <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
97.                         </rdf:List>
98.                       </rdf:rest>
99.                     </rdf:List>
100.                  </swrl:arguments>
101.                </swrl:BuiltinAtom>
102.              </rdf:first>
103.              <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
104.            </swrl:AtomList>
105.          </expr:expressionBody>
106.        </expr:Expression>
107.      </process:hasEffect>
108.    </process:Result>
109.  </process:hasResult>
110.  <process:composedOf>
111.    <process:If-Then-Else rdf:ID="LoanAmount_If-Then-Else_S1">
112.      <process:ifCondition>
113.        <expr:Condition rdf:ID="LoanAmountCondition_S1">
114.          <expr:expressionLanguage rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#SWRL"/>
115.          <expr:expressionBody rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">
116.            <swrl:AtomList>
117.              <rdf:first>
118.                <swrl:BuiltinAtom>
119.                  <swrl:builtin rdf:resource="http://www.w3.org/2003/11/swrlb#lessThanOrEqual"/>
120.                  <swrl:arguments>
121.                    <rdf:List>
122.                      <rdf:first rdf:resource="#LoanAmount_S1"/>
123.                      <rdf:rest>
124.                        <rdf:List>
125.                          <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1000000</rdf:first>
126.                          <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
127.                        </rdf:List>
128.                      </rdf:rest>
129.                    </rdf:List>
130.                  </swrl:arguments>
131.                </swrl:BuiltinAtom>
132.              </rdf:first>
```

รูปที่ ก.1 อาวล์-เอสโพรเซสโมเดลของบริการเงินกู้ของธนาคารแรก (ต่อ)

```
133.            <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
134.          </swrl:AtomList>
135.        </expr:expressionBody>
136.      </expr:Condition>
137.    </process:ifCondition>
138.    <process:then>
139.      <process:Sequence rdf:ID="Bank_Sequence_S1">
140.        <process:components>
141.          <process:ControlConstructList rdf:ID="LoanHistory_ControlConstructList_S1">
142.            <list:first>
143.              <process:Perform rdf:ID="LoanHistoryPerform_S1">
144.                <process:process>
145.                  <process:AtomicProcess rdf:ID="LoanHistoryProcess_S1"/>
146.                </process:process>
147.              </process:Perform>
148.            </list:first>
149.            <list:rest>
150.              <process:ControlConstructList rdf:ID="Bank_ControlConstructList_S1">
151.                <list:first>
152.                  <process:If-Then-Else rdf:ID="Purpose_If-Then-Else_S1">
153.                    <process:ifCondition>
154.                      <expr:Condition rdf:ID="PurposeCondition_S1">
155.                        <expr:expressionLanguage
                              rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#SWRL"/>
156.                        <expr:expressionBody rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">
157.                          <swrl:AtomList>
158.                            <rdf:first>
159.                              <swrl:BuiltinAtom>
160.                                <swrl:builtin rdf:resource="http://www.w3.org/2003/11/swrlb#equal"/>
161.                                <swrl:arguments>
162.                                  <rdf:List>
163.                                    <rdf:first rdf:resource="#LoanPurpose_S1"/>
164.                                    <rdf:rest>
165.                                      <rdf:List>
166.                                        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
                                            http://localhost/ontology/domain/RealEstate.owl#RealEstate</rdf:first>
167.                                        <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
168.                                      </rdf:List>
169.                                    </rdf:rest>
170.                                  </rdf:List>
171.                                </swrl:arguments>
172.                              </swrl:BuiltinAtom>
173.                            </rdf:first>
174.                            <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
175.                          </swrl:AtomList>
176.                        </expr:expressionBody>
177.                      </expr:Condition>
178.                    </process:ifCondition>
179.                    <process:then>
180.                      <process:Perform rdf:ID="DepartmentApprovalPerform_S1">
181.                        <process:process>
182.                          <process:AtomicProcess rdf:ID="DepartmentApprovalProcess_S1"/>
183.                        </process:process>
184.                      </process:Perform>
185.                    </process:then>
186.                    <process:else>
187.                      <process:Perform rdf:ID="BranchApprovalPerform_S1">
188.                        <process:process>
189.                          <process:AtomicProcess rdf:ID="BranchApprovalProcess_S1"/>
190.                        </process:process>
191.                      </process:Perform>
192.                    </process:else>
193.                  </process:If-Then-Else>
194.                </list:first>
195.                <list:rest rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/ObjectList.owl#nil"/>
196.              </process:ControlConstructList>
197.            </list:rest>
198.          </process:ControlConstructList>
199.        </process:components>
200.      </process:Sequence>
201.    </process:then>
202.    <process:else>
203.      <process:Perform rdf:ID="HeadquarterApprovalPerform_S1">
204.        <process:process>
205.          <process:AtomicProcess rdf:ID="HeadquarterApprovalProcess_S1"/>
206.        </process:process>
207.      </process:Perform>
208.    </process:else>
209.  </process:If-Then-Else>
210. </process:composedOf>
211. </process:CompositeProcess>
212.</rdf:RDF>
```

รูปที่ ก.1 อาวล์-เอสโพรเซสโมเดลของบริการเงินกู้ของธนาคารแรก (ต่อ)

```
1.  <?xml version="1.0"?>
2.  <rdf:RDF xmlns:process="http://www.daml.org/services/owl-s/1.1/Process.owl#"
      xmlns:list="http://www.daml.org/services/owl-s/1.1/generic/ObjectList.owl#"
      xmlns:swrl="http://www.w3.org/2003/11/swrl#"
      xmlns:time="http://www.isi.edu/~pan/damltime/time-entry.owl#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:owl="http://www.w3.org/2002/07/owl#"
      xmlns:expr="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#"
      xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
      xmlns="http://localhost/ontology/local/orderprocessingservice.owl#"
      xmlns:service="http://www.daml.org/services/owl-s/1.1/Service.owl#"
      xmlns:grounding="http://www.daml.org/services/owl-s/1.1/Grounding.owl#"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema#"/
      xmlns:profile="http://www.daml.org/services/owl-s/1.1/Profile.owl#"
      xml:base="http://localhost/ontology/local/OrderService1.owl">
3.    <owl:Ontology rdf:about="">
4.      <owl:imports rdf:resource="http://www.daml.org/rules/proposal/swrl.owl"/>
5.      <owl:imports rdf:resource="http://www.daml.org/rules/proposal/swrlb.owl"/>
6.      <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.1/Profile.owl"/>
7.      <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.1/Grounding.owl"/>
8.    </owl:Ontology>
9.    <rdf:Description rdf:about="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#AlwaysTrue">
10.     <expr:expressionBody rdf:parseType="Literal">
11.       <swrl:AtomList xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
12.         xmlns:swrl="http://www.w3.org/2003/11/swrl#" rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil">
13.       </swrl:AtomList>
14.     </expr:expressionBody>
15.     <expr:expressionLanguage rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#SWRL"/>
16.   </rdf:Description>
17.   <process:hasLocal>
18.     <process:Local rdf:ID="Age_S1"/>
19.   </process:hasLocal>
20.   <process:hasPrecondition>
21.     <expr:Condition rdf:ID="AgeCondition_S1">
22.       <expr:expressionLanguage rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#SWRL"/>
23.       <expr:expressionBody rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">
24.         <swrl:AtomList>
25.           <rdf:first>
26.             <swrl:BuiltinAtom>
27.               <swrl:builtin rdf:resource="http://www.w3.org/2003/11/swrlb#greaterThanOrEqual"/>
28.               <swrl:arguments>
29.                 <rdf:List>
30.                   <rdf:first rdf:resource="#Age_S1"/>
31.                   <rdf:rest>
32.                     <rdf:List>
33.                       <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#int">18</rdf:first>
34.                       <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
35.                     </rdf:List>
36.                   </rdf:rest>
37.                 </rdf:List>
38.               </swrl:arguments>
39.             </swrl:BuiltinAtom>
40.           </rdf:first>
41.           <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
42.         </swrl:AtomList>
43.       </expr:expressionBody>
44.     </expr:Condition>
45.   </process:hasPrecondition>
46.   <process:CompositeProcess rdf:ID="OrderProcessingService">
47.     <process:composedOf>
48.       <process:Sequence rdf:ID="Sequence_4">
49.         <process:components>
50.           <process:ControlConstructList rdf:ID="ControlConstructList_6">
51.             <list:first>
52.               <process:Perform rdf:ID="OrderPlacingPerform">
53.                 <process:process>
54.                   <process:AtomicProcess rdf:ID="OrderPlacingProcess_S1"/>
55.                 </process:process>
56.               </process:Perform>
57.             </list:first>
58.             <list:rest>
59.               <process:ControlConstructList rdf:ID="ControlConstructList_9">
60.                 <list:first>
61.                   <process:Split-Join rdf:ID="Split-Join_8">
62.                     <process:components>
63.                       <process:ControlConstructBag rdf:ID="ControlConstructBag_11">
64.                         <list:rest>
65.                           <process:ControlConstructBag rdf:ID="ControlConstructBag_13">
66.                             <list:rest rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/ObjectList.owl#nil"/>
67.                             <list:first>
```

รูปที่ ก.2 อาวล์-เอสโพรเซสโมเดลของบริการสั่งซื้อไวน์บริการแรก

```
                        <process:Perform rdf:ID="InventoryCheckingPerform">
                         <process:process>
                          <process:AtomicProcess rdf:ID="InventoryCheckingProcess_S1"/>
                         </process:process>
                        </process:Perform>
                       </list:first>
                      </process:ControlConstructBag>
                     </list:rest>
                     <list:first>
                      <process:Perform rdf:ID="CreditCheckingPerform">
                       <process:process>
                        <process:AtomicProcess rdf:ID="CreditCheckingProcess_S1"/>
                       </process:process>
                      </process:Perform>
                     </list:first>
                    </process:ControlConstructBag>
                   </process:components>
                  </process:Split-Join>
                 </list:first>
                 <list:rest>
                  <process:ControlConstructList rdf:ID="ControlConstructList_17">
                   <list:first>
                    <process:Perform rdf:ID="OrderStatusNotificationPerform">
                     <process:process>
                      <process:AtomicProcess rdf:ID="OrderStatusNotificationProcess_S1"/>
                     </process:process>
                    </process:Perform>
                   </list:first>
                   <list:rest rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/ObjectList.owl#nil"/>
                  </process:ControlConstructList>
                 </list:rest>
                </process:ControlConstructList>
               </list:rest>
              </process:ControlConstructList>
             </process:components>
            </process:Sequence>
           </process:composedOf>
          </process:CompositeProcess>
         </rdf:RDF>
```

รูปที่ ก.2 อาวล์-เอสโพรเซสโมเดลของบริการสั่งซื้อไวน์บริการแรก (ต่อ)

```
1.  <?xml version="1.0"?>
2.  <rdf:RDF
      xmlns:process="http://www.daml.org/services/owl-s/1.1/Process.owl#"
      xmlns:list="http://www.daml.org/services/owl-s/1.1/generic/ObjectList.owl#"
      xmlns:swrl="http://www.w3.org/2003/11/swrl#"
      xmlns:time="http://www.isi.edu/~pan/damltime/time-entry.owl#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:owl="http://www.w3.org/2002/07/owl#"
      xmlns:expr="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#"
      xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
      xmlns:service="http://www.daml.org/services/owl-s/1.1/Service.owl#"
      xmlns:grounding="http://www.daml.org/services/owl-s/1.1/Grounding.owl#"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
      xmlns:profile="http://www.daml.org/services/owl-s/1.1/Profile.owl#"
      xmlns="http://localhost/ontology/local/hotelservice.owl#"
      xml:base="http://localhost/ontology/local/HotelService1.owl">
3.    <owl:Ontology rdf:about="">
4.     <owl:imports rdf:resource="http://www.daml.org/rules/proposal/swrl.owl"/>
5.     <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.1/Profile.owl"/>
6.     <owl:imports rdf:resource="http://www.daml.org/rules/proposal/swrlb.owl"/>
7.     <owl:imports rdf:resource="http://www.daml.org/services/owl-s/1.1/Grounding.owl"/>
8.    </owl:Ontology>
9.    <rdf:Description rdf:about="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#AlwaysTrue">
10.     <expr:expressionBody rdf:parseType="Literal">
11.       <swrl:AtomList xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
12.          xmlns:swrl="http://www.w3.org/2003/11/swrl#"
13.          rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"></swrl:AtomList>
14.     </expr:expressionBody>
15.     <expr:expressionLanguage rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#SWRL"/>
16.   </rdf:Description>
17.    <process:hasLocal>
18.     <process:Local rdf:ID="CustomerType_S1"/>
19.    </process:hasLocal>
20.   <process:CompositeProcess rdf:ID="HotelService">
21.     <process:composedOf>
22.      <process:Sequence rdf:ID="Sequence_2">
```

รูปที่ ก.3 อาวล์-เอสโพรเซสโมเดลของบริการจองโรงแรมบริการแรก

```
23.         <process:components>
24.           <process:ControlConstructList rdf:ID="ControlConstructList_4">
25.             <list:first>
26.               <process:Repeat-While rdf:ID="Repeat-While_3">
27.                 <process:whileCondition>
28.                   <expr:Condition rdf:ID="CustomerTypeCondition">
29.                     <expr:expressionLanguage
30.                       rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/Expression.owl#SWRL"/>
31.                     <expr:expressionBody rdf:datatype="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">
32.                       <swrl:AtomList>
33.                         <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
34.                         <rdf:first>
35.                           <swrl:BuiltinAtom>
36.                             <swrl:builtin rdf:resource="http://www.w3.org/2003/11/swrlb#equal"/>
37.                             <swrl:arguments>
38.                               <rdf:List>
39.                                 <rdf:first rdf:resource="#CustomerType_S1"/>
40.                                 <rdf:rest>
41.                                   <rdf:List>
42.                                     <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#String">VIP</rdf:first>
43.                                     <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
44.                                   </rdf:List>
45.                                 </rdf:rest>
46.                               </rdf:List>
47.                             </swrl:arguments>
48.                           </swrl:BuiltinAtom>
49.                         </rdf:first>
50.                       </swrl:AtomList>
51.                     </expr:expressionBody>
52.                   </expr:Condition>
53.                 </process:whileCondition>
54.                 <process:whileProcess>
55.                   <process:Perform rdf:ID="MeetingRoomAllocationPerform">
56.                     <process:process>
57.                       <process:AtomicProcess rdf:ID="MeetingRoomAllocationProcess_S1"/>
58.                     </process:process>
59.                   </process:Perform>
60.                 </process:whileProcess>
61.               </process:Repeat-While>
62.             </list:first>
63.             <list:rest>
64.               <process:ControlConstructList rdf:ID="ControlConstructList_9">
65.                 <list:first>
66.                   <process:Perform rdf:ID="HallAllocationPerform">
67.                     <process:process>
68.                       <process:AtomicProcess rdf:ID="HallAllocationProcess_S1"/>
69.                     </process:process>
70.                   </process:Perform>
71.                 </list:first>
72.                 <list:rest>
73.                   <process:ControlConstructList rdf:ID="ControlConstructList_12">
74.                     <list:first>
75.                       <process:Perform rdf:ID="BookingRecordPerform">
76.                         <process:process>
77.                           <process:AtomicProcess rdf:ID="BookingRecordProcess_S1"/>
78.                         </process:process>
79.                       </process:Perform>
80.                     </list:first>
81.                     <list:rest rdf:resource="http://www.daml.org/services/owl-s/1.1/generic/ObjectList.owl#nil"/>
82.                   </process:ControlConstructList>
83.                 </list:rest>
84.               </process:ControlConstructList>
85.             </list:rest>
86.           </process:ControlConstructList>
87.         </process:components>
88.       </process:Sequence>
89.     </process:composedOf>
90.   </process:CompositeProcess>
91. </rdf:RDF>
```

รูปที่ ก.3 อาวล์-เอสโพรเซสโมเดลของบริการจองโรงแรมบริการแรก (ต่อ)

ภาคผนวก ข

ผลงานตีพิมพ์

ผลงานตีพิมพ์ซึ่งเป็นส่วนหนึ่งของงานวิจัยมีดังนี้

1. Proceedings of 6th International Conference on Distributed Applications and Interoperable Systems, LNCS 4025, Bologna, Italy (2006): 113-127 ในบทความเรื่อง Discovering Semantic Web Services with Process Specifications โดยผู้แต่งคือ Piya Suwannopas และ Twittie Senivongse

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Discovering Semantic Web Services with Process Specifications

Piya Suwannopas and Twittie Senivongse

Department of Computer Engineering, Chulalongkorn University
Phyathai Road, Pathumwan, Bangkok 10330 Thailand
piya.su@student.chula.ac.th, twittie.s@chula.ac.th

Abstract. Service discovery is one of the crucial issues for service-oriented architectural model. Recently the trend is towards semantic discovery by which semantic descriptions are the basis for service matchmaking instead of simple search based on service attributes. OWL-S is a widely adopted semantic specification for Web Services which comprises three profiles. Among those, process model is the profile that describes dynamic behaviour of Web Services in terms of functional aspects and process flows, and is generally aimed for service enactment, composition, and monitoring. This paper presents a new approach to use OWL-S process model for service discovery purpose. A Web Service can have its internal process described as an OWL-S process model specification, and a service consumer can query for a Web Service with a particular process detail. Matchmaking will be based on flexible ontological matching and evaluation of constraints on the functional behaviour and process flow of the Web Service. The architecture for process-based discovery is also presented.

1 Introduction

Service discovery is an important part of service-oriented computing in which services, as building blocks for building applications, are provided and distributed in large-scale open environment [1]. Provided services will publish generalised descriptions of their capability to a matchmaker whereas service consumers consult the matchmaker to identify potential services that most closely satisfy their needs. The effectiveness of service discovery relies on the richness of service metadata and the matchmaking mechanism that utilises the expressiveness of the metadata. Current Web Services Standards realise this concept and provide UDDI [2] as a standard registry that performs matchmaking based on matching of syntactic service attribute values.

From our previous study [3], a service description model has been defined as a result of an empirical survey about service advertisements on the Internet (Fig. 1). The model shows that service advertisements should reflect different aspects of service capabilities; some are simple characteristics and may be in the form of simple attributes whereas some are more complex capabilities and require some specification languages to express them. (Those highlighted in Fig. 1 have no correspondences in

UDDI.) This model is generic, meaning that it is independent from any specific representation languages and can be used simply for information or for other purposes such as automatic service discovery or composition.

One way to enrich service metadata is by using ontology languages to represent service descriptions. This approach is gaining a lot of attention in Web Services community as ontology languages are expressive for describing several aspects of service capabilities and ontological reasoning also provides a way to infer more about the capabilities. Semantic Web Services are Web Services in which ontologies ascribe meanings to published service descriptions so that software systems representing prospective service consumers can interpret and invoke them [4]. With this vision, the Web Ontology Language for Services (OWL-S) consortium contributes with an OWL-S specification [5] which is the building block for encoding rich semantic service descriptions in a way that builds naturally upon OWL language. OWL-S consists of three profiles, namely service profile, process model, and service grounding. Service profile defines basic and functional properties of the service as well as functional behaviour. Process model details service operation in terms of functional behaviour, control structure, and data flow structure required to execute the service. Service grounding specifies details of how to access the service by mapping from an abstract service specification (process model) to concrete specification (WSDL). It can be seen that OWL-S and the model in Fig. 1 share some characteristic; they both model services with simple attributes and more complex specifications.

Our previous work [6] proposes an integrated service profile that corresponds to the model in Fig. 1. The integrated service profile is a collection of ontology-based profiles for services, including the attribute, structural, behavioural, and rule profiles,

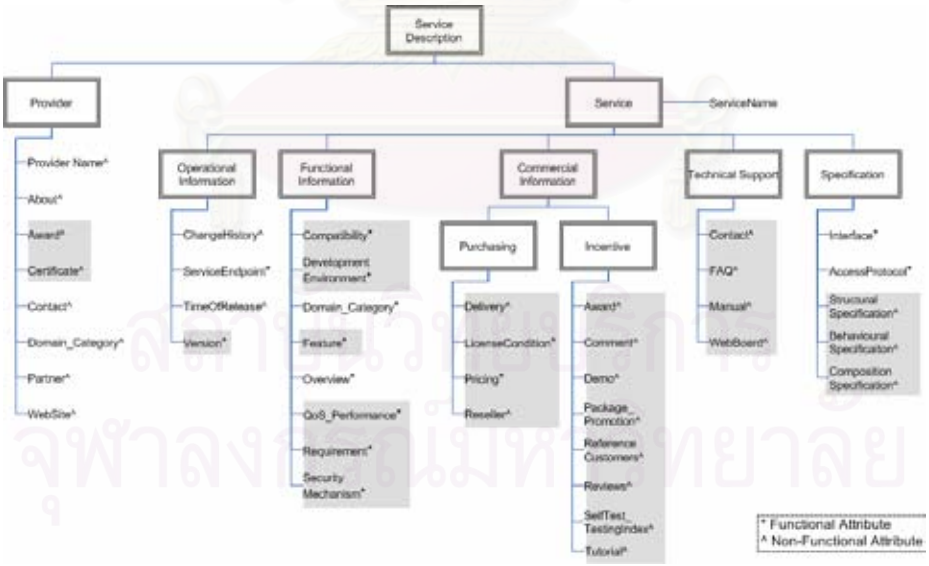


Fig. 1. Service description model from survey [3].

and it overlaps with OWL-S. This paper extends the integrated service profile with the focus on the composition specification of a service. Composition specification shows how simple components are composed into a service and may be expressed as a hierarchy of goal and subgoals or as a workflow of tasks for service execution [1]. This paper is interested in describing the composition specification as a workflow and we borrow OWL-S process model to represent the workflow specification.

OWL-S process model is found in use by researches in service composition and workflow coordination and monitoring, but it can also be used for in-depth analysis for matchmaking to see whether the service meets process constraints required by the service consumer. This is to check a dynamic aspect of the service. For example, the service consumer may want to find a software store with a workflow such that, after processing the purchase order of the customer, the store registers the customer for the software training programme. The store service with such automatic registration for training should be preferable to ones without training. Sometimes the flow may have a constraint such that automatic training registration is available only if the purchase is worth more than 0.5 million bahts (Thailand currency). Such a constraint will have to be taken into account during matchmaking. Here we present an example of the services using OWL-S process model to describe their internal processes. A service consumer can issue a process-based query. The services are queried on their functional behaviour and flow of their process. Ontological reasoning and evaluation of the rule-based constraints on the behaviour and process flow are considered.

The rest of the paper starts with Section 2 that discusses related work. Section 3 outlines the constructs of OWL-S process model for process specification. Section 4 gives an example of the process specifications of three services described using OWL-S process model. Matching criteria are summarised in Section 5 and used in Section 6 to consider matching for a query. Section 7 presents a process-based discovery framework and Section 8 concludes the paper.

2 Related Work

Semantics-based service discovery is accomplished mainly by the use of ontology to describe service capabilities. Web Services Modeling Ontology (WSMO) [7] provides a framework for describing semantic Web Services with Web Services Modeling Language (SWML) [8] as a formal language that realises the framework. WSML defines semantics in terms of four elements: ontologies, goals, Web Service descriptions, and mediators. Ontologies provide vocabularies, concepts, instances, and axioms that will be used by other elements. Goals are similar to queries. Web Service descriptions describe capability in terms of assumption, precondition, postcondition, effect, and allow for interface and orchestration specifications. As WSMO shares with OWL-S the vision that ontologies are essential to support automatic discovery, it is possible for our work to adopt either of their process-related specifications. However, at the moment OWL-S can be implemented without stipulating framework and several tools exist. We adopt OWL-S process model for process specification in this paper.

Most of research work in service discovery area focuses on search based on a particular aspect of the service and little is found to concentrate on process-based discovery. UDDI version 4 is incorporating an ontology-based taxonomy for the standard categories of Business Entity and Business Service entries that are registered with UDDI [9]. This will allow UDDI to be able to look for the businesses or services of a specialised or generalised category. The work in [10] shows how ontology describing general knowledge of a particular service domain can be used for search. The work in [11], [12] focuses on searching functional behaviour but they do not consider search with behavioural constraints. In [13], an efficient search algorithm is devised for services described by OWL-S but the search considers only the OWL-S service profile. In [14], process ontology is used as a basis for service discovery. The process ontology is described by the service process, constituent subtasks, connection ports between subtasks and connection mechanisms, and exceptions within the process. The query is done by a PQL language. Unlike our approach, the process ontology in this work follows the goal-subgoal model of service composition, not the workflow model, and it does not accommodate for process constraints.

Service discovery and service composition share a characteristic such that both aim to identify services that can satisfy users' requirements. Nevertheless, service discovery tends to identify individual services that can answer to a particular query, whereas service composition identifies a group of services that can work together to satisfy a certain goal. In the area of Web Service composition, OWL-S process model is used in several researches for describing Web Services. In [15], an AI planner called OWLS-Xplan is proposed to compose Web Services. An OWL-S process model is used to specify input, precondition, output, and effect of the goal (i.e. the composite service) and of the individual Web Services to be composed. The goal in OWL-S process model will be translated into a planning domain description in PDDL in order for the planner to generate a plan sequence as a workflow of individual Web Services. The work in [16] integrates an OWL reasoner with an AI planner and shows how OWL or SWRL [17] is used to encode the preconditions and effects of the Web Services in the composition process. The Web Services are also described by OWL-S process model. By using OWL, the composition gains the reasoning power of OWL in the evaluation of the preconditions and the update of the effects that have impacts on real world knowledge. Although these researches above conduct some analysis on OWL-S process model, they concern the functional behaviour part of the process model in service composition. In our work, we focus on analysing not only the functional behaviour part but also the workflow part of individual Web Services in order to find any single services that can satisfy the query.

3 OWL-S Process Model

This section briefly describes the constructs of OWL-S process model that are of interest to this paper. A particular service is described by a service model and a process is a subclass of the service model. Fig. 2 shows OWL-S process ontology [5] with the classes and properties that altogether describe how a service works. A process

describes its functional behaviour by specifying inputs, outputs, preconditions, and effects (IOPE) of its performance. As the name implies, a precondition is a logical expression which must hold for the process to be successfully invoked. Local refers to an auxiliary parameter that is bound to the precondition and is useful for determining the logical value of the precondition. Result refers to a coupled output and effect and can be constrained by an incondition property which specifies the logical condition under which the result occurs; hence the corresponding output and effect become conditional output and conditional effect. Result variable is also an auxiliary parameter that is bound to a result and useful for determining the associated incondition.

The process is further described as a composition of subprocesses. The subprocess can be atomic, composite, or simple process. An atomic process is one which has no further subprocesses, is directly invocable, and executes in a single step. A composite process is decomposed into other non-composite or composite processes. The decomposition can be specified by using control constructs, i.e. sequence, split, split-join, any-order, choice, if-then-else, iterate, repeat-while, repeat-until. A simple process is an abstraction that provides a view of some atomic process or a simplified representation of some composite process and is not invocable.

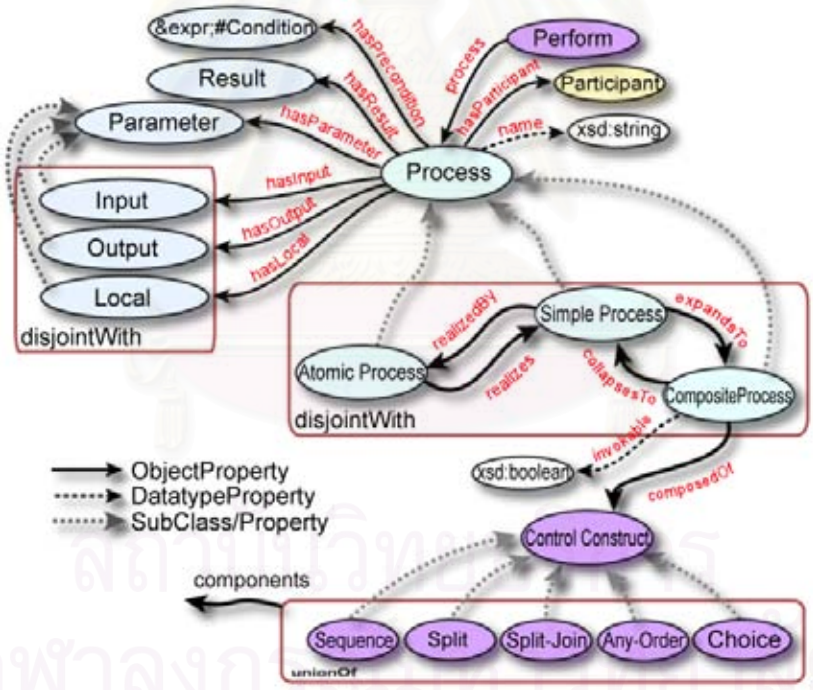


Fig. 2. Top level of process ontology [5]

Since the constraints in OWL-S process model – either the preconditions, the conditions of the results, or the guards on the control flow – are represented as logical

formula, these logical expressions are treated as literals – either XML literals or string literals. Therefore several languages can be used to express these constraints (e.g. SWRL, RDF, KIF, PDDL). In this paper we represent such process constraints with SWRL rule expressions.

4 Process Specifications

Bank loan service is used as an example for process-based discovery. Fig. 3 shows the first part of the process specification of a loan service S_1 written in OWL-S process model. This part describes the functional behaviour of S_1.

```
1.  <process:CompositeProcess rdf:ID="LoanService">
2.    <process:hasInput>
3.      <process:Input rdf:ID="CustomerInfo"/>
4.    </process:hasInput>
5.    <process:hasOutput>
6.      <process:Output rdf:ID="LoanInterestRate"/>
7.    <process:hasOutput/> …
8.    <process:hasLocal rdf:resource="#IncomePerMonth"/>
9.    <process:hasPrecondition>
10.     <expr:SWRL-Condition rdf:ID="IncomeCondition">
11.       <expr:expressionLanguage rdf:resource="&Expression.owl#SWRL"/>
12.       <expr:expressionBody rdf:datatype="Literal">
13.         swrlb:greaterThanOrEqual(#IncomePerMonth,10000) →
14.         hasIncomeStatus(#ValidIncome,"xsd:True")
15.       </expr:expressionBody>
16.     </expr:SWRL-Condition>
17.   </process:hasPrecondition>
18.   <process:hasResultVar rdf:resource="#LoanAmount"/>
19.   <process:hasResult>
20.     <process:Result rdf:ID="PremiumLoanResult">
21.       <process:inCondition>
22.         <expr:SWRL-Condition rdf:ID="PremiumLoanCondition">
23.           <expr:expressionLanguage rdf:resource="&Expression.owl#SWRL"/>
24.           <expr:expressionBody rdf:datatype="Literal">
25.             swrlb:greaterThan(#LoanAmount,300000) →
26.             hasPremiumLoanStatus(#PremiumLoanStatus,"xsd:True")
27.           </expr:expressionBody>
28.         </expr:SWRL-Condition>
29.       </process:inCondition>
30.       <process:hasEffect>
31.         <expr:SWRL-Condition rdf:ID=”PremiumCreditCardCondition”>
32.           <expr:expressionBody rdf:datatype="Literal">
33.             → chargedPremiumCreditCard(#LoanService, #PremiumCreditCardFee)
34.               swrlb:equal(#PremiumCreditCardFee, 0)
35.           </expr:expressionBody>
36.         </expr:SWRL-Condition>
37.       </process:hasEffect>
38.     </process:Result>
39.   </process:hasResult>
40.   <process:hasResult>
41.   <process:Result rdf:ID="NormalLoanResult">
42.       …
```

Fig. 3. Functional behaviour of S_1 in OWL-S process model

From the figure, the service requires customer information as an input (line 2-4), and gives loan interest rate as an output (line 5-7). The service has a precondition such that the consumer needs to have income at least 10,000 bahts per month in order to use the service (line 9-17). The effects of this service are conditional, depending on the loan amount. If the loan is more than 300,000 bahts, it is a premium loan (line 21-29) and the consumer is entitled to apply for a premium credit card. This effect is further constrained by the annual credit card fee which is equal to 0 (line 30-37). On the other hand, if the loan is not more than 300,000 bahts, it is a normal loan (line 41) and the credit card effect will be subject to the annual fee. Note that all the constraints are expressed as SWRL rules.

The second part of the process specification of S_1 involves its workflow. This is depicted in Fig. 4. Suppose, in general, a loan service is composed of several classes of loan approval. Department approval process is performed when the loan amount is small or the loan is not critical and the decision can be made by the loan department manager. Branch approval process is performed when the loan is more critical but the decision can still be made within the branch by the branch manager. Otherwise the loan application has to be approved at the head quarter. The bank will maintain loan history of the customers for future reference.

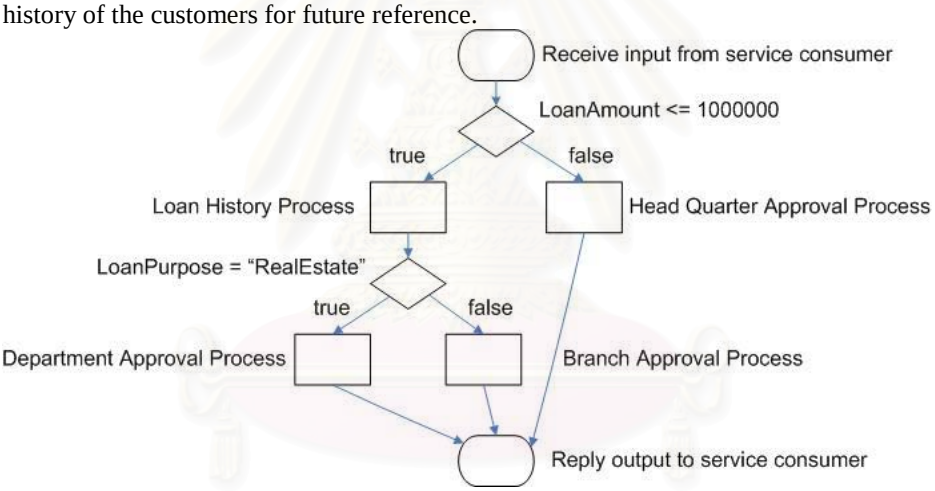


Fig. 4. Process flow of service S_1

Fig. 5 shows a snippet of OWL-S process specification for Fig. 4. The first guard condition checks whether the loan amount is less than or equal to 1 million bahts (line 64-72). The second guard condition determines whether the purpose of loan is for real estate (line 88-96).

For further comparison, we assume there are two more candidate services S_2 and S_3. These two services exhibit the same functional behaviour as S_1 (c.f. Fig. 3) but they have a slightly different workflow as in Fig. 6 and Fig. 7 respectively.

```
<process:composedOf>
 <process:If-Then-Else rdf:ID="LoanAmount_If-Then-Else">
  <process:ifCondition>
   <expr:Condition rdf:ID="LoanAmountCondition">
    <expr:expressionLanguage rdf:resource="&Expression.owl#SWRL"/>
    <expr:expressionBody rdf:datatype="Literal">
     swrlb:lessThanOrEqual(#LoanAmount,1000000) →
     hasLoanAmountStatus(#SmallLoanAmount,"xsd:True")
    </expr:expressionBody>
   </expr:Condition>
  </process:ifCondition>
  <process:then>
   <process:Sequence rdf:ID="Bank_Sequence">
    <process:components>
     <process:ControlConstructList rdf:ID="LoanHistory_ControlConstructList">
      <list:first>
       <process:Perform rdf:ID="LoanHistoryPerform">
        <process:process>
         <process:AtomicProcess rdf:ID="LoanHistoryProcess"/>
        </process:process>
       </process:Perform>
      </list:first>
      <list:rest>
       <process:ControlConstructList rdf:ID="Bank_ControlConstructList">
        <list:first>
         <process:If-Then-Else rdf:ID="Purpose_If-Then-Else">
          <process:ifCondition>
           <expr:Condition rdf:ID="PurposeCondition">
            <expr:expressionLanguage rdf:resource="&Expression.owl#SWRL"/>
            <expr:expressionBody rdf:datatype="Literal">
             swrlb:equal(#LoanPurpose,"RealEstate") →
             hasPurposeStatus(#RealEstatePurpose,"xsd:True")
            </expr:expressionBody>
           </expr:Condition>
          </process:ifCondition>
          <process:then>
           <process:Sequence rdf:ID="Department_Sequence">
            <process:components>
             <process:ControlConstructList rdf:ID="Department_ControlConstructList">
              <list:first>
               <process:Perform rdf:ID="DepartmentApprovalPerform">
                <process:process>
                 <process:AtomicProcess rdf:ID="DepartmentApprovalProcess"/>
                </process:process>
               </process:Perform>
                    …
```

Fig. 5. Process flow of S_1 in OWL-S process model

5 Matching Criteria

To determine whether a process specification of a service can fulfill a service consumer's needs, matchmaking will perform ontological matching on the concepts within the specification and evaluate constraints on the functional behaviour and the guards on the control constructs in order to determine the actual behaviour of the service. Several matching criteria are defined:

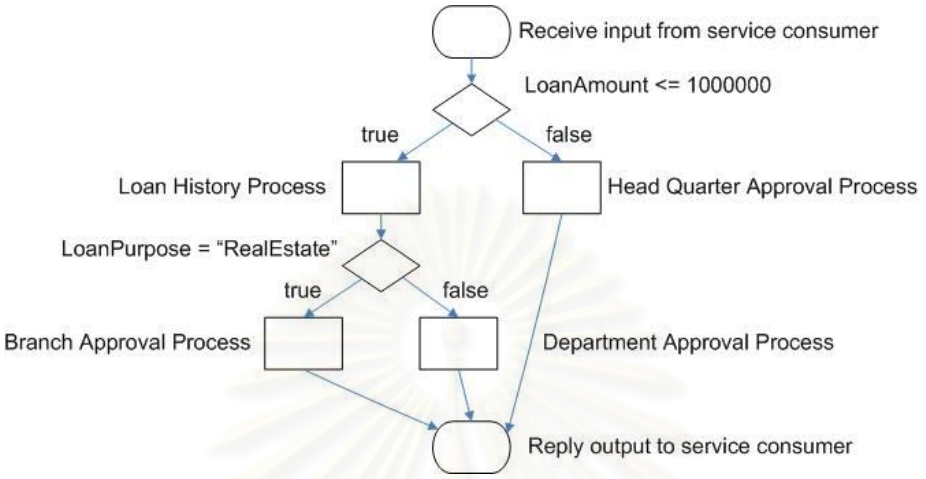


Fig. 6. Process flow of service S_2

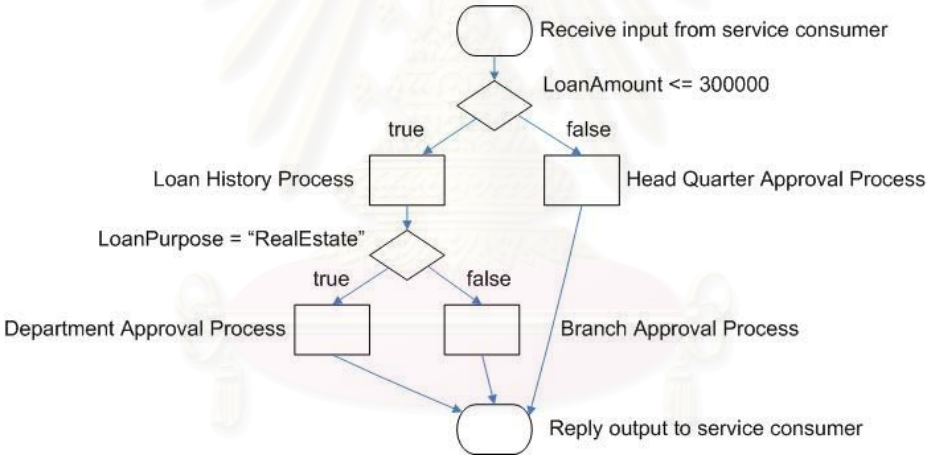


Fig. 7. Process flow of service S_3

5.1 Matching Ontological Concepts

Matching by subsumption and equivalence is the basis for matching ontological concepts in the query and the process specification. This approach is based on the IS-A taxonomy of the concepts shared within the service domain and has been adopted in literature including [10], [18], [6].

Let C_Q be the concept specified in the query and C_P be the concept in the process specification:

(i) *If $C_Q \equiv C_P$ then C_P is an exact match for C_Q,* where $\equiv$ means is equivalent to.

(ii) *If $C_P \sqsubseteq C_Q$ then C_P is a specialised match for C_Q,* where $\sqsubseteq$ means is subsumed by (i.e. C_P is more specific than C_Q).

(iii) *If $C_Q \sqsubseteq C_P$ then C_P is a generalised match for C_Q.* This means the concept in the query is more specific than, and is subsumed by, the one in the process specification.

(iv) *If $(C_Q \not\sqsubseteq C_P) \wedge (C_P \not\sqsubseteq C_Q) \wedge (C_Q \sqsubseteq C_C) \wedge (C_P \sqsubseteq C_C)$ then C_P is a partial match for C_Q,* where $\not\sqsubseteq$ means is not subsumed by and C_C is a node in the same IS-A taxonomy. This means it is acceptable for the concept in the process specification to be a match for the concept in the query provided that the two concepts have common characteristics through a common parent concept.

(v) *If none of the above relationships exist then C_P is a failed match for C_Q.*

5.2 Matching Numerical Ranges

Matching two numerical ranges compares the ranges of the possible values that are defined in the constraints. The degree of matching for numerical ranges can be determined as described below.

Let N_Q be a nonempty set of numerical range values of the expression in the query (E_Q), and N_P be a nonempty set of numerical range values of the expression in the process specification (E_P):

(i) *If $N_P \subseteq N_Q$ then E_P is an exact match for E_Q*

(ii) *If $N_Q \subseteq N_P$ then E_P is a plug-in match for E_Q*

(iii) *If $(N_P \cap N_Q \neq \phi) \wedge (N_P \not\subseteq N_Q) \wedge (N_Q \not\subseteq N_P)$ then E_P is a weak match for E_Q*

(iv) *If $N_P \cap N_Q = \phi$ then E_P is a failed match for E_Q*

5.3 Matching Logical Constraint

The service will match to the query if, by applying a set of values obtained from the query into the rule expression, the rule evaluation hits and returns true as a result. The expression in the head atom of the rule may be a numerical constraint or constraint on some data values, and these may require ontological reasoning, numerical computation, and also rule reasoning. We consider a match only when such evaluation returns true.

5.4 Matching Process Model

To check whether a process specification satisfies the query, we consider matching on all aspects of the functional behaviour and the processes within the workflow. For

each aspect, it may need to perform ontological matching (Section 5.1) before considering other kind of constraint matching (Sections 5.2-5.3). The process specification will match the query if it satisfies the following:

(i) *input, unconditional output, unconditional effect, and process without guard satisfy ontological match in Section 5.1, and*

(ii) *precondition, conditional output, conditional effect, and process with guard satisfy relevant matching criteria in Section 5.2-5.3*

In other words, let $\mathbb{R}_Q$ and $\mathbb{R}_P$ be the sets of functional behaviour and workflow processes (with and without constraints) within the query and the process specification respectively:

ProcessModelMatch($\mathbb{R}_Q$, $\mathbb{R}_P$) = true $\Leftrightarrow$

$$(\mathbb{R}_Q \subseteq \mathbb{R}_P) \wedge (\forall i, \exists j : (i \in \mathbb{R}_Q) \wedge (j \in \mathbb{R}_P) \wedge (i \ \Theta \ j))$$

where Θ means having a kind of match as in Sections 5.1-5.3.

6 Process-Based Discovery

Assume a service consumer wants to apply for a 400,000-baht loan with a bank in order to buy a house. The consumer wants the bank that allows a loaner to apply for a credit card with no annual fee and approve the loan application at loan department level. This is to ensure that the loan process is quick. The consumer earns 20,000 bahts a month.

We present a query ($\mathbb{Q}$) as a collection of relation expressions. A relation expression is in the form of *property(subject, object)* which corresponds to an RDF statement <*subject, property, object*>. For a constraint that relates to a numerical value, such numerical constraint is represented as *property(argument, relationaloperator, literalvalue1, [literalvalue2,] unit)*. For the example above, the relation expressions are superscripted by symbols *C, E, G, and P* which refer to precondition, effect, guard, and process respectively:

$\mathbb{Q}$ ={hasIncomePerMonth(IncomePerMonth, 20000)^C ,

hasPremiumCreditCardFee(PremiumCreditCardFee, Equal, 0, baht)^E ,

hasLoanAmount(LoanAmount, 400000)^G ,

hasLoanPurpose(LoanPurpose, Housing)^G ,

hasProcess(Process, DepartmentApprovalProcess)^P }

To determine whether a service is a match, its process specification will also be treated as a collection of relation expressions in order to check against the set of relation expressions of the query. The rule expressions embedded in the process specification will be extracted and translated into a rule language in order to use a rule reasoning engine to check whether the rule is satisfied. In our implementation, SWRL rule will be translated into Jess script in order to use Jess engine [19].

If we look at S_1 and the query, to check whether the precondition holds for the query, we use the criterion to match numerical ranges (Section 5.2) and the consumer's income is an *exact match* and hence valid to use the service. To check the effect, we have to determine what S_1 will give as an effect since it is conditional. We first check the incondition by using matching of numerical ranges on the loan amount and the premium credit card effect is satisfied with an *exact match*. Then we use again the numerical range matching criterion to check whether the premium credit card offers 0 baht annual fee. This also returns an *exact match*. When all aspects of the functional behaviour of S_1 match to the query, S_1 is a potential service but we have to check further on its process flow. (In this example, the functional behaviour of S_2 and S_3 also matches to the query because we assume earlier that all three services exhibit the same functional behaviour.)

To consider the workflow of the service, we associate each process with guards that determine its performance. For example, the rules for all approval processes within the process specification of S_1 are listed below:

!hasLoanAmount(LoanAmount, LessThanOrEqual, 1000000, baht) →
    hasProcess(Process, HeadQuarterApprovalProcess);
hasLoanAmount(LoanAmount, LessThanOrEqual, 1000000, baht) →
    hasProcess(Process, LoanHistoryProcess);
hasLoanAmount(LoanAmount, LessThanOrEqual, 1000000, baht),
    hasLoanPurpose(LoanPurpose, RealEstate) →
    hasProcess(Process, DepartmentApprovalProcess);
hasLoanAmount(LoanAmount, LessThanOrEqual, 1000000, baht),
    !hasLoanPurpose(LoanPurpose, RealEstate) →
    hasProcess(Process, BranchApprovalProcess);

To check whether S_1 performs the requested process under the context of a particular query, we check whether the associated guards fire. This is possible when the information necessary for evaluating the guards can be obtained from the service consumer or from the process specification itself. In this example, the consumer requests for a department approval process. The first guard on loan amount fires with *exact match* by considering numerical ranges matching against the loan amount of the consumer. For the second guard on loan purpose, we first use ontological matching (Section 5.1) to check the ontological value RealEstate. Assume that there is a domain ontology which defines an IS-A taxonomy for RealEstate with subconcepts such as Housing and Land. S_1's purpose will be a *generalised match*, and by matching logical constraints (Section 5.3), this second guard will also fire. Therefore, S_1 will perform department approval process under the constraints placed by the query. When S_1 matches with all aspects defined in the query, it will be returned as a match to the consumer. With this approach, S_2 will fail to match the query because the consumer's loan purpose will not cause the loan purpose guard associated with its department approval process to fire. Similarly, S_3 will also fail to match the query because the consumer's loan amount does not satisfy the loan amount guard associated with its department approval process.

Process-based discovery is effective when a shared process ontology of a particular service domain is assumed. The shared process ontology defines common pattern of the process within a domain which includes internal tasks and relevant conditions.

This approach is possible as the concept of business process patterns exists [20], [21]. Service providers should publish process specifications that are derived from the domain process ontology, and service consumers should have some knowledge about the behaviour and workflow of the domain in order to compose an effective query. In our example, it should be commonly known that a bank loan process usually involves several classes of approval, and factors that influence the approvals include loan amount, loan purpose, and earning capability of the loaner. Although this process is internal to the bank, it is not classified business information since bank staff would normally give such information to the loaners. With a shared process ontology, the service consumer can submit a query without having to know other details of the candidate Web Services which may be considered as classified business rules; in our case, the service consumer does not need to know that the bank with a process specification such as S_1 has set a boundary of 1 million bahts for a head quarter approval. Process specifications are maintained by service providers; our approach does not require service consumers to have access to them.

7 Discovery Framework

The agent-based discovery framework in our previous work [6] is extended to accommodate process-based discovery. We develop the components within the architecture in Fig. 8 while also adopting existing ontology-based tools and rule engine.

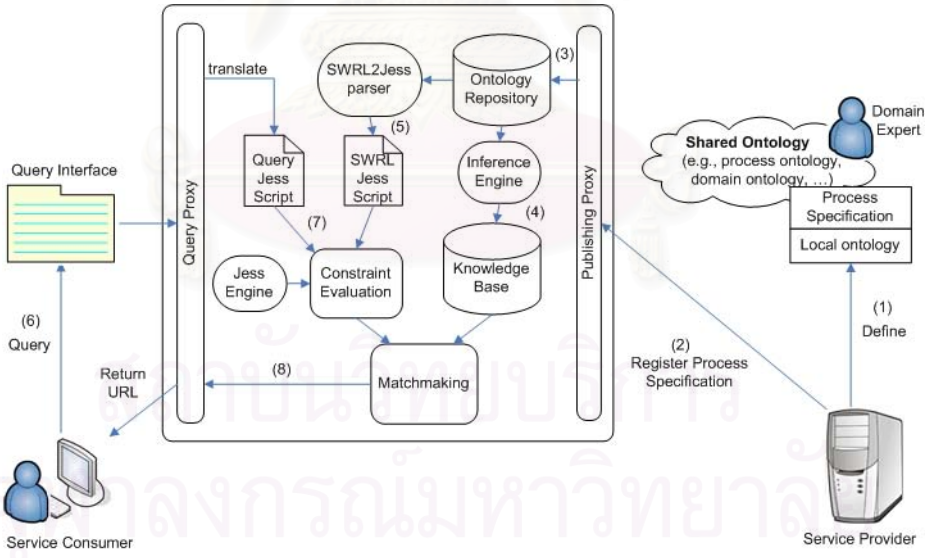


Fig. 8. Process-based discovery framework

In the figure, a service provider will define the process specification of the service as well as any necessary local ontology (1), using an ontology editor (e.g. Protégé). The

definition may be based on shared ontology of the domain, which is defined by service domain experts. The service provider maintains the process specification and the local ontology, but also registers the specification with the agent via the publishing proxy (2). The publishing proxy will store the URL of the process specification and local ontology in the ontology repository (3). The agent may preprocess to extract knowledge and to reason from the shared ontologies prior to the matchmaking by using an inference engine (e.g. Jena [22]); the results are stored in a knowledge base (4). At discovery time, the process specification will be processed and rule constraints are extracted and translated into a rule script by a parser (i.e. SWRL2Jess parser) (5). The agent can provide the service consumers with a GUI template that corresponds to the process ontology of the domain so that the consumers can specify query onto the process specifications more easily (6). Internally, the query will be translated into RDF-based relation expressions and will pass through the query proxy. Rule constraints in the query are translated into a rule script so that it is evaluated against constraints in the process specification (7). The constraint evaluation module is integrated with a rule engine (e.g. Jess engine). Matchmaking module considers matching criteria and reports the result in an XML document which will be returned to the consumer (8).

8 Conclusion

We present a new approach to service discovery by using OWL-S process model to model functional behaviour and workflow of the services and querying on such process specifications. Constraints can be placed on the functional behaviour and guard the flow of process execution. Matchmaking uses ontological reasoning and constraints evaluation to determine the actual behaviour of the services. Service consumers can then look for the services with a satisfied internal process.

The example in this paper shows a query concerning if-then-else and sequence constructs. Query based on other constructs is also meaningful and possible. We are in the process of finishing the integration of process-based discovery with the framework in [6] so that the integrated service profile is more complete and fits well with the service description model in Fig. 1.

References

1. Huhns, M. N., Singh, M. P.: Service-Oriented Computing: Key Concepts and Principles. IEEE Internet Computing. January-February (2005) 75-81
2. uddi.org: UDDI: Universal Description, Discovery, and Integration of Web Services (Online). (2002). http://www.uddi.org
3. Tapabut, C., Senivongse, T., Futatsugi, K.: Defining Attribute Templates for Descriptions of Distributed Services. In: Proceedings of 9th Asia-Pacific Software Engineering Conference (APSEC 2002), Gold Coast, Australia, December (2002) 425-434
4. Burstein, M. et al.: Semantic Web Services Architecture. IEEE Internet Computing. September-October (2005) 72-81

5. OWL-S Coalition. OWL-S 1.1 Release (online). http://www.daml.org/services/owl-s/1.1/
6. Sriharee, N., Senivongse, T.: Matchmaking and Ranking of Semantic Web Services Using Integrated Service Profile. To be published in International Journal of Metadata, Semantics and Ontologies, Vol. 1, No. 2, Inderscience Publishers
7. WSMO. Web Services Modeling Ontology (online). (2004). http://www.wsmo.org
8. Bruijn, D.J., Lausen, H., Polleres, A., Fensel, D.: The Web Service Modeling Language WSML: An Overview. DERI Technical Report, June 16 (2005)
9. Paolucci, M., Sycara, K.: UDDI Spec TC V4 Proposal Semantic Search (online). (2004). http://www.oasis-open.org/committees/uddi-spec/doc/req/uddi-spec-tc-req029-semanticsearch-20040308.doc
10. Trastour, D., Bartolini, C., Gonzalez-Castillo, J.: A Semantic Web Approach to Service Description for Matchmaking of Services. In: Proceedings of the International Semantic Web Working Symposium (SWWS'01) (2001)
11. Paolucci, M. et al.: Semantic Matching of Web Services Capabilities. In: Proceedings of the 1st International Semantic Web Conference (ISWC 2002), Sardinia (Italy), Lecture Notes in Computer Science, Vol. 2342. Springer Verlag (2002)
12. Sivashanmugan, K., Verma, K., Sheth, A., Miller, J.: Adding Semantics to Web Services Standards. In: Proceedings of the International Conference on Web Services (2003)
13. Srinivasan, N., Paolucci, M., Sycara, K.: An Efficient Algorithm for OWL-S Based Semantic Search in UDDI. In: Proceedings of 1st International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), San Diego, CA, USA, July 6, (2004)
14. Klein, M., Bernstein, A.: Searching for Services on the Semantic Web Using Process Ontologies. The Emerging Semantic Web – Selected papers from 1st Semantic Web Working Symposium. I. Cruz et al. (Eds.) IOS press, Amsterdam (2002) 159-172
15. Klusch, M., Gerber, A., Schmidt, M.: Semantic Web Service Composition Planning with OWLS-Xplan. In: Proceedings of 1st Intl. AAAI Fall Symposium on Agents and the Semantic Web, Arlington, VA, USA, AAAI Press (2005)
16. Sirin, E., Parsia, B.: Planning for Semantic Web Services. In Proceedings of Semantic Web Services Workshop at 3rd International Semantic Web Conference (ISWC'04) (2004)
17. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A Semantic Web Rule Language combining OWL and RuleML. (Online). (2003). http://daml.org/2003/11/swrl/
18. Li, L., Horrocks, I.: A Software Framework for Matchmaking Based on Semantic Web Technology. In: Proceedings of 12th International World Wide Web Conference (2003)
19. Jess the Rule Engine for the JAVA™ Platform. (online). http://herzberg.ca. sandia.gov/jess
20. Havey, M.: Essential Business Process Modeling. O'Rielly (2005)
21. Barros, O. H.: Business Information System Design Based on Process Patterns and Frameworks. (online). (2004). http://www.bptrends.com
22. Jena Semantic Web Framework: Jena. (online). http://jena.sourceforge.net/ index.html

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายปิยะ สุวรรณโนภาส เกิดเมื่อวันที่ 23 ตุลาคม พ.ศ. 2526 ที่จังหวัดกรุงเทพ มหานคร สำเร็จการศึกษาระดับปริญญาบัณฑิต หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ จาก จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2548 และได้เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ณ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ. 2548

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย