

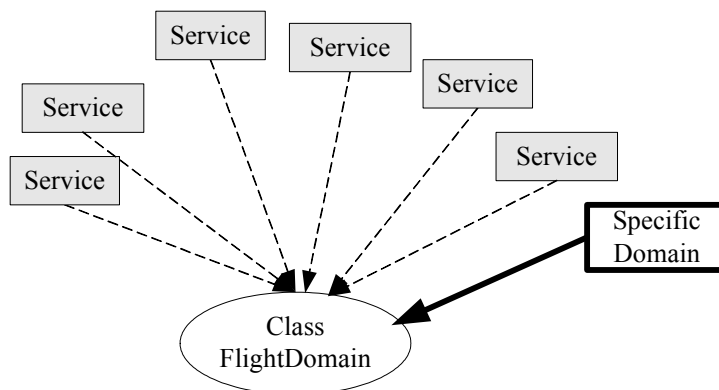
### บทที่ 3

## การเพิ่มคุณภาพของการค้นหาบริการโดยการกรองบริการการจับคู่โดยระบุโดเมน และการจับคู่โดยออนโทโลยีของชุดคำศัพท์

ในบทนี้จะอธิบายวิธีการที่นำเสนอในวิทยานิพนธ์ในการลดปัญหาการค้นหาแล้วได้ข้อมูลที่ผิดพลาดซึ่งเป็นที่มาของคุณภาพการค้นหาที่ต่ำซึ่งมีสองวิธี โดยการนิยามบริการด้วยโดเมนบริการและการนิยามบริการกับกลุ่มคำที่มีความหมายเหมือนกัน ซึ่งอยู่ในขั้นตอนที่ 2 ของแผนภาพขั้นตอนการค้นหาบริการในรูปที่ 2.31 ซึ่งในบทนี้จะอธิบายถึงหลักการและตัวอย่างการทดสอบของทั้งสองวิธี

### 3.1 การจับคู่โดยระบุโดเมน (Domain Specific Matching)

**Domain Specific Matching** ให้ผู้ใช้ระบุถึงโดเมนที่ชัดเจนเช่น เลือกโดเมนจาก combo box ที่เป็น Flight ระบบก็จะเลือกเอาทุกบริการที่เป็น Flight domain ออกมาดังในรูปที่ 3.1 วิธีนี้จะง่ายและถูกต้องแน่นอน ระบบไม่จำเป็นต้องวิเคราะห์ถึงคำสับสนเนื่องจาก มีตัวเลือกที่คงตัวให้ผู้ใช้เลือก จากวิธีนี้ค่า recall และ precision ที่วัดได้จะเป็น 1 หรือถูกต้อง 100 %

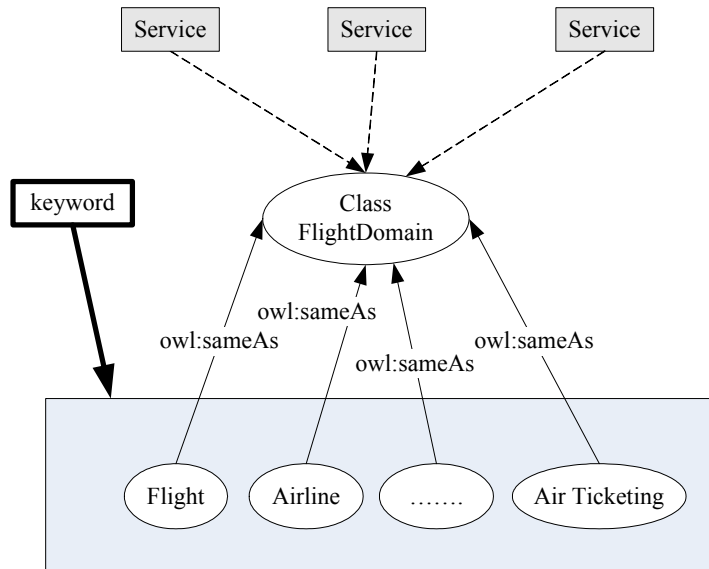


รูปที่ 3.1 การกรองบริการแบบระบุโดเมน

### 3.2 การจับคู่โดยออนโทโลยีของชุดคำศัพท์ (Words Catalog Ontology Matching)

**Words Catalog Ontology Matching** ผู้ใช้ระบุโดเมนในคำสับสน วิธีนี้จำเป็นต้องมีการสร้างชุดคำศัพท์ของคำที่มีความหมายเหมือนกันไว้สำหรับเปรียบเทียบกับคำสับสน เช่นกลุ่มคำที่มีความหมายเดียวกับ Flight domain คือ Flight, airline, air ticketing, air service... เนื่องจากผู้ใช้แต่ละคนอาจจะระบุคำสับสนที่แตกต่างกันไป ความถูกต้องและครอบคลุมของวิธีนี้จึงขึ้นอยู่กับ

กับความละเอียดและครอบคลุมเพียงพอของกลุ่มคำที่มีความหมายเหมือนกันดังรูปที่ 3.2



รูปที่ 3.2 การสร้างกลุ่มชุดคำศัพท์ของคำที่มีความหมายเหมือนกัน

โดยหลักการในวิธีนี้อาศัยคุณสมบัติ *sameAs* ของโอดับเบิลยูแอลในการสร้างกลุ่มคำที่มีความหมายเหมือนกัน และคุณสมบัติ *differentFrom* ในการสร้างข้อกำหนดว่าบางกลุ่มคำที่มีความหมายเหมือนกันแต่จะมีความแตกต่างกัน เช่น

Flight *sameAs* AirReservation, FlightInfo

Book *sameAs* Booking, AirReservation, FlightInfo, RoomReservation, Carrental, CarReservation

ในตัวอย่างหากระบุว่าค้นหาบริการกลุ่มคำที่มีความหมายเหมือนกับ Book อาจจะได้บริการที่เป็นโดเมนเกี่ยวกับโรงแรมและรถมาด้วยดังนั้นจึงควรมีการระบุ *differentFrom* เพื่อกำหนดความแตกต่างเช่น

AirReservation *differentFrom* RoomReservation, Carrental, CarReservation เป็นต้น

เมื่อใช้วิธีนี้อัลกอริทึมก็จะเลือกเอาบริการที่แตกต่างจาก AirReservation ไปเปรียบเทียบกับบริการที่ได้จากการสืบค้น Book เพื่อตัดบริการเหล่านั้นออกให้เหลือเพียงบริการที่เกี่ยวข้องกับโดเมนของการบินเท่านั้น

### 3.3 การทดลอง

#### 3.3.1 การนิยามบริการด้วยโดเมนบริการ

ในการทดสอบจะเริ่มต้นโดยทำการนิยามบริการเข้ากับออนโทโลยีโดเมนตั้งตัวอย่างง่าย ๆ

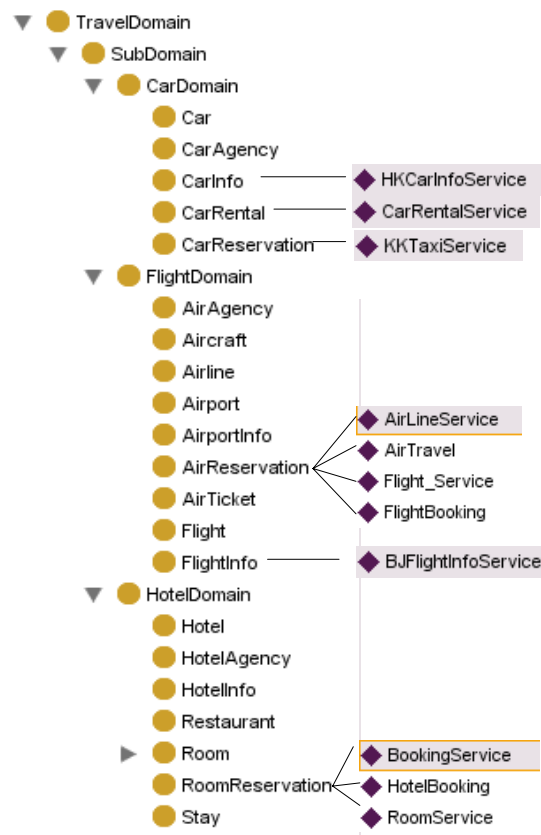
ใน รูปที่ 3.3 และรูปที่ 3.4

The image shows a web form for configuring a service. It contains the following elements:

- Service ID:** A text input field containing the placeholder text "xxxxxxxxxx".
- Service Name:** A text input field containing the text "FlightBooking".
- Service Domain:** A section header in italics.
- Travel:** A dropdown menu with "Flight" selected. The dropdown list is open, showing "Flight", "Hotel", and "Car".
- Shop:** A dropdown menu with "Book" selected.

รูปที่ 3.3 การระบุโดเมนให้กับบริการ

เมื่อผู้ให้บริการระบุโดเมนให้แก่บริการข้อมูลจะถูกเก็บไว้ในออนโทโลยีของโดเมนในรูปแบบของโอดับเบิ้ลยูแอล รูปที่ 3.4 เป็นตัวอย่างของบริการที่ทำการระบุโดเมนของแต่ละบริการ และในรูปที่ 3.5 เป็นตัวอย่างโค้ดภาษาโอดับเบิ้ลยูแอลของออนโทโลยีโดเมนบริการ



รูปที่ 3.4 ตัวอย่างบริการที่ลงทะเบียนกับออนโทโลยีโดเมน

```

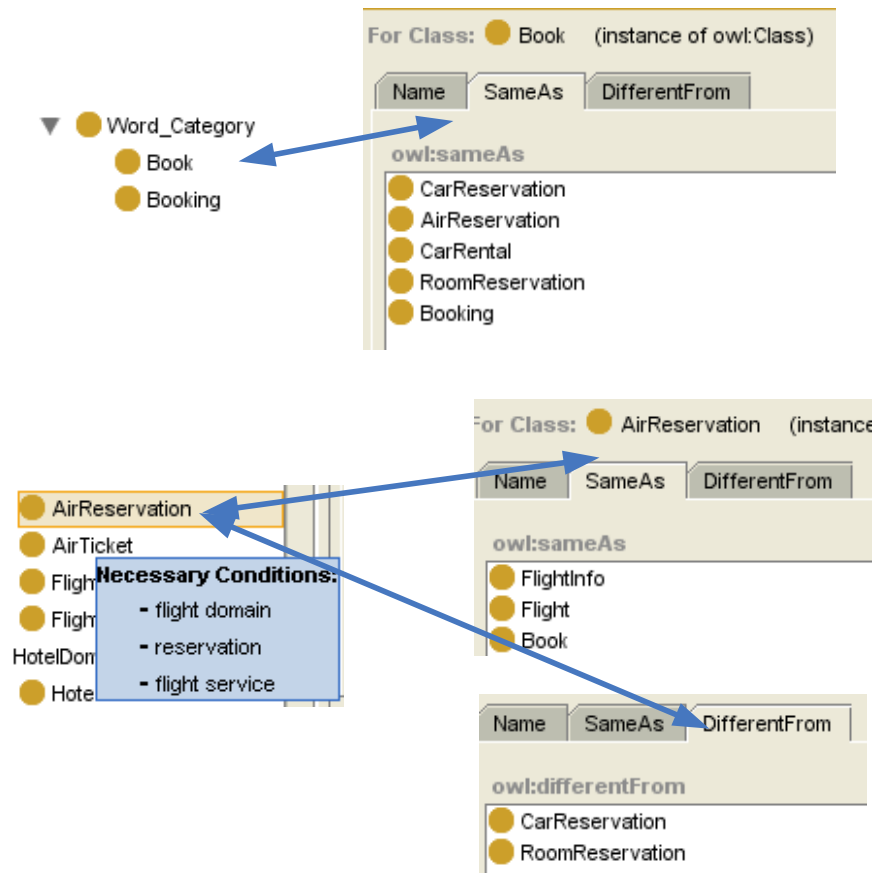
<AirReservation rdf:ID="FlightBooking">
  <hasServiceCategory rdf:resource="#FlightDomainService"/>
</AirReservation>
<AirReservation rdf:ID="AirTravel">
  <Airport rdf:ID="Suwannapoom"/>
<CarRental rdf:ID="CarRentalService"/>
<AirReservation rdf:ID="Flight_Service">
<FlightInfo rdf:ID="BJFlightInfoService"/> .....

```

รูปที่ 3.5 ตัวอย่างออนโทโลยีบริการในภาษาไอดีบีแอล

### 3.3.2 การนิยามกลุ่มคำที่มีความหมายเหมือนกัน

ในส่วนนี้จะทำการนิยามกลุ่มคำที่มีความหมายเหมือนกันและแตกต่างกันจากด้วยคุณสมบัติ *sameAs* และ *differentFrom* เป็นดังตัวอย่างในรูปที่ 3.6



รูปที่ 3.6 ตัวอย่างการนิยามกลุ่มคำที่มีความหมายเหมือนกัน

### 3.3.3 การทดลองค้นหาบริการเว็บเซอร์วิส

หัวข้อนี้เป็นนำเสนอส่วนติดต่อกับผู้ใช้ต้นแบบอย่างง่ายของระบบเพื่อเป็นแนวทางในการนำไปพัฒนาต่อ

#### ตัวอย่างกรณีที่ 1 การค้นหาแบบทั่วไปซึ่งมีบริการที่ต้องการอยู่ในระบบ

พิจารณาผู้ใช้ Warakorn ทำการค้นหาบริการซึ่งจองตั๋วเครื่องบินเดินทางจาก กทม. ไปยังหาดใหญ่ เริ่มจากขั้นตอนแรกผู้ใช้จะสืบค้นเพื่อให้ได้มาซึ่งบริการที่อยู่ในโดเมนที่ถูกต้องซึ่งในวิทยานิพนธ์นี้ได้นำเสนอการเลือกโดเมนจากที่ระบบมีให้เลือก (Domain Specific Matching) และการให้ผู้ใช้ระบุระบุชื่อและคำที่เกี่ยวข้องกับบริการ (Words Catalog Ontology Matching)

รูปที่ 3.7 ตัวอย่างการค้นหบริการโดยเลือกจากโดเมนบริการที่มีมาให้ ( Domain Specific Matching )

ขั้นตอนการค้นหบริการโดยแยกตามโดเมนที่ระบุจะมีคำสั่ง RDQL ดังตัวอย่างในรูปที่ 3.8

```
SELECT ?service
WHERE (?service, <rdf:type>, <p:AirReservation>)
USING p FOR <http://localhost:8080/travelontology.owl#>
```

รูปที่ 3.8 ตัวอย่างการสืบค้นโดเมนบริการ

```
service
=====
<http://localhost:8080/travelontology.owl#FlightBooking>
<http://localhost:8080/travelontology.owl#AirLineService>
<http://localhost:8080/travelontology.owl#AirTravel>
<http://localhost:8080/travelontology.owl#Flight_Service>
```

รูปที่ 3.9 ตัวอย่างบริการที่ได้จากการสืบค้นโดเมนบริการ

ซึ่งในตัวอย่างนี้จากการสืบค้นจากโดเมนที่เกี่ยวกับการจองตั๋วเครื่องบินจะได้รับการมาดังในรูปที่ 3.9 โดยตัวอย่างการนำหลักการนี้ไปประยุกต์ใช้สามารถดูเพิ่มเติมได้ในส่วนของภาคผนวก ก

ซึ่งเมื่อได้รับการที่เกี่ยวข้องกับชนิดหรือโดเมนที่ผู้ใช้ต้องการแล้วก็จะเข้าสู่การสืบค้นส่วนหน้าการทำงานของบริการโดยการระบุพารามิเตอร์ของบริการ

นอกจากการเลือกบริการจากโดเมนที่ระบบกำหนดให้ผู้ใช้ยังสามารถค้นหบริการตามการระบุชื่อหรือข้อมูลที่เกี่ยวข้องกับบริการ ซึ่งจะต่างกับการสืบค้นด้วยสำคัญคือจะนำค่าที่ผู้ใช้ระบุไปเปรียบเทียบกับกลุ่มคำศัพท์ที่นิยามขึ้นไว้เป็นออนโทโลยีของกลุ่มคำที่มีความหมายเดียวกัน จากนั้นจึงทำการเลือกบริการที่เกี่ยวข้องกับโดเมนที่เป็นเรื่องเดียวกับกลุ่มคำเหล่านี้ ดังตัวอย่าง

การสืบค้นในแต่ละกรณีดังนี้

### ตัวอย่างการสืบค้นด้วย “flight booking”

ตัวอย่างนี้เป็นการระบุคำสืบค้น มากกว่าหนึ่งคำซึ่งจะใช้วิธีแยกคำไปสืบค้นแล้วนำผลที่ได้มาเปรียบเทียบเพื่อเลือกบริการผลลัพธ์ที่เหมือนกันของทั้งสองคำสืบค้นมาเป็นผลที่ได้

รูปที่ 3.10 ตัวอย่างการค้นหบริการโดยผู้ใช้ระบุตามที่ต้องการ

ส่วนนี้จะทำการแยกคำเพื่อนำไปสืบค้นคำที่มีความหมายเหมือนกันเพื่อนำไปค้นหาบริการจากโดเมนที่พบต่อไป ซึ่งคำสืบค้นด้วย RDQL จะเป็นไปดังในรูปที่ 3.11 และผลที่ได้จากการสืบค้นจะเป็นไปดังในรูปที่ 3.12

```
SELECT ?srv
WHERE (?word, <owl:sameAs>, <p:Booking>)
      (?services, <owl:sameAs>, ?word)
      (?srv, <rdf:type>, ?services)
USING p FOR <http://localhost:8080/travelontology.owl#>
```

```
SELECT ?srv
WHERE (?word, <owl:sameAs>, <p:Flight>)
      (?services, <owl:sameAs>, ?word)
      (?srv, <rdf:type>, ?services)
USING p FOR <http://localhost:8080/travelontology.owl#>
```

รูปที่ 3.11 ตัวอย่างการสืบค้นโดยใช้กลุ่มชุดคำศัพท์ของคำที่มีความหมายเดียวกัน

```

SRV
=====
<http://localhost:8080/travelontology.owl#Flight_Service>
<http://localhost:8080/travelontology.owl#AirTravel>
<http://localhost:8080/travelontology.owl#FlightBooking>
<http://localhost:8080/travelontology.owl#AirLineService>
<http://localhost:8080/travelontology.owl#CarRentalService>
<http://localhost:8080/travelontology.owl#KKTaxiService>
<http://localhost:8080/travelontology.owl#RoomService>
<http://localhost:8080/travelontology.owl#HotelBooking>
<http://localhost:8080/travelontology.owl#BookingService>

SRV
=====
<http://localhost:8080/travelontology.owl#Flight_Service>
<http://localhost:8080/travelontology.owl#AirTravel>
<http://localhost:8080/travelontology.owl#FlightBooking>
<http://localhost:8080/travelontology.owl#AirLineService>
<http://localhost:8080/travelontology.owl#BJFlightInfoService>

```

รูปที่ 3.12 ผลที่ได้จากการสืบค้นในรูปที่ 3.11

จากรูปที่ 3.12 จะเห็นว่าจากคำสืบค้นแรกค้นจากกลุ่มคำที่มีความหมายเหมือนกับ booking จะได้บริการออกมามากมายรวมถึงบริการในโดเมนอื่น ๆ เนื่องจาก booking ถูกจัดให้อยู่ในกลุ่มคำความหมายเดียวกับ book ดังนั้นจึงดึงบริการที่เกี่ยวข้องกับ book ออกมาด้วย ส่วนคำสืบค้นที่ค้นจากกลุ่มคำที่มีความหมายเหมือนกับ flight นั้นจะได้เฉพาะบริการที่มาจากกลุ่มคำในโดเมนของการบิน ดังนั้นบริการที่ถูกเลือกจากการสืบค้นในตัวอย่างนี้จะเป็นบริการที่เหมือนกันของคำสืบค้นทั้งสองได้แก่ FlightBooking, AirLineService, AirTravel และ Flight\_Service

#### ตัวอย่างการสืบค้นด้วย “airReservation”

ตัวอย่างนี้จะเป็นการสืบค้นจากกลุ่มคำซึ่งอาจจะทำให้ได้ข้อมูลจากโดเมนอื่นมาด้วย เนื่องจาก airReservation ถูกจัดให้อยู่ในกลุ่มเดียวกันกับ book ซึ่ง book จะหมายถึงรวมถึง carRental, carReservation และ roomReservation ด้วย ดังนั้นบริการที่ได้จะมีบริการจากโดเมนอื่นมาปะปนดังนั้นจึงควรมีการตั้งคุณสมบัติ differentFrom เข้ามาช่วย โดยการสืบค้นกลุ่มคำที่มีความแตกต่างจาก airReservation แล้วนำบริการที่ได้จากส่วนนี้ไปลบออกจากกลุ่มบริการที่ได้จากการสืบค้นในคำสืบค้นแรกดังในรูปที่ 3.13 และรูปที่ 3.14



```

SELECT    ?srv
WHERE     (?word, <owl:sameAs>, <p:AirReservation>)
          (?services, <owl:sameAs>, ?word)
          (?srv, <rdf:type>, ?services)
USING    p FOR <http://localhost:8080/travelontology.owl#>

```

```

SELECT    ?srv
WHERE     (?word, <owl:differentFrom>, <p:AirReservation>)
          (?srv, <rdf:type>, ?word)
USING    p FOR <http://localhost:8080/travelontology.owl#>

```

รูปที่ 3.13 ตัวอย่างที่สองของการสืบค้นโดยใช้กลุ่มชุดคำศัพท์ของคำที่มีความหมายเดียวกัน

```

srv
=====
<http://localhost:8080/travelontology.owl#Flight_Service>
<http://localhost:8080/travelontology.owl#AirTravel>
<http://localhost:8080/travelontology.owl#FlightBooking>
<http://localhost:8080/travelontology.owl#AirLineService>
<http://localhost:8080/travelontology.owl#BJFlightInfoService>
<http://localhost:8080/travelontology.owl#Flight_Service>
<http://localhost:8080/travelontology.owl#AirTravel>
<http://localhost:8080/travelontology.owl#FlightBooking>
<http://localhost:8080/travelontology.owl#AirLineService>
<http://localhost:8080/travelontology.owl#Flight_Service>
<http://localhost:8080/travelontology.owl#AirTravel>
<http://localhost:8080/travelontology.owl#FlightBooking>
<http://localhost:8080/travelontology.owl#AirLineService>
<http://localhost:8080/travelontology.owl#CarRentalService>
<http://localhost:8080/travelontology.owl#KKTaxiService>
<http://localhost:8080/travelontology.owl#RoomService>
<http://localhost:8080/travelontology.owl#HotelBooking>
<http://localhost:8080/travelontology.owl#BookingService>

srv
=====
<http://localhost:8080/travelontology.owl#KKTaxiService>
<http://localhost:8080/travelontology.owl#RoomService>
<http://localhost:8080/travelontology.owl#HotelBooking>
<http://localhost:8080/travelontology.owl#BookingService>

```

รูปที่ 3.14 ผลที่ได้จากการสืบค้นในรูปที่ 3.13

จากรูปที่ 3.14 เมื่อนำผลที่ได้จากการสืบค้นด้วยกลุ่มคำที่มีความแตกต่างกับ airReservation ไปเปรียบเทียบเพื่อตัดบริการเหล่านั้นออกจากบริการที่ได้จากคำสืบค้นแรก และเมื่อตัดบริการที่ซ้ำออกจะได้บริการที่ได้รับเลือกคือ FlightBooking, AirLineService, AirTravel และ Flight\_Service ซึ่งกับที่ได้ในวิธีแรกที่เลือกจากโดเมนโดยตรง

นอกจากนี้หากคำสืบค้นที่ผู้ใช้ระบุมีหลายคำจะแยกมาหากกลุ่มคำที่มีความหมายเดียวกัน จากนั้นเมื่อได้บริการจะนำบริการที่ได้จากแต่ละคำมาทำการเปรียบเทียบหาบริการที่เหมือนกันต่อไป โดยตัวอย่างการนำหลักการนี้ไปประยุกต์และทดสอบสามารถดูเพิ่มเติมได้ในภาคผนวก ก การขั้นตอนการกรองบริการนี้จะทำให้ได้บริการที่ตรงกับชนิดหรือโดเมนที่ต้องการเพื่อนำไปค้นหาต่อในขั้นตอนต่อไปซึ่งเป็นการเปรียบเทียบพารามิเตอร์ที่ผู้ใช้ระบุกับที่บริการเหล่านั้น

มีดังในรูปที่ 3.15

**Service Domain** : Travel > Flight  
> AirReservation

Query

**Parameters**

DepartureAirport

ArrivalAirport

DepartureDate

ArrivalDate

AirSrcvClass

Airline

FlightType

Price

AirSeat

รูปที่ 3.15 ทำการระบุพารามิเตอร์หลังจากได้บริการในโดเมนที่ถูกต้อง

ในขั้นตอนนี้จะสืบค้นโดยจับคู่ระหว่างพารามิเตอร์ที่ผู้ใช้ระบุกับที่บริการนิยามไว้ซึ่งผลที่ได้จะเป็นดังตัวอย่างในตารางที่ 4

ตารางที่ 4 ตัวอย่างรายละเอียดของบริการซึ่งแสดงพารามิเตอร์ที่ตรงกับที่ผู้ใช้ระบุ

Service	Domain	parameter
AirlineService	Flight	DepartureAirport, ArrivalAirport, DepartureDate, ArrivalDate, AirSrcvClass, Airline
FlightBooking	Flight	DepartureAirport, ArrivalAirport, ArrivalDate, FlightType, DepartureDate, AirSrcvClass, Price
Air Travel	Flight	DepartureAirport, ArrivalAirport, DepartureDate, ArrivalDate, AirSeat, Airline
FlightService	Flight	DepartureAirport, ArrivalAirport, ArrivalDate, DepartureDate, AirSrcvClass, FlightType

ซึ่งจะเห็นได้ว่าบริการทั้ง 4 บริการนี้มีคุณสมบัติที่สามารถตอบสนองตามที่ผู้ร้องขอทั้งหมดคือ DepartureAirport, ArrivalAirport, ArrivalDate

### 3.4 การวัดคุณภาพการค้นหา

ซึ่งในการค้นหาบริการจากโดเมนด้วยสองวิธีข้างต้นเมื่อเปรียบเทียบกับการค้นหาจาก

ระบบแบบทั่วไปซึ่งใช้แบบค้นจากคำสำคัญ book และ flight จะได้ผลดังตารางที่ 5

ตารางที่ 5 บริการที่ได้จากการค้นหาด้วยวิธีต่างๆ

	<b>Keyword-based mechanisms</b>	<b>Domain Specific Matching</b>	<b>Words Catalog Ontology Matching</b>
Discovery result (services)	<b>Flight_Service</b> <b>FlightBooking</b> BJFlightInfoService BookingService HotelBooking	<b>FlightBooking</b> <b>AirLineService</b> <b>AirTravel</b> Flight_Service	<b>FlightBooking</b> <b>AirLineService</b> <b>AirTravel</b> Flight_Service

### 3.4.1 การวัดค่า Recall และ Precision

ซึ่งกฎเกณฑ์ใช้ในการวัดคุณภาพในการค้นหาข้อมูล 2 ค่า ซึ่งนำมาประยุกต์ใช้ในการวัดคุณภาพการค้นหาบริการได้แก่ Recall และ Precision นั้น เมื่อวัดจากผลที่ได้ในตารางที่ 5 จะได้ดังที่ปรากฏในตารางที่ 6

ตารางที่ 6 ค่า Recall และ Precision ที่ได้จากการค้นหาได้ของวิธีแต่ละแบบ

	<b>Relevant services</b>	<b>Keyword-based mechanisms</b>	<b>Domain Specific Matching</b>	<b>Words Catalog Ontology Matching</b>
Discovery result (services)	4	2 relevant 3 irrelevant	4 relevant	4 relevant
Precision (%)	-	$2/5 = 40\%$	$4/4 = 100\%$	$4/4 = 100\%$
Recall (%)	-	$2/4 = 50\%$	$4/4 = 100\%$	$4/4 = 100\%$

จากตารางที่ 6 การค้นหาแบบคำสำคัญจะเช็คจากชื่อบริการที่มีคำว่า ‘book’ และ ‘flight’ ซึ่งจะทำให้ได้บริการที่มีคำพวกนี้อยู่ในชื่อ แต่อาจจะไม่เกี่ยวข้องกับชนิดหรือโดเมนบริการที่ต้องการเช่นในตัวอย่างนี้ BJFlightInfoService, BookingService, HotelBooking ไม่ได้อยู่ในโดเมนที่เกี่ยวข้องกับที่ผู้ใช้ต้องการคือบริการจองตั๋วเครื่องบิน แต่กลับอยู่ในผลการค้นหาด้วยซึ่งเมื่อวัดค่า precision โดยการนำจำนวนบริการที่เกี่ยวข้องกับที่ต้องการซึ่งได้รับมาจากการค้นหาซึ่งมีแค่ 2 บริการ (Flight\_Service, FlightBooking) หาดด้วยจำนวนบริการทั้งหมดที่ได้รับกลับมาจากการค้นหาคือ 5 บริการ (Flight\_Service, FlightBooking, BJFlightInfoService, BookingService, HotelBooking) จะได้เป็นค่าคุณภาพการค้นหาเป็น 40% ในขณะที่เมื่อวัดค่า recall โดยการนำจำนวนบริการที่เกี่ยวข้องกับที่ต้องการซึ่งได้รับมาจากการค้นหาซึ่งมีแค่ 2 บริการ

(Flight\_Service, FlightBooking) ทหารด้วยจำนวนบริการที่เกี่ยวข้องกับที่ต้องการทั้งหมดที่มีอยู่ในระบบซึ่งมีอยู่ 4 บริการ (FlightBooking, AirLineService, AirTravel, Flight\_Service) จะได้ค่าคุณภาพการค้นหาเป็น 50% ในขณะที่อีก 2 วิธีจะได้บริการซึ่งเกี่ยวข้องกับที่ต้องการทั้งหมดออกมาทำให้คุณภาพการค้นหาเป็น 100 % แม้นในคำสืบค้นจะมีคำที่สื่อความหมายได้แตกต่างกัน แต่จะได้บริการในโดเมนที่ถูกต้อง นั่นคือไม่มีบริการที่ไม่ต้องการได้รับกลับมาและได้รับบริการที่ต้องการครบถ้วน

จะเห็นได้ว่าระบบไม่ต้องการมีการสืบค้นหลายครั้งเพียงแต่นำคำสืบค้นมาเปรียบเทียบกับกลุ่มชุดคำศัพท์ของคำที่มีความหมายเหมือนกันก็จะสามารถระบุไปยังโดเมนนั้น ๆ เพื่อดึงเอาข้อมูลทั้งหมดที่สร้างความสัมพันธ์หรือลงทะเบียนไว้กับคลาสนั้น ๆ ออกมาได้เลย

นอกจากนี้การนิยามข้อมูลด้วยภาษาโอบีดับเบิลยูแอล จาก Protege ยังช่วยป้องกันให้ข้อมูลไม่เกิดความซ้ำซ้อน เนื่องจากไม่สามารถสร้างคลาสที่มีชื่อเหมือนกันได้ ซึ่งจาก 2 วิธีข้างต้นนั้นมีข้อดีและข้อเสียแตกต่างกันไปดังที่แสดงในตารางที่ 7

ตารางที่ 7 เปรียบเทียบระหว่างวิธี Domain Specific กับวิธี Words Catalog Ontology

	Domain Specific	Words Catalog Ontology
ข้อดี	คุณภาพในการสืบค้นสูง (high Recall and high Precision)	มีความยืดหยุ่น ผู้ใช้สามารถระบุคำสืบค้นตามต้องการ
ข้อด้อย	ไม่มีความยืดหยุ่น ผู้ใช้ไม่สามารถระบุคำสืบค้นที่ต้องการได้	คุณภาพการสืบค้นขึ้นอยู่กับความละเอียดของกลุ่มคำเหมือน

### 3.4.2 การวัดระดับความเหมือนของการจับคู่

การวัดนี้จะทำการตรวจสอบว่าบริการที่ได้มามีคุณสมบัติหรือพารามิเตอร์ตรงกับที่ผู้ใช้ระบุอยู่ในระดับใด ซึ่งระดับความเหมือนของการจับคู่ (Matching Degree) ของแต่ละบริการจะเป็นดังในตารางที่ 8

ตารางที่ 8 แสดงระดับความเหมือนของการจับคู่และชนิดของการจับคู่ของแต่ละบริการ

R = 3	S	K	Type of Matching
AirlineService	6	3	Plugin
FlightBooking	7	3	Plugin
Air Travel	6	3	Plugin
FlightService	6	3	Plugin

จากตารางที่ 8 ค่า  $R$  คือจำนวนคุณสมบัติของบริการที่ผู้ใช้งานซึ่งในตัวอย่างนี้คือ 3 ค่า  $S$  เป็นจำนวนคุณสมบัติของแต่ละบริการมี ส่วนค่า  $K$  เป็นจำนวนคุณสมบัติที่แต่ละบริการมีเหมือนกับที่ผู้ใช้งาน ซึ่งในตัวอย่างนี้ทุกอย่างมีคุณสมบัติดังที่ผู้ใช้งานคือ 3 บริการทั้งหมด ซึ่งการที่แต่ละบริการมีคุณสมบัติ มากกว่าที่ผู้ใช้งานแต่สามารถตอบสนองตามที่ผู้ใช้งานต้องการได้หมดนั้น การจับคู่แบบนี้ถูกจัดให้เป็นชนิดการจับคู่แบบ การจับคู่แบบเสริม (Plugin) ซึ่งถือว่าดีที่สุดในอันดับสองและถือว่าการจับคู่ที่มีความแม่นยำสูง

### 3.5 สรุป

ในบทนี้อธิบายถึงวิธีการและการทดสอบตัวอย่างการนิยามบริการกับโดเมนออนไลน์ เพื่อช่วยให้การค้นหบริการซึ่งเลือกจากแต่ละโดเมนให้คุณภาพของการค้นหาที่สูงเนื่องจากได้บริการที่ต้องการครบถ้วนและไม่มีบริการที่ไม่เกี่ยวข้องปะปนอยู่ นอกจากนี้ยังได้แสดงตัวอย่างการวัดระดับความเหมือนของการจับคู่ของพารามิเตอร์ในแต่ละบริการอีกด้วย

ซึ่งในบทต่อไปจะอธิบายถึงขั้นตอนการค้นหบริการที่ต่อจากในบทนี้ซึ่งเป็นการนำข้อมูลความชอบของผู้ใช้มาช่วยในการเลือกบริการที่มีความใกล้เคียงกับที่ผู้ใช้งานต้องการมากที่สุด