

การสืบค้นเว็บเซอร์วิสแบบสื่อความหมาย โดยใช้ซอฟต์แวร์เอเจนต์

Web Service Discovery using Semantic-based Service Search Agent

สมเกียรติ สอนนวล งามนิจ อัจฉรินทร์ สมจิตร อัจฉรินทร์ วีระยุทธ ทองเครือ

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น

E-mail : somkiet.sorn@kku.ac.th, ngamnij@kku.ac.th, somjit@kku.ac.th, theerayut@kku.ac.th

บทคัดย่อ

จุดประสงค์ของงานวิจัยนี้เพื่อนำเสนอวิธีการเก็บรวบรวมเอกสารอธิบายเว็บเซอร์วิสจากแหล่งต่างๆ ในอินเทอร์เน็ต โดยใช้ตัวแทนการค้นหาคำหรือซอฟต์แวร์เอเจนต์ ช่วยเก็บรวบรวมข้อมูลคำอธิบายเว็บเซอร์วิส แล้วนำข้อมูลที่ได้มาวิเคราะห์หาคำสำคัญและให้คะแนนความสำคัญแก่คำเหล่านั้น เพื่อจัดกลุ่มคำและเอกสาร เว็บเซอร์วิสที่สัมพันธ์กัน โดยอาศัยฐานข้อมูลเวิร์ดเน็ตช่วยในการพิจารณาความคล้ายคลึงกันของคำอย่าง สื่อความหมาย ผลลัพธ์ของคำที่มีการจัดกลุ่มกันจะถูกเก็บอยู่ในโครงสร้างออนโทโลยี เพื่อให้ผู้ใช้งานเว็บ เซอร์วิสเข้ามาสืบค้นเว็บเซอร์วิสที่ต้องการได้จากระบบอย่างสื่อความหมายซึ่งจะทำให้ผู้ใช้งานสามารถ สืบค้นข้อมูลเว็บเซอร์วิสได้ตรงกับความต้องการอย่างแท้จริง

คำสำคัญ :

เว็บเซอร์วิสแบบสื่อความหมาย ซอฟต์แวร์เอเจนต์

Abstract

The purpose of this research is to collect data and categorize web service by using a new method. Many special software agents are used to gather information from several web service sources. The information gathered will then go through several calculation processes to determine their keywords. The algorithm also involves scoring keywords to rank their importance. WordNet is used in many places along the process to link the semantic of the keywords to make the grouping of keywords more effective. Once all of the keywords are grouped, they will be put into ontology for user to use as a search engine database—serving users need semantically, and effectively.

Keyword :

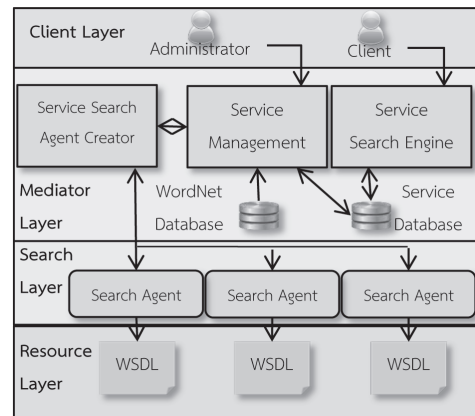
semantic webservice, software agent

1. บทนำ

ปัจจุบันเทคโนโลยีเว็บเซอร์วิส (Web service) [1] ได้ถูกออกแบบขึ้นมาเพื่อเป็นตัวกลางการสื่อสารระหว่างระบบที่มีความแตกต่างกันให้ทำงานร่วมกันได้ สามารถแก้ปัญหาการบูรณาการข้อมูลจากแหล่งข้อมูลที่หลากหลาย และจะต้องมีการประกาศข้อมูลการให้บริการและคำอธิบายวิธีการเรียกใช้เรียกว่า WSDL [2] เพื่อผู้ใช้บริการเข้ามาค้นหาบริการเว็บเซอร์วิสตามต้องการซึ่งการค้นหาเว็บเซอร์วิสนั้นยังคงเป็นลักษณะ keyword-based อยู่ทำให้ได้ข้อมูลที่ไม่ตรงกับที่ต้องการหรือเกิดความซ้ำซ้อนกันของเว็บเซอร์วิสในงานวิจัยนี้จึงได้นำเสนอวิธีการใช้ซอฟต์แวร์เอเจนต์ (Software Agent) [3] เป็นตัวกลางในการสื่อสารและเก็บรวบรวมเอกสารคำอธิบายเว็บเซอร์วิสจากแหล่งให้บริการเว็บเซอร์วิสต่างๆ แล้วนำเอกสารเหล่านั้นมาประมวลผลสรุปหาคำสำคัญ และให้คะแนนความสำคัญแก่คำเหล่านั้น โดยอาศัยฐานข้อมูล WordNet [4] ช่วยในการหาความหมายและความสัมพันธ์ของคำต่างๆ เพื่อจัดกลุ่มเอกสารได้อย่างสื่อความหมาย ซึ่งเมื่อผู้ใช้งานเว็บเซอร์วิสเข้ามาสืบค้นเว็บเซอร์วิสจากระบบก็จะทำให้สามารถสืบค้นข้อมูลได้อย่างสื่อความหมาย ทำให้ได้ผลลัพธ์ตรงตามความหมายที่ต้องการมากที่สุด

2. การออกแบบระบบ

ผู้วิจัยได้ออกแบบสถาปัตยกรรมของระบบเพื่อใช้สืบค้นเว็บเซอร์วิสจากแหล่งข้อมูลที่สามารถเข้าถึงได้ผ่านอินเทอร์เน็ตโดยใช้ซอฟต์แวร์เอเจนต์ แล้วนำมาสกัดเอาคำสำคัญและความหมายของเว็บเซอร์วิสอย่างอัตโนมัติ แล้วนำมาจัดกลุ่มเข้าไปในออนโทโลยี (Ontology) [5] กลางตามความหมายที่ได้ เพื่อแก้ปัญหาการค้นหาเว็บเซอร์วิสที่มีอยู่มากมายในอินเทอร์เน็ตซึ่งมีส่วนประกอบดังแสดงในภาพที่ 1



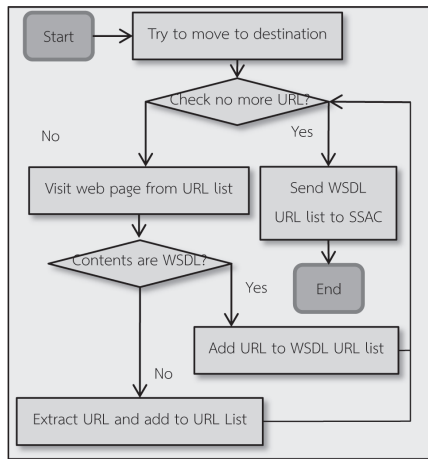
ภาพที่ 1 สถาปัตยกรรมของระบบ

2.1 ระดับชั้นแหล่งข้อมูล (Resource Layer)

ในส่วนนี้เป็นเอกสารคำอธิบายเว็บเซอร์วิสจากแหล่งต่างๆ ที่ Search Agent จะเข้าไปสืบค้นเพื่อนำมาหาความหมายและจัดกลุ่ม

2.2 ระดับชั้นค้นหา (Search Layer)

ในส่วนนี้เป็นส่วนที่ Search Agent เคลื่อนเข้าไปสืบค้นข้อมูลจากแหล่งต่างๆ ในอินเทอร์เน็ตโดย Search Agent จะทำงานภายใต้ Agent Container โดยมีขั้นตอนการทำงานโดยรวมคือ หากปลายทางมี Container ชนิดเดียวกัน และได้เชื่อมต่อถึงกันแล้ว Search Agent ที่เป็นชนิด Mobile Agent [6] นี้จะพยายามย้ายตัวเองไปที่ปลายทาง แล้วเริ่มทำงานโดยเข้าไปที่ IP เริ่มต้นที่ได้รับมาเพื่อเปิดดูหน้าเพจ แล้วตรวจสอบหน้าเพจนั้นว่าเป็น WSDL หรือไม่ หากเป็น WSDL ก็จะเก็บเป็นผลการค้นหา หากไม่ใช่ WSDL ก็จะสกัดเอาเฉพาะ URL ที่ปรากฏในหน้าเพจนั้นเก็บไว้ที่รายการ URL ที่จะสืบค้นต่อไป จากนั้น Search Agent จะตรวจสอบหน้าเพจต่อไปจากรายการ URL ที่จะสืบค้น ซึ่ง Search Agent จะทำงานวนไปเรื่อยๆ จนกว่ารายการ URL จะหมดหรือได้ผลลัพธ์ตามจำนวนที่ต้องการ แล้วจึงจะส่งผลการค้นหาค้นกลับไปที่ Service Search Agent Creator (SSAC) และจบการทำงานตัวเองลง ดังแสดงในภาพที่ 2

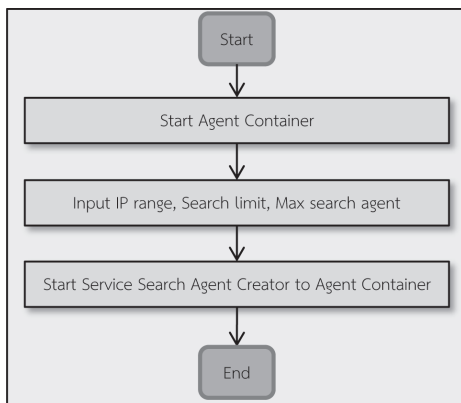


ภาพที่ 2 การทำงานของ Search Agent

2.3 ระดับชั้นตัวกลางของระบบ (Mediator Layer)

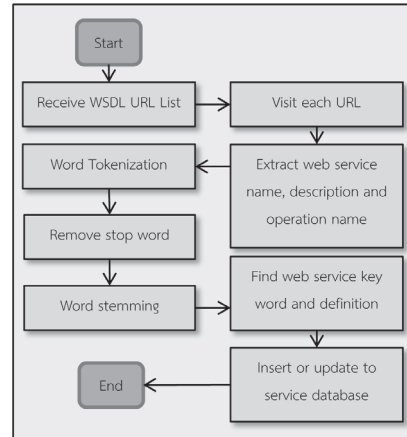
1) โมดูล Service Management มีการทำงานสองส่วน คือ

ส่วนแรกเริ่มการทำงาน Service Search Agent Creator เริ่มต้นจากการเริ่มการทำงาน Agent Container ก่อน จากนั้นจึงกำหนดข้อกำหนดการทำงานของ Service Search AgentCreator ได้แก่ ขอบเขต IP ที่จะค้นหา ผลลัพธ์สูงสุดที่ต้องการ จำนวน Search Agent สูงสุดที่จะทำงานพร้อมกัน จากนั้นจึงเริ่มการทำงาน Service Search AgentCreator ให้เริ่มสร้าง Search Agent ดังแสดงในภาพที่ 3



ภาพที่ 3 ขั้นตอนเริ่มการทำงาน Service Search AgentCreator

การทำงานส่วนที่สองของ Service Management คือการประมวลผลคำอธิบายเว็บเซอร์วิสดังแสดงในภาพที่ 4



ภาพที่ 4 ขั้นตอนการประมวลผลข้อมูลของ Service Management

Service Management เข้าไปยังแต่ละ URL ของรายการ WSDL ที่ได้รับมาอีกครั้งเพื่อสกัดเอาข้อมูลดังนี้

- คำอธิบายเว็บเซอร์วิสและแต่ละ operation
- ```
<documentation>value</documentation>
```
- ชื่อของแต่ละโอเปอเรชันจาก
- ```
<operation name="value" ></operation>
```
- ชื่อเว็บเซอร์วิสจาก<service name="value"></service>

จากนั้นนำเอาข้อมูลที่ได้มาตัดอักษรพิเศษต่าง ๆ ออกให้เหลือเพียงคำที่เป็นตัวอักษรภาษาอังกฤษเท่านั้น และรวมช่องว่างระหว่างคำ (space) ให้เหลือช่องเดียว แล้วแยกคำที่ติดกันโดยแยกคำใหม่จากคำที่เริ่มด้วยตัวอักษรตัวพิมพ์ใหญ่ เช่น GetWeatherForecastByCity จะแยกได้ 5 คำดังนี้ Get, Weather, Forecast, By และ City เป็นต้น จากนั้นตัดคำขยาย (stop word) ออกแล้วการทำ word stemming คือการทำให้คำใน

รูปต่างๆ กลับมาเป็นรูปสามัญโดยอาศัยฐานข้อมูล คำศัพท์ WordNet ตัวอย่างเช่น คำว่า cats เปลี่ยน กลับเป็น cat เป็นต้น

จากนั้นจะเป็นการให้คะแนนคำแต่ละคำ ดังแสดง ตามอัลกอริทึมต่อไปนี้

(ขั้นตอนการกำหนดชุดข้อมูลที่จะบันทึกข้อมูล)

- ให้ $Word_i$ คือ คำที่จะประมวลผล โดยที่ i คือตัวเลขจำนวนเต็มที่เป็นลำดับของคำสำคัญ
- ให้ $WordScore_i$ คะแนนของ $Word_i$ โดยที่ i คือตัวเลขที่เป็นลำดับของ $Word_i$
- ให้ $WordSynset_i$ คือ ชุดของคำพ้องความหมายของ $Word_i$ โดยที่ i คือ ตัวเลขจำนวนเต็มที่เป็นลำดับของ $Word_i$

(ขั้นตอนการให้คะแนนจากการนับจำนวนคำซ้ำ)

- นับจำนวนคำซ้ำเก็บใน $WordScore_i$ ของคำนั้น และลบคำที่ซ้ำออก

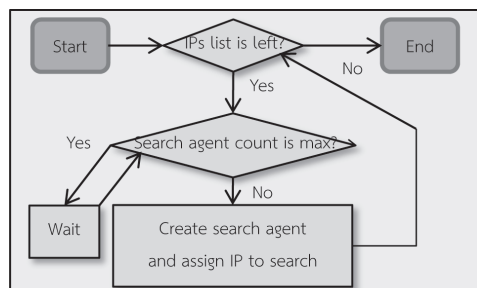
(ขั้นตอนการหาความหมายของคำที่สื่อถึง WSDL นั้นมากที่สุด)

- ทำซ้ำแต่ละ $Word_i$
 - ให้ $Synset_j$ คือ ชุดของคำพ้อง ความหมายของคำสำคัญที่จะสืบจาก WordNet โดยที่ j คือ ตัวเลขจำนวนเต็มที่เป็นลำดับของ $Synset_j$
 - ให้ $SynsetScore_j$ คือ คะแนนของ $Synset_j$ ที่จะสืบค้น โดยที่ j คือ ตัวเลขจำนวนเต็มที่เป็นลำดับของ $Synset_j$
 - สืบค้นข้อมูล Synsets ของคำสำคัญจากฐานข้อมูล WordNet เก็บไว้ที่ $Synset_j$
- ทำซ้ำแต่ละ $Synset_j$
 - ให้ $WordDefinition$ คือ ความหมายของ $Synset_j$
 - สืบค้นคำพ้องใน $Synset_j$ จากฐานข้อมูล WordNet
 - นับจำนวนคำพ้องที่ปรากฏใน $Word_i$ เก็บไว้ที่ $SynsetScore_j$
 - นับจำนวน $Word_i$ ที่ปรากฏใน $WordDefinition$ เพิ่มใน $SynsetScore_j$
- เก็บ $Synset_j$ ที่มี $SynsetScore_j$ มากที่สุดไปยัง $WordSynset_i$

(ขั้นตอนการเพิ่มคะแนนจากคำที่มีความหมายเดิม)

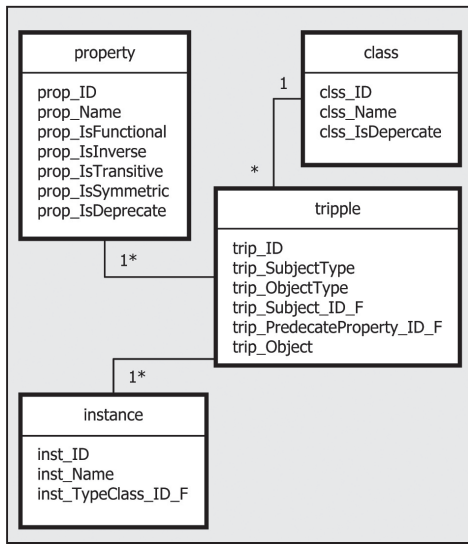
- ทำซ้ำแต่ละ $WordSynset_i$
 - ให้ $WordForm_j$ คือ ชุดคำสั้งพ้อง ความหมายของ $WordSynset_i$ โดยที่ j คือ ตัวเลขจำนวนเต็มที่เป็นลำดับของ $WordForm_j$
 - หากมีคำใน $WordForm_j$ ปรากฏใน $Word_i$ ที่มี $WordScore_i$ น้อยกว่าออกและนำ $WordScore_i$ ของคำ ที่น้อยกว่าคูณกับน้ำหนักแบบ Synonym แล้วเพิ่มใน $WordScore_i$ ของคำว่าที่ $WordScore_i$ มากกว่า
 - ลบ $WordScore_i$ และ $WordSynset_i$ ของ $Word_i$ ที่ลบออกในข้อที่แล้วออก

2) โมดูล Service Search Agent Creator มีหน้าที่ในการสร้าง Search Agent ขึ้นตามจำนวนสูงสุดที่กำหนดไว้ แล้วกำหนด IP จากช่วง IP ที่จะค้นหาให้ Search Agent ทีละ 1 IP เป็น URL เริ่มต้นให้ Serch Agent ค้นหา หาก Search Agent ตัวใดหยุดการทำงาน Service Search Agent Creator จะสร้าง Search Agent ใหม่ขึ้นมาแทนที่ ตามจำนวนสูงสุด หากรายการ IP ถูกค้นหาหมดแล้ว และ Search Agent จบการทำงานทุกตัวแล้ว Service Search Agent Creator จะส่งต่อรายการ WSDL URL กลับไปให้ Service Management แล้วหยุดการทำงานตัวเองลง ดังแสดงในภาพที่ 5

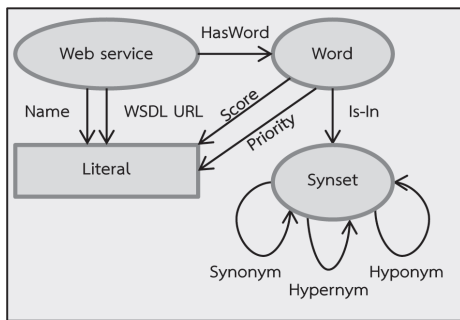


ภาพที่ 5 ขั้นตอนการสร้าง Search Agent

3) ส่วน Service Database มีการออกแบบฐานข้อมูลเชิงสัมพันธ์เพื่อให้รองรับความสัมพันธ์ของออนโทโลยีของเว็บเซอร์วิส และคำสำคัญที่ผ่านการประมวลผลแล้ว และออกแบบมาให้รองรับออนโทโลยีทุกรูปแบบ เพื่อง่ายต่อการพัฒนาต่อไปในภายหลัง ดังแสดงในภาพที่ 6 และได้ออกแบบออนโทโลยี แสดงความสัมพันธ์ของ class Web service, Word และ Synset ไว้ดังแสดงในภาพที่ 7



ภาพที่ 6 ฐานข้อมูลเชิงสัมพันธ์เพื่อรองรับออนโทโลยี



ภาพที่ 7 ออนโทโลยีแสดงความสัมพันธ์ของเว็บเซอร์วิสและคำสำคัญ

โดยจะมีข้อมูลออนโทโลยีหลักของ class Web service, Word และ Synset รวมทั้ง property แสดงความสัมพันธ์ของ class เก็บไว้ที่ตาราง property, class และ tripple ตั้งแต่เริ่มต้นระบบ ส่วนของข้อมูลที่สืบค้นและผ่านการประมวลผลแล้วจะถูกเก็บในตาราง instance และ triple

4) ฐานข้อมูล WordNet ส่วนนี้เป็นฐานข้อมูลคำศัพท์ภาษาอังกฤษที่ประกอบด้วยออนโทโลยีของความสัมพันธ์ของ คำศัพท์ รวมทั้งยังมีความหมายของคำศัพท์ ซึ่งจะมีชุดของคำศัพท์ที่พ้องความหมายกัน โดยส่วนนี้จะเป็นส่วนเสริมให้การทำงานของระบบ

ดึงข้อมูลคำศัพท์ต่างๆ ไปประกอบการประมวลผลหาความหมายและจัดกลุ่มในงานวิจัยนี้ได้ใช้ JAWS API [7] ที่มีพื้นฐานบนภาษา Java ในการเชื่อมต่อกับฐานข้อมูล WordNet

5) ส่วนเชื่อมต่อกับผู้ใช้เว็บเซอร์วิสที่เข้ามาค้นหาบริการ เป็นการค้นหาบริการเว็บเซอร์วิส จากคำที่ต้องการ โดยจะค้นหาจากคำที่ synonym, hypernym, hyponym ที่ต่างกัน ไม่เกิน 5 ลำดับชั้น ทั้งหมดนำมาหาค่าคะแนนจากสูตรที่ (1) ดังนี้

$$SCORE = WEIGHT \times PRIORITY_SCORE \quad (1)$$

โดยที่ SCORE = ค่าคะแนนของเว็บเซอร์วิส

WEIGHT = ค่าน้ำหนักแต่ละลำดับชั้น

PRIORITY_SCORE = คะแนนความสำคัญ

ของคำ

โดยคิดคะแนนความสำคัญของคำจากสูตรที่

(2) ดังนี้

$$PRIORITY_SCORE = 6 - WORD_PPRIORITY$$

(2)

โดยที่ WORD_PPRIORITY = ความสำคัญของ

คำและกำหนดค่าน้ำหนักแต่ละลำดับชั้นดังตารางที่ 1

ตารางที่ 1 ค่าน้ำหนักแต่ละลำดับชั้น [8]

ประเภทความสัมพันธ์	ค่าน้ำหนัก
Repetition	1.0
Synonym	0.9
Hyperym, Hyponym (Lever 1)	0.6
Hyperym, Hyponym (Lever 2-3)	0.2
Hyperym, Hyponym (Lever 4-5)	0.1

หากมีเว็บเซอร์วิสปรากฏซ้ำให้นำค่าคะแนนมารวมกันเป็นคะแนนของเว็บเซอร์วิสนั้น แล้วเรียงรายการเว็บเซอร์วิสที่ได้ตามลำดับค่าคะแนนจากมากไปหาน้อย ซึ่งรายการเว็บเซอร์วิสนี้จะเป็นผลการค้นหาที่จะนำไปแสดงให้ผู้ใช้บริการเว็บเซอร์วิส

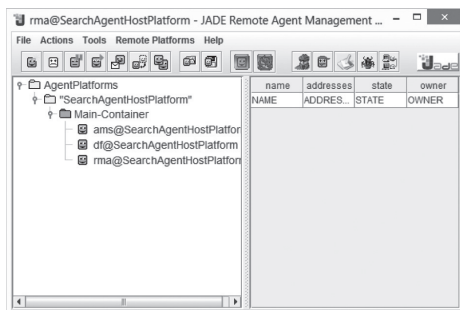
3. ผลการทดลอง

จากการทดลองสร้าง Service Search Agent Creator และ Search Agent โดยใช้ภาษา Java และใช้ JADE API [9] ซึ่งมีพื้นฐานบนภาษา Java เป็นตัวช่วยในการสร้าง Agent ซึ่งได้ผลการทดลองดังนี้

1) รัน JADE CONTAINER จาก Command Line ดังนี้

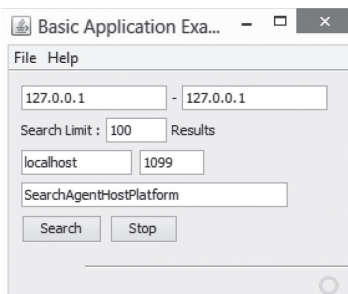
```
java -cp jade.jar jade.Boot -gui -deregister ture
platform-id SearchAgentHostPlatform
```

ซึ่งจะได้ GUI ของ JADE Container ดังแสดงในภาพที่ 8



ภาพที่ 8 GUI ของ JADE CONTAINER

2) เริ่มโปรแกรมจัดการ Agent ดังแสดงในภาพที่ 9



ภาพที่ 9 โปรแกรมจัดการ Agent

จากภาพที่ 9 โปรแกรมมีรายละเอียดตามลำดับดังนี้

- ช่วงของ IP ที่จะให้ Agent เข้าไปสืบค้นข้อมูล
- ผลลัพธ์สูงสุดที่ต้องการ

- Server ที่ JADE container ทำงานอยู่
- Port ของ JADE container ซึ่งโดยปกติจะเป็น 1099

- ชื่อของ JADE container ที่ตั้งไว้ในขั้นตอนเริ่มการทำงาน JADE Container

จากนั้นจึงกดปุ่ม Search เพื่อเริ่มการทำงานของ Search Agent Creator แล้ว Search Agent Creator จะสร้าง Search Agent ขึ้นมาตามเงื่อนไขที่กำหนด และเข้าไปสืบค้นข้อมูลตามช่วง IP ที่ระบุโดยการทดลองนี้จะจำลองแหล่งข้อมูล WSDL ให้เป็นเครื่องเดียวกัน

ข้อมูลที่ใช้ทดลองเป็น WSDL ของเว็บเซอร์วิสการพยากรณ์อากาศจากเว็บ CDYNE [10] จากการที่ Search Agent ได้ส่งลิงค์ WSDL คืนมาแล้ว Service Management ได้สกัดเอาข้อมูลที่จะประมวลผลซึ่งประกอบด้วยข้อมูลจากแท็กและแอตทริบิวต์ของ WSDL ดังนี้

```
<documentation>value</documentation>
<operation name="value"></operation>
<service name="value"></service>
```

ตัวอย่างข้อมูลสำคัญที่จะนำมาประมวลผลมีดังนี้

```
Gets Information for each
WeatherDAllows you to get your City
Forecast Over the Next 7 Days, which
is updated hourly.
U.S. OnlyAllows you to get your City's
U.S. Only
GetWeatherInformation
GetCityForecastByZIP
GetCityWeatherByZIP
```

เมื่อข้อมูลข้างต้นผ่านการประมวลผลโดย Service Management แล้ว จะได้คำสำคัญของการคำอธิบายเว็บเซอร์วิสการพยากรณ์อากาศดังแสดงในตารางที่ 2

ตารางที่ 2 ผลการประมวลผลหาคำสำคัญ

Priority	Synset ID	Word	Score
1	11358426	weather	4.0
2	6659152	forecast	2.0
3	6546125	information	2.0
4	8406385	city	2.0
5	13559365	zip	2.0
6	15009123	day	1.0

ข้อมูลที่ได้จะถูกบันทึกลงฐานข้อมูล Service Database ซึ่งใช้ MySQL [11] เป็นตัวจัดการฐานข้อมูล แล้วเปิดให้ผู้ใช้เว็บเซอร์วิสเข้ามาสืบค้นได้ ดังตัวอย่าง การสืบค้นเว็บเซอร์วิสผ่าน Service Search Engine

ตารางที่ 3 การสืบค้นเว็บเซอร์วิสโดยใช้คำค้น email mail address

No.	Webservice Name	Score
1	DOTSEmailValidation2	7.000
2	EmailVerNoTestEmail	5.750
3	EmailVerify	5.571
4	ChatService	5.000
5	NorthAmericanAddressVerificationService	5.000

การวัดประสิทธิภาพการสืบค้นเว็บเซอร์วิสเชิงความหมาย จะพิจารณาจากระดับความแม่นยำ (Precision) และครบถ้วน (Recall) ของการสืบค้นโดยจะวัดความถูกต้องของผลการค้นหา 5 อันดับแรกว่าถูกต้องตามคำค้นหรือไม่ การวัดประสิทธิภาพการสืบค้นสามารถคำนวณได้จากสมการดังนี้

$$F - \text{measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

$$\text{Precision} = \frac{A}{A + B} \times 100$$

$$\text{Recall} = \frac{A}{A + C} \times 100$$

โดย Precision คือค่าความถูกต้อง

Recall คือค่าความครบถ้วน

A คือจำนวนเอกสารที่ถูกต้องและถูกสืบค้น

B คือจำนวนเอกสารที่ไม่ถูกต้องและถูกสืบค้น

C คือจำนวนเอกสารที่ถูกต้องแต่ไม่ถูกสืบค้น

ตารางที่ 4 ผลการสืบค้นหาเว็บเซอร์วิสเชิงความหมาย

ลำดับ	คำค้น	A	B	C
1	Email mail address	3	2	0
2	phone number	3	2	0
3	airport flight travel	4	1	0
4	hotel booking suite	3	2	0
5	money rate tax	4	1	0

ตารางที่ 5 ผลการวัดประสิทธิภาพ

ลำดับ	คำค้น	Precision	recall	FMeasure
1	Email mail address	60%	100%	75%
2	phone number	60%	100%	75%
3	airport flight travel	80%	100%	88%
4	hotel booking suite	60%	100%	75%
5	money rate tax	80%	100%	88%

4. สรุป

การวัดประสิทธิภาพการสืบค้นเว็บเซอร์วิสเชิงความหมายจากกลุ่มคำค้นผลลัพธ์ที่ได้ถือว่าค่อนข้างมีประสิทธิภาพดี เพราะระบบสามารถทำงานและประมวลผลนำเว็บเซอร์วิสที่ตรงกับคำค้นมาแสดงและสามารถนำเว็บเซอร์วิสอื่นที่เกี่ยวข้องกับคำค้นมาแสดงได้ตรงกับวัตถุประสงค์และสมมติฐานที่ตั้งไว้

เอกสารอ้างอิง

- [1] Tim Berners-Lee.(2010). Web Services [Online]. Available: <http://www.w3.org/DesignIssues/WebServices.html>
- [2] W3Schools. (2013).WSDL Tutorial [Online]. Available: <http://www.w3schools.com/wsd/default.asp>
- [3] Wikipedia. (2013). Software agent [Online]. Available: http://en.wikipedia.org/wiki/Software_agent
- [4] George A. Miller. (2012). WordNet A lexical database for English [Online]. Available: <http://wordnet.princeton.edu>
- [5] W3Schools. (2013).RDF Tutorial [Online]. Available: <http://www.w3schools.com/rdf/default.asp>
- [6] Wikipedia. (2013). Mobile Agent [Online]. Available: http://en.wikipedia.org/wiki/Mobile_agent
- [7] Brett Spell. (2009). Java API for WordNet Searching(JAWS)[Online]. Available: <http://lyle.smu.edu/~tspell/jaws>
- [8] ศรีสุตา พิละมาตย์. (2550). การจัดกลุ่มเอกสารบนเว็บ เพื่อการสืบค้นเชิงความหมาย. วิทยานิพนธ์ปริญญา วิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น.
- [9] Telecom Italia. (2013). Java Agent DEvelopment Framework [Online]. Available: <http://jade.tilab.com>
- [10] CDYNE. Weather [Online]. Available: <http://wsf.cdyne.com/WeatherWS/Weather.asmx>
- [11] Oracle Corporation and/or its affiliates.(2013). MySQL Open source database[Online]. Available: <http://www.mysql.com/>
- [12] sampan. WordNet [Online]. Available: <http://sampaan.wordpress.com/wordnet/>
- [13] ธรรมรัตน์ บุญรอด. (2550). แบบจำลอง การสืบค้นข้อมูลสารสนเทศโดยซอฟต์แวร์เอเจนต์. การศึกษาอิสระปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น.
- [14] Viriya Kaewmarin, Ngamnij Arch-int, Somjit Arch-int. (2008). Semantic Web Service Discovery and Integration using Service Search Crawler. Computational Intelligence for Modelling Control &Automatio. 10-12 Dec. 2008, KhonKaen University, KhonKaen , Thailand.
- [15] Membrane. (2010). Directory of SOAP Web Services [Online]. Available: <http://www.service-repository.com>
- [16] xmethods.Emerging web services standards [Online].Available: <http://www.xmethods.net/ve2/index.po>
- [17] สุพจน์ บัวเลิง. (2555). การแยกกลุ่มปัญหา โดยใช้ เว็บเชิงความหมาย และการทำเหมืองข้อมูล. การศึกษาอิสระปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น