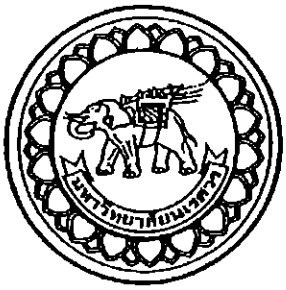


ระบบการค้นหาเว็บเซอร์วิส

WEB SERVICES SEARCH ENGINE

นางสาวกฤษณา ไชยยา รหัส 50364942

นายศราวุฒิ โยชนะ รหัส 50365307

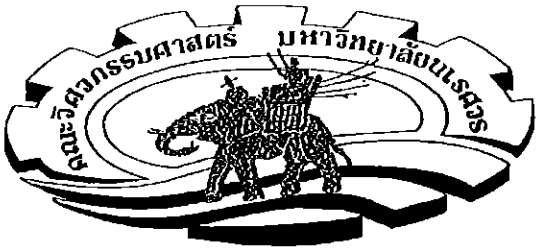
ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 17 พ.ย. 2554
เลขทะเบียน 1570957
เลขเรียกหนังสือ ว/ร.
มหาวิทยาลัยนเรศวร ว2825
2553

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2553



ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน ระบบการค้นหาเว็บเซอร์วิส

ผู้ดำเนินโครงงาน นางสาวกฤษณา ไชยยา รหัส 50364942

นายศราวุฒิ โยชนะ รหัส 50365307

อาจารย์ที่ปรึกษา ดร.วรลักษณ์ คงเด่นฟ้า

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2553

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนห**นึ่งของ**การศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ดร.วรลักษณ์ คงเด่นฟ้า)

.....กรรมการ

(นายเศรษฐา ตั้งค้าวานิช)

.....กรรมการ

(นายภาณุพงศ์ สอนคม)

หัวข้อโครงงาน	ระบบการค้นหาเว็บเซอร์วิส		
ผู้ดำเนินโครงงาน	นางสาวกฤษณา	ไชยยา	รหัส 50364942
	นายศราวุฒิ	โยชนะ	รหัส 50365307
อาจารย์ที่ปรึกษา	ดร.วรลักษณ์	คงเค่นฟ้า	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2553		

.....

บทคัดย่อ

เว็บเซอร์วิสได้รับความสนใจและเป็นที่นิยมในการพัฒนามากในปัจจุบัน เนื่องจากสามารถลดข้อจำกัดในการสื่อสารและแลกเปลี่ยนข้อมูลบนเทคโนโลยีที่แตกต่างกันได้ สามารถเรียกใช้ API ได้อย่างหลากหลาย ในทางธุรกิจ เว็บเซอร์วิสสร้างการค้าและบริการที่เป็นไปอย่างอัตโนมัติ มีการแลกเปลี่ยนข้อมูลผ่านทางเว็บเซอร์วิส ลดต้นทุนในการพัฒนาระบบและมีความคล่องตัว การพัฒนาจึงเป็นไปอย่างต่อเนื่อง การใช้งานเว็บเซอร์วิสโดยปกติแล้ว มักค้นหาข้อมูลใน Repository อาทิเช่น UDDI แต่ UDDI เป็นเพียงระบบลงทะเบียนขององค์กรผู้ให้บริการ ข้อมูลที่มีจำนวนมาก การค้นหาเว็บเซอร์วิสดังกล่าว จึงยุ่งยากและไม่สะดวกพอสำหรับการค้นหา ผู้จัดทำจึงเล็งเห็นว่าควรมีการพัฒนา "ระบบการค้นหาเว็บเซอร์วิส" เพื่อใช้เป็นเทคโนโลยีทดแทนสำหรับการค้นหาเพิ่มความสะดวกและลดระยะเวลาในการค้นหาจากข้อมูลของผู้ให้บริการที่มีจำนวนมากใน UDDI

ระบบการค้นหาเว็บเซอร์วิสนี้รองรับการทำงานในส่วนของ Searching และ Ranking โดยอาศัยหลักการของออนโทโลจีและแท็กเข้ามาช่วยในการค้นหา เป็นการกำหนดรายละเอียดบริการต่างๆด้วยแท็ก โดยแท็กนั้นได้มาจาก NAICS Ontology และแท็กที่กำหนดเองโดยผู้ใช้ แล้วทำการเปรียบเทียบแท็กกับ Boolean Expression ที่ได้จากการวิเคราะห์คำค้นหา เพื่อค้นหาบริการที่ต้องการ มีจัดลำดับผลการค้นหาเพื่อช่วยจัดเรียงข้อมูลที่มีความสำคัญและตรงความต้องการ การพัฒนาโครงงานนี้ สามารถตอบสนองในการช่วยสำหรับการค้นหาเว็บเซอร์วิส สำหรับผู้พัฒนาเว็บเซอร์วิสในด้านต่างๆที่ต้องการ สืบค้นได้ง่าย มีความสะดวกและรวดเร็ว

Project Title	WEB SERVICES SEARCH ENGINE	
Name	Miss.Kridsana Chaiya	ID. 50364942
	Mr.Sarawoot Yochana	ID. 50365307
Project Advisor	Woralak Kongdenfha, Ph.D.	
Major	Computer Engineering.	
Department	Electrical and Computer Engineering.	
Academic Year	2010	

.....

ABSTRACT

Today, the internet has evolved from a network of static web pages into a dynamic and interactive environment where data are instantly processed and transferred. This evolution has been brought with the introduction of web services that enables the development of data and applications in a distributed, but yet integrated environment. However, this success comes with a pain of developers who need to seek for appropriate web services scatter all over the internet. While the technology like the Universal Description, Discovery and Integration (UDDI), was invented with intentions to automate service discovery and integrations, it was unsuccessful. The reason is that UDDI contains inaccurate service information, hence it is not useful for service searches. One may argue that enormous amount of search engines available on the internet can be used for service searches. Unfortunately, results returned by web search engines are often mixed up web services with static web pages. This requires efforts from developers to separate web services from those static web pages. In this thesis, we therefore propose a system that is specialized for web service searches.

Among search methods employed by web search engines, keyword-based search is the most successful one. But we have found that keywords alone are not adequate for finding useful for finding web services as the search keywords may be limited or not accurate enough to represent what users need. We therefore adopt the concept-based search method for finding web services, i.e., the search keywords are extended with concepts from WordNet and a set of domain specific ontologies. We particularly extend the search keywords through four concepts: synonyms, generalization, specialization and any other types of associations. The search keywords as well as their extensions are then used to form a Boolean expression as an input to

search for related web services. By this way, the possibilities of web services to be matched are increased based on such extensions. In addition, those different concepts used to generate search keywords are also used as weighting criteria for ranking the search results. As a proof of concept, we have developed a web service search engine using the .NET platform. To validate our developed system, we have adopted a service repository that enables service providers to tag their services once published them in such a repository. These service tags are matched against the generated Boolean expression. The result shows that the web services searches are improved based on the ontology-based extensions generated for the search keywords.

กิตติกรรมประกาศ

ขอขอบคุณ ดร.วรลักษณ์ ดงเด่นฟ้า สำหรับคำแนะนำในการทำงาน การวางแผนระบบ รวมถึงการออกแบบพัฒนาโปรแกรมระบบการค้นหาเว็บเซอร์วิสนี้ ช่วยในการให้คำปรึกษาและคำแนะนำในทุกขั้นตอนการทำงานของการจัดทำโครงงาน รวมถึงเสียสละเวลาอย่างมากเพื่อช่วยตรวจสอบและเสนอแนะข้อบกพร่องที่ควรแก้ไขของระบบในขั้นตอนต่าง ๆ

รวมถึง ดร.สุรเดช จิตประไพกุลศาลและ ดร.พงศ์พันธ์ กิจสนาโยธิน ที่ช่วยในการให้คำแนะนำในการพัฒนาระบบการค้นหา อาจารย์ภาณุพงศ์ สอนคมและ อาจารย์เศรษฐา ตั้งค้าวานิช ที่ให้ข้อเสนอแนะและตรวจสอบความถูกต้องของระบบในการปรับปรุงโครงงานให้ดียิ่งขึ้น จนทำให้การจัดทำโครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

ทั้งนี้ ขอขอบพระคุณครอบครัวที่คอยให้กำลังใจ เป็นหนึ่งในเบื้องหลังของความสำเร็จในการพัฒนาโครงงานนี้ เพื่อนทุกคนที่คอยให้การช่วยเหลือในทำงานและให้คำปรึกษาในการเขียนโปรแกรม เป็นกำลังใจและให้ความช่วยเหลือในทุกด้าน ผู้พัฒนาโครงงานจึงขอขอบพระคุณทุกท่านไว้ ณ โอกาสนี้

กฤษณา ไชยยา

ศราวุฒิ โยชนะ

สารบัญ

หน้า

บทคัดย่อภาษาไทย .....ก

บทคัดย่อภาษาอังกฤษ .....ข

กิตติกรรมประกาศ .....ง

สารบัญ .....จ

สารบัญตาราง .....ญ

สารบัญรูป .....ฎ

บทที่ 1 บทนำ.....1

1.1 ที่มาและความสำคัญ 1

1.2 วัตถุประสงค์ของโครงงาน 2

1.3 ขอบเขตของโครงงาน 2

1.4 ขั้นตอนการดำเนินงาน 3

1.5 ประโยชน์ที่คาดจะได้รับ.....4

1.6 งบประมาณที่ใช้..... 4

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....5

2.1 เว็บเซอร์วิส(Web Service).....5

2.1.1 มาตรฐานหลักของ Web Service 5

2.1.1.1 SOAP (Simple Object Access Protocol).....5

2.1.1.2 WSDL (Web Services Description Language).....6

2.1.1.3 UDDI (Universal Description, Discovery and Integration).....6

2.1.2 ความแตกต่างของ UDDI กับระบบการค้นหาเว็บเซอร์วิส 6

2.2 เอ็กซ์เอ็มแอล (XML) 7

2.2.1 องค์ประกอบของ XML.....7

2.2.2 ข้อดีของ XML.....8

2.3 ระบบการค้นหา (Search Engine).....9

2.2.1 ประเภทของระบบการค้นหา (Search Engine).....9

2.3.1.1 แบบอาศัยการจัดเก็บข้อมูลเป็นหลัก.....9

สารบัญ (ต่อ)

หน้า

2.3.1.2 แบบอ้างอิงใน "ชุดคำสั่งเมตะ" (Meta Search Engine).....9

2.3.2 ประโยชน์ของระบบการค้นหา (Search Engine).....9

2.3.3 การทำงานของระบบการค้นหา (Search Engine).....10

2.3.4 ตัวอย่างเว็บไซต์ที่ให้บริการระบบค้นหา (Search Engine).....11

2.3.5 เทคนิคการค้นหาข้อมูลโดยใช้บูลีนโอเปอร์เรเตอร์.....12

2.3.5.1 การใช้ AND 12

2.3.5.2 การใช้ OR 12

2.3.5.3 การใช้ NOT 12

2.3.6 การค้นหาข้อมูลด้วยวิธีอื่นๆ.....13

2.3.6.1 การระบุระยะห่างระหว่างคำค้น.....13

2.3.6.2 การตัดคำ (Truncation).....13

2.4 ครอว์เลอร์ (Crawler).....14

2.4.1 เว็บครอว์เลอร์ (Web Crawler).....14

2.4.2 องค์ประกอบของเว็บครอว์เลอร์.....14

2.4.2.1 ฐานข้อมูล.....14

2.4.2.2 ซอฟแวร์.....14

2.4.3 โฟกัสครอว์เลอร์ (Focused Crawler) 15

2.4.4 TF.IDF 15

2.4.5 Vector Space Model.....16

2.4.6 โครงสร้างของโฟกัสครอว์เลอร์.....17

2.4.6.1 Focused Crawler working process.....17

2.4.6.2 Seed pages fetching sub-system.....17

2.4.6.3 Topic keywords generating sub-system.....17

2.4.6.4 Similarity Computing Engine 18

2.5 ฐานข้อมูล (Database) 18

2.5.1 ระบบจัดการฐานข้อมูล.....18

2.5.2 การออกแบบฐานข้อมูล.....19

2.5.2.1 รูปแบบข้อมูลแบบลำดับขั้นหรือโครงสร้างแบบลำดับขั้น.....19

สารบัญ (ต่อ)

หน้า

2.5.2.2 รูปแบบข้อมูลแบบเครือข่าย (Network data Model).....19

2.5.2.3 รูปแบบความสัมพันธ์ข้อมูล (Relation data model).....19

2.5.3 การออกแบบฐานข้อมูลเชิงสัมพันธ์.....20

2.5.4 การออกแบบฐานข้อมูลในระดับตรรกะ.....21

2.6 ภาษาสืบค้น (Query Language).....21

2.6.1 ภาษาสืบค้นข้อมูลเชิงโครงสร้าง (Structure Query Language: SQL).....21

2.6.2 เอกซ์พาท (XPATH).....22

2.6.3 เอกซ์คิวรี (XQuery).....22

2.6.4 Language Integrated Query (LINQ).....22

2.6.4.1 LINQ to Object.....23

2.6.4.2 LINQ to SQL.....23

2.6.4.3 LINQ to XML.....23

2.7 Ontology 23

2.7.1 RDF.....25

2.7.2 NAICS.....25

2.7.3 OWL.....26

บทที่ 3 ออกแบบระบบ.....27

3.1 การออกแบบระบบ (System Design).....27

3.1.1 การทำงานของระบบ (Conceptual Design).....28

3.1.2 โครงสร้างของโปรแกรม (Program Structure).....29

3.1.3 Interface Chart.....30

3.1.4 Use Case Diagrams.....31

3.1.5 Use Case Descriptions.....32

3.1.6 Class Diagram.....33

3.1.7 Activity Diagram.....34

3.1.8 Sequence Diagram.....35

สารบัญ (ต่อ)

หน้า

3.2 ข้อมูล (Data).....36

3.2.1 ฐานข้อมูลของระบบฐานข้อมูลแท็ก (Tagging System).....36

3.2.1.1 Database Design.....36

3.2.1.2 ER Diagram.....37

3.2.2 ฐานข้อมูลของระบบการแสดงผล (Interface System).....38

บทที่ 4 การพัฒนาและใช้งานระบบ.....40

4.1 Hardware และ Software Requirement.....40

4.1.1 Hardware Requirement.....40

4.1.2 Software Requirement.....40

4.2 อินพุต เอาต์พุต และเซ็ตข้อมูล.....41

4.2.1 ข้อมูลอินพุต.....41

4.2.1.1 อินพุตจากระบบการแสดงผล (Interface System).....41

4.2.1.2 อินพุตจากระบบการฐานข้อมูลแท็ก (Tagging System).....41

4.2.2 ข้อมูลเอาต์พุต.....42

4.2.1.1 เอาต์พุตที่ส่งให้แก่ระบบการฐานข้อมูลแท็ก (Tagging System).....42

4.2.1.2 เอาต์พุตที่ส่งให้แก่ระบบการแสดงผล (Interface System).....42

4.2.3 เซ็ตข้อมูล.....43

4.3 ระบบการค้นหา (Search System).....44

4.3.1 ประโยคสืบค้น Boolean Expression.....44

4.3.2 การค้นหาข้อมูลโดยการเปรียบเทียบแท็กข้อมูล.....46

4.3.3 การทำงานสื่อสารของเว็บเซอร์วิสในระบบการค้นหาเว็บเซอร์วิส..... 48

4.3.4 Pseudo code ในการทำงานของระบบค้นหาเว็บเซอร์วิส.....50

4.4 ระบบจัดลำดับการค้นหา (Ranking System).....53

4.4.1 การจัดลำดับผลของการค้นหา (Ranking).....53

4.4.2 Similarity Thought Document Hierarchy.....55

4.4.3 Pseudo code ของการทำงานของระบบ Ranking 56

สารบัญ (ต่อ)

หน้า

บทที่ 5 ผลการทดลอง.....59

5.1 ระบบค้นหา (Search System).....59

5.1.1 วิธีการทดสอบระบบการค้นหาเว็บเซอร์วิส.....59

5.1.1.1 การวัดประสิทธิภาพการทำงานด้านความถูกต้องของข้อมูลในการค้นหา.....59

5.1.1.1 การวัดประสิทธิภาพการทำงานด้านความเร็วในการค้นหา.....60

5.1.2 ผลการทดสอบ.....60

5.1.3 สรุปผลการทดสอบ.....61

บทที่ 6 บทสรุป.....63

6.1 สรุปผลการดำเนินการ.....63

6.2 ปัญหาและอุปสรรคในการพัฒนา.....63

6.3 ข้อเสนอแนะ.....64

6.4 แนวทางในการพัฒนาต่อในอนาคต.....64

เอกสารอ้างอิง.....65

ประวัติผู้เขียนโครงงาน.....66

สารบัญตาราง

ตารางที่		หน้า
1.1	ขั้นตอนและแผนการดำเนินงาน	3
3.1	ข้อมูล Interface Chart	30
3.2	Use Case Description ของผู้ใช้	32
3.3	Use Case Description ของ Machine	32
4.1	แสดงรายการของ Hardware Requirement	40
4.2	แสดงรายการของ Software Requirement	40
4.3	สัญลักษณ์เงื่อนไข Boolean Expression	44
4.4	ความหมายและการใช้งานเงื่อนไขของ Boolean Expression	45
4.5	ค่าน้ำหนักที่ใช้พิจารณาตามเงื่อนไขของ Boolean Expression ของแต่ละข้อมูล	53
5.1	ผลการทดสอบการวัดประสิทธิภาพการทำงานด้านความถูกต้องในการค้นหา	60
5.2	ผลการทดสอบประสิทธิภาพการทำงานด้านความเร็วในการค้นหา	61

สารบัญรูป

รูปที่		หน้า
2.1	องค์ประกอบของ SOAP	5
2.2	DynamicsCommunication	6
2.3	UDDI ใน Web Services	7
2.4	ลักษณะการทำงานของเว็บครอว์เลอร์	14
2.5	TF.IDF Weighting Schemes	15
2.6	Cosine Product	16
2.7	Cosine Similarity Measure	16
2.8	กระบวนการทำงานของ Focused Crawler	17
2.9	แสดงถึงส่วนประกอบของ LINQ	23
3.1	แสดงการทำงานของระบบการค้นหาเว็บเซอร์วิส	27
3.2	แสดง Conceptual Design	28
3.3	Program Structure	30
3.4	Use Case Diagram ของระบบ	31
3.5	Class Diagram ของระบบ	33
3.6	Activity Diagram ของระบบ	34
3.7	Sequence Diagram ของระบบ	35
3.8	รูปแสดงตารางของแท็กในฐานข้อมูลระบบ Tagging System	36
3.9	รูปแสดงตารางและความสัมพันธ์ของแท็กและบริการในฐานข้อมูล	37
3.10	โครงสร้าง ER – DIAGRAM ของฐานข้อมูลแท็ก	38
3.11	Search Interface	39
4.1	ไฟล์ XML อินพุตของ Boolean Expression ของระบบจากส่วนระบบการแสดงผล	41
4.2	ไฟล์ XML อินพุตของ URL ที่ได้จากระบบการฐานข้อมูลแท็ก	42
4.3	ไฟล์ XML เอาต์พุตของ URL ที่ได้จากระบบการฐานข้อมูลแท็ก	42
4.4	ไฟล์ XML เอาต์พุตของผลการค้นหาที่ส่งให้แก่ระบบการแสดงผล	43
4.5	เงื่อนไขของ Boolean Expression ในการดึงข้อมูล	44
4.6	หน้า Interface สำหรับการค้นหาเว็บเซอร์วิส	48
4.7	ตัวอย่างการค้นหาเว็บเซอร์วิส โดยการกรอกข้อมูลคำค้นหาและแสดงผลการค้นหา	48
4.8	ตัวอย่างการค้นหาข้อมูลที่ใช้ในระบบตามเงื่อนไขของ Boolean Expression	49

สารบัญรูป(ต่อ)

รูปที่		หน้า
4.9	ตัวอย่างระบบเว็บเซอร์วิสในการค้นหาข้อมูล แสดงเซอร์วิสที่ใช้ ผลการค้นหา และน้ำหนักของข้อมูลตามเงื่อนไขของ Boolean Expression	49
4.10	Pseudo code การ Query ข้อมูลจากฐานข้อมูล	50
4.11	ผลของข้อมูลที่ได้จากการ Query ข้อมูลจากฐานข้อมูล	50
4.12	Pseudo code การอ่านไฟล์อินพุต Boolean Expression	51
4.13	Pseudo code การตรวจสอบข้อมูลที่ซ้ำกัน	51
4.14	Pseudo code การตรวจสอบเงื่อนไข OR AND	52
4.15	รูปแสดงการทำงานของระบบ Searching และ Ranking	54
4.16	สูตรการหาค่า Similarity Hierarchy	55
4.17	Pseudo code การตรวจสอบเงื่อนไข OR AND	56
4.18	Pseudo code ในการหาค่าความสัมพันธ์ใน Similarity Hierarchy ของ Association	57
4.19	Pseudo code การเรียงลำดับข้อมูล	57
4.20	ผลการค้นหา URL ที่มีการจัดลำดับการค้นหาแล้ว	58
4.21	ตัวอย่างการค้นหาเว็บเซอร์วิส โดยการกรองข้อมูลคำค้นหาและแสดงผลการค้นหา	58
5.1	แบบสอบถามที่ใช้ในการวัดประสิทธิภาพการทำงานด้านความถูกต้องในการค้นหา	59
5.2	แผนภูมิผลการทดสอบการวัดประสิทธิภาพการทำงานด้านความถูกต้องในการค้นหา	60

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

เว็บเซอร์วิส เป็นแอพพลิเคชั่นที่ทำงานในลักษณะให้บริการแก่แอพพลิเคชั่นหรือโปรแกรมอื่นๆ ผ่านเว็บ โดยใช้ภาษา XML (The Extensible Markup Language) เป็นสื่อในการแลกเปลี่ยนข้อมูลระหว่างกัน ทำให้ผู้ใช้สามารถเรียกใช้บริการต่างๆของแอพพลิเคชั่น ที่อยู่บนแพลตฟอร์มใดๆก็ได้ โดยค้นหาเว็บเซอร์วิสได้โดยที่ไม่จำเป็นต้องรู้ที่อยู่จริงของ แอพพลิเคชั่นหรือโปรแกรมนั้นๆ ปัจจุบันเทคโนโลยีด้านนี้ได้รับความนิยมมากและมีแนวโน้มที่จะได้รับความสนใจด้านการใช้งานมากยิ่งขึ้นในอนาคต เนื่องจากสามารถลดข้อจำกัดการใช้งานการสื่อสารข้อมูลและแลกเปลี่ยนข้อมูลระหว่างกันบนเทคโนโลยีที่แตกต่างกันได้ ในทางธุรกิจเว็บเซอร์วิสสามารถสร้างให้เกิดพันธมิตรทางการค้าโดยการค้นหาของ UDDI มีการทำธุรกิจ การค้าและบริการที่เป็นไปอย่างอัตโนมัติ โดยการแลกเปลี่ยนข้อมูลผ่านทางเว็บเซอร์วิส สามารถลดต้นทุนในด้านพัฒนาระบบได้ดี สามารถตอบสนองความต้องการของลูกค้าให้มีความคล่องตัว ให้ใช้ประโยชน์จากแอพพลิเคชั่นต่าง ๆภายใต้ระบบงานที่แตกต่างกันได้อย่างรวดเร็วและคุ้มค่า รวมถึงสามารถพัฒนาช่องทางการเข้าถึงข้อมูลทางธุรกิจสอดคล้องกับผู้ใช้แต่ละกลุ่มโดยไม่จำกัดสถานที่

ในการเรียกใช้งานเว็บเซอร์วิสนั้น จำเป็นต้องรู้คุณลักษณะของเว็บเซอร์วิสจากเอกสาร WSDL (Web Services Description Language) มีการร้องขอบริการจากผู้ขอใช้บริการ การรอผลจากผู้ให้บริการ โดยใช้โพรโตคอล SOAP (Simple Object Access Protocol) และสิ่งที่สำคัญในการเรียกใช้งานเว็บเซอร์วิสคือ ต้องรู้ที่อยู่ของบริการนั้นๆ จาก UDDI (Universal Description, Discovery, and Integration) เพื่อสามารถค้นหาข้อมูลเว็บเซอร์วิสได้ตามต้องการ

UDDI (Universal Description, Discovery, and Integration) เป็นเทคโนโลยีหนึ่งของเว็บเซอร์วิส สำหรับให้ผู้ให้บริการทำการลงทะเบียนข้อมูลเว็บเซอร์วิสบน UDDI Registry จุดประสงค์ของ UDDI Registry นั้นเพื่อประกาศการให้บริการของตนเองสู่สาธารณะ ให้ผู้ใช้บริการทำการเลือกค้นหาข้อมูลและบริการที่มีทั้งหมดจากระบบนี้ UDDI Registry เป็นเพียงระบบลงทะเบียนขององค์กรผู้ให้บริการเท่านั้น งานด้านนี้มีข้อมูลจำนวนมาก การค้นหาเว็บเซอร์วิสดังกล่าว จึงมีความยุ่งยากและไม่สะดวกพอสำหรับการค้นหา เนื่องจากเป็นการค้นหาจากข้อมูลทั้งหมดที่มีจำนวนค่อนข้างมาก จึงมีการพัฒนา "ระบบการค้นหาเว็บเซอร์วิส" เพื่อใช้เป็นเทคโนโลยีทดแทนสำหรับการค้นหา เพิ่มความสะดวกและลดระยะเวลาในการค้นหาจากข้อมูลของผู้ให้บริการที่มีจำนวนมากใน UDDI Registry

ระบบการค้นหาในการค้นหาเว็บเซอร์วิสในปัจจุบัน สามารถลดข้อจำกัดทางด้านนี้ได้ แต่การให้บริการยังคงไม่เพียงพอเท่ากับความต้องการที่จะเพิ่มขึ้น และยังไม่สามารถค้นหาเว็บเซอร์วิสได้ตรงความต้องการมากเท่าที่ควร จึงมีการพัฒนาโครงงาน "ระบบการค้นหาเว็บเซอร์วิส" ขึ้น เพื่อสามารถทำการค้นหางานด้านนี้รวมถึงบริการต่างๆได้อย่างหลากหลาย รวดเร็วและตรงความต้องการ " ระบบการค้นหาเว็บเซอร์วิส " นี้ พัฒนาโดยใช้ Boolean Expression ซึ่งพิจารณาจากคำค้นหาของผู้ใช้ ทำการเปรียบเทียบกับแท็กที่ผู้ให้บริการเว็บเซอร์วิส (Provider) ที่กำหนดขึ้นโดยแท็กนั้นๆ ได้มาจาก NAICS Ontology และแท็กที่กำหนดเองโดยผู้ใช้ (User Defined Tag) และมีการแสดงผลข้อมูลบริการที่ทำการจัดอันดับผลของการค้นหา แสดงแก่ผู้ใช้เลือกบริการเว็บเซอร์วิส ให้มีความใกล้เคียงและตรงกับความต้องการ โดยสามารถให้ข้อมูลที่ดีกว่าการค้นหาจาก UDDI Register โดยตรง

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อสร้าง "ระบบการค้นหาเว็บเซอร์วิส" ให้สามารถค้นหาข้อมูลงานด้านเว็บเซอร์วิสตามที่ต้องการได้

1.2.2 เพื่อสร้าง "ระบบการค้นหาเว็บเซอร์วิส" ที่มีการรับอินพุตที่เป็น Boolean Expression ใช้ในการเปรียบเทียบกับแท็กที่ผู้ให้บริการเว็บเซอร์วิส (Service Provider) ที่กำหนดขึ้นโดยแท็กนั้นได้มาจาก NAICS Ontology และแท็กที่กำหนดเองโดยผู้ใช้ (User Defined Tag)

1.2.3 เพื่อสร้าง "ระบบการค้นหาเว็บเซอร์วิส" ที่มีการจัดอันดับการค้นหา (Ranking) ของผลการค้นหาเว็บเซอร์วิส ใน Ontology ที่เซอร์วิสนั้นๆ เกี่ยวข้องอยู่

1.3 ขอบเขตของโครงงาน

1.3.1 ระบบการค้นหาของเว็บเซอร์วิสที่สร้างขึ้นนั้น มีการรับข้อมูลเป็น Boolean Expression โดยไม่สนใจ Query Phase หรือประโยคการค้นหาใดๆ

1.3.2 ระบบการค้นหาของเว็บเซอร์วิสที่สร้างขึ้นนั้น มีการเปรียบเทียบ Boolean Expression กับแท็กที่ผู้ให้บริการเว็บเซอร์วิส (Provider) กำหนดขึ้นให้กับเว็บเซอร์วิส ซึ่งแท็กนั้นๆ ได้มาจาก NAICS Ontology และแท็กที่กำหนดเองโดยผู้ใช้ (User Defined Tag)

1.3.3 ระบบมีการจัดลำดับการค้นหา (Ranking) ของผลการค้นหาเว็บเซอร์วิส ใน Ontology ที่เซอร์วิสนั้นๆ เกี่ยวข้องอยู่

1.4 ขั้นตอนและแผนการดำเนินงาน

ขั้นตอนและแผนการดำเนินงานในการทำ "ระบบการค้นหาเว็บเซอร์วิส" ดังนี้

1) ศึกษาโครงสร้างและระบบการทำงานของเว็บเซอร์วิส (Web Services)

2) ศึกษาโครงสร้างและระบบการทำงานของระบบการค้นหา (Search Engine)

3) ศึกษาโครงสร้างและระบบการทำงานของ Ontology, XML, OWL และ NAICS

4) ศึกษาโครงสร้างและระบบการทำงานของระบบการค้นหาและการจัดลำดับผลของการค้นหา (Ranking)

5) วิเคราะห์และออกแบบระบบการค้นหาเว็บเซอร์วิส (Web Services Search Engine)

6) พัฒนาระบบการค้นหาเว็บเซอร์วิส (Web Services Search Engine)

7) ทดสอบและประเมินผลระบบการค้นหาเว็บเซอร์วิส (Web Services Search Engine)

8) สรุปผลการดำเนินโครงงาน และจัดทำรูปเล่มปริญญานิพนธ์

ตารางที่ 1.1 ขั้นตอนและแผนการดำเนินงาน

รายละเอียด	ปี 2553							ปี 2554		
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1) ศึกษาโครงสร้างและระบบการทำงานของเว็บเซอร์วิส										
2) ศึกษาโครงสร้างและระบบการทำงานของการค้นหา										
3) ศึกษาโครงสร้างและระบบการทำงานของ Ontology, XML, OWL, OWLs และ NAICS										
4) ศึกษาโครงสร้างและระบบการทำงานของระบบการค้นหาและการจัดลำดับผลการค้นหา (Ranking)										
5) วิเคราะห์และออกแบบระบบการค้นหาเว็บเซอร์วิส										
6) พัฒนาระบบการค้นหา										
7) ทดสอบและประเมินผลระบบ										
8) สรุปผลการดำเนินโครงงานและจัดทำรูปเล่มปริญญานิพนธ์										

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ประโยชน์ต่อผู้ทำการศึกษา

- ได้พัฒนาความรู้เทคโนโลยีด้านเว็บเซอร์วิส (Web Services)
- ได้พัฒนาความรู้เทคโนโลยีด้านระบบการค้น (Search Engine)
- ได้พัฒนาความรู้เทคโนโลยีของ Ontology, XML, OWL, OWLs และ NAICS
- ได้พัฒนาความรู้เทคโนโลยีของการจัดลำดับผลของการค้นหา (Ranking) และระบบฐานข้อมูล (Database System)
- ได้เพิ่มความรู้ความเข้าใจในวิธีการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming)

1.5.2 ประโยชน์ต่อผู้ใช้งาน

ได้ระบบการค้นหาเว็บเซอร์วิสที่สามารถค้นหาข้อมูลด้านเว็บเซอร์วิสได้ตามต้องการของผู้พัฒนางานด้านเว็บเซอร์วิส(Web Services Developer) ซึ่งมีการการค้นหาเว็บเซอร์วิส โดยการเปรียบเทียบ Boolean Expression กับแท็กที่ผู้ให้บริการเว็บเซอร์วิส (Service Provider) กำหนดขึ้นจาก NAICS และแท็กที่กำหนดเองโดยผู้ใช้ (User Defined Tag) มีการจัดอันดับผลของการค้นหา (Ranking) ของผลการค้นหาบริการของเว็บเซอร์วิส ใน Ontology ที่เซอร์วิสนั้นๆ เกี่ยวข้องอยู่แสดงแก่ผู้ใช้ในการเลือกเว็บเวอร์วิสที่ต้องการได้ตามต้องการ

1.6 งบประมาณที่ใช้

1.6.1 ค่าวัสดุสำนักงาน	800	บาท
1.6.2 ค่าเอกสาร	700	บาท
1.6.3 ค่าใช้จ่ายในการจัดทำรูปเล่มรายงาน	500	บาท
รวมเป็นเงินทั้งสิ้น(สองพันบาทถ้วน)	2,000	บาท

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงหลักการและทฤษฎีที่เกี่ยวข้องในการนำมาประยุกต์ใช้และพัฒนาระบบการค้นหาเว็บเซอร์วิส ซึ่งสิ่งที่จะศึกษาในบทนี้มีดังนี้

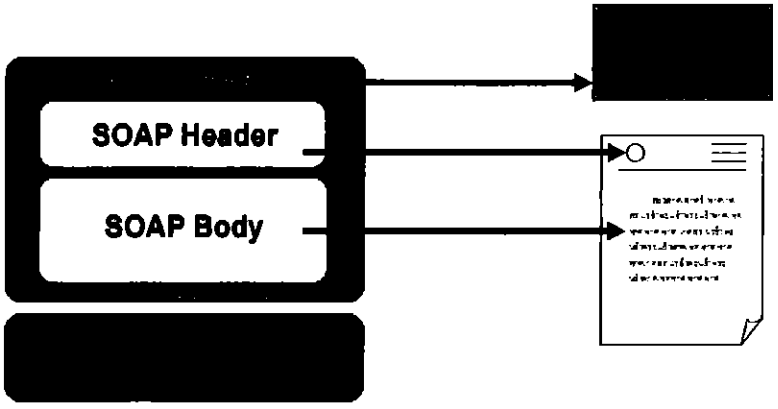
2.1 เว็บเซอร์วิส (Web Services) [6]

เว็บเซอร์วิส (Web Services) เป็นแอพพลิเคชั่นที่ทำงานในลักษณะให้บริการแก่แอพพลิเคชั่นหรือโปรแกรมอื่นๆผ่านเว็บ เป็นระบบซอฟต์แวร์ที่ออกแบบมาเพื่อสนับสนุนการแลกเปลี่ยนข้อมูลกัน ระหว่างเครื่องคอมพิวเตอร์ผ่านระบบเครือข่าย โดยใช้ XML (The Extensible Markup Language) เป็นสื่อในการแลกเปลี่ยนข้อมูลระหว่างกัน โดยผู้ใช้สามารถเรียกใช้บริการต่างๆของแอพพลิเคชั่นที่อยู่บนแพลตฟอร์มใดๆก็ได้ โดยไม่จำเป็นต้องรู้ที่อยู่จริงของแอพพลิเคชั่นหรือโปรแกรมนั้นๆ

ในการเรียกใช้งานเว็บเซอร์วิสนั้น จำเป็นต้องรู้คุณลักษณะของเว็บเซอร์วิสจากเอกสาร WSDL (Web Services Description Language) มีการร้องขอบริการจากผู้ขอใช้บริการ การรอผลจากผู้ให้บริการ โดยใช้โพรโตคอล SOAP (Simple Object Access Protocol) และสิ่งที่สำคัญในการเรียกใช้งานเว็บเซอร์วิสคือ ต้องรู้ที่อยู่ของบริการนั้นๆ จาก UDDI (Universal Description, Discovery, and Integration) เพื่อสามารถค้นหาข้อมูลเว็บเซอร์วิสได้ตามต้องการ

2.1.1 มาตรฐานหลักของเว็บเซอร์วิส (Web Services)

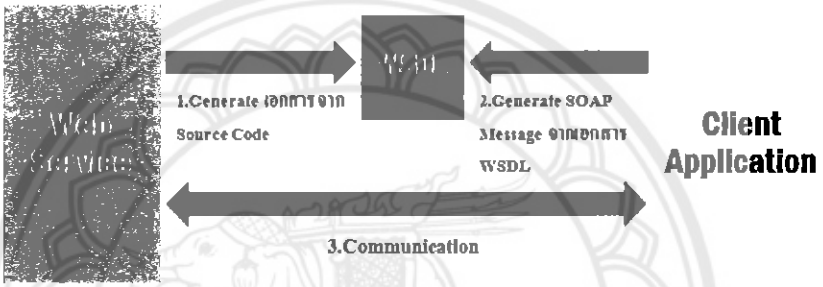
2.1.1.1) SOAP (Simple Object Access Protocol) เว็บเซอร์วิสเป็นรูปแบบของการออกแบบโมเดลสื่อสาร ในลักษณะของการกระจายการติดต่อสื่อสารที่เป็นตัวกลาง



รูปที่ 2.1 องค์ประกอบของ SOAP

โดยโพรโทคอลที่ใช้ในการสื่อสารคือ SOAP (Simple Object Access Protocol) เป็นโพรโทคอล ในการติดต่อแลกเปลี่ยนข้อมูลระหว่างกันเป็นโพรโทคอลการสื่อสารในระดับ Application Layer หรือในระดับแอพพลิเคชั่นโดยอาศัยผ่านอินเทอร์เน็ตโพรโทคอล อย่างเช่น HTTP, SMTP, FTP เป็นต้น เป็นโพรโทคอลที่พัฒนารากฐานมาจาก XML โดยมาตรฐานของ SOAP

2.1.1.2) WSDL (Web Services Description Language) เป็นภาษา XML ที่ใช้ในการอธิบายเว็บเซอร์วิส ซึ่งทำให้ผู้เรียกใช้เว็บเซอร์วิสหรือโปรแกรมที่ต้องการเรียกใช้เว็บเซอร์วิสทราบข้อมูลของโอเปอร์เรชันในการให้บริการ ในการส่งข้อมูลและได้รับข้อมูลกลับมาตลอดจนทราบอินเทอร์เน็ตโปรโตคอลที่จะต้องใช้ในการติดต่อเว็บเซอร์วิสและที่อยู่ของเว็บเซอร์วิส

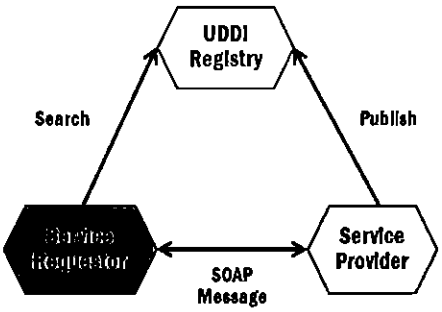


รูปที่ 2.2 Dynamic communication

2.1.1.3) UDDI (Universal Description, Discovery and Integration) เป็น Directory ที่เก็บหรือลงทะเบียนเว็บเซอร์วิสของผู้ให้บริการ สามารถใช้ UDDI ในการประกาศว่า บริการใดๆบ้างที่ให้บริการ ให้ ผู้ใช้สามารถค้นหาเว็บเซอร์วิส ที่ต้องการผ่าน UDDI ได้

2.1.2 ความแตกต่างของ UDDI กับระบบการค้นหาเว็บเซอร์วิส

UDDI เป็นเทคโนโลยีหนึ่งของเว็บเซอร์วิส สำหรับให้ผู้ให้บริการทำการลงทะเบียนข้อมูลเว็บเซอร์วิสบน UDDI Registry จุดประสงค์ของ UDDI Registry นั้นเพื่อประกาศการให้บริการของตนเองสู่สาธารณะ ให้ผู้ใช้บริการทำการเลือกค้นหาข้อมูลและบริการที่มีทั้งหมดจากระบบนี้ UDDI Registry เป็นเพียงระบบลงทะเบียนขององค์กรผู้ให้บริการเท่านั้น



รูปที่ 2.3 UDDI ใน Web Services

เนื่องจากงานด้านนี้มีข้อมูลจำนวนมาก การค้นหาเว็บเซอร์วิสดังกล่าว จึงมีความยุ่งยากและไม่สะดวกพอสำหรับการค้นหา เนื่องจากเป็นการค้นหาจากข้อมูลทั้งหมดที่มีจำนวนค่อนข้างมาก จึงมีการพัฒนา "ระบบการค้นหาเว็บเซอร์วิส" เพื่อใช้เป็นเทคโนโลยีทดแทนสำหรับการค้นหา เพิ่มความสะดวกและลดระยะเวลาในการค้นหาจากข้อมูลของผู้ให้บริการที่มีจำนวนมาก ใน UDDI Registry

2.2 เอ็กซ์เอ็มแอล (XML)

XML (Extensive Markup Language) เป็นภาษาที่พัฒนาขึ้นมาใหม่เพื่อมาช่วยเสริมการทำงานของ HTML ซึ่งเป็นการทำงานกับข้อมูลโดยตรงสามารถให้รายละเอียดเกี่ยวกับข้อมูลและการเปลี่ยนแปลงข้อมูลโดยแอพพลิเคชั่นบนเว็บได้ชัดเจน และใช้ฟอร์มที่ยืดหยุ่นได้ตามมาตรฐาน HTML หรือ Hyper Text Markup Language XML มีการกำหนดกฎและรายละเอียดของเนื้อหาเอกสารที่เรียกว่า Document Type Definition (DTD) และ XML Schema

XML จะเป็นส่วนหนึ่งของ HTML ซึ่ง XML จะให้รายละเอียดเกี่ยวกับข้อมูล เช่น ชื่อเมือง อุณหภูมิ ความกดอากาศ ส่วน HTML เป็นการกำหนดแท็กต่างๆ ที่จะทำให้ข้อมูลแสดงออกมาในรูปแบบไหน ซึ่งข้อมูลจะสามารถแสดงออกมาได้หลายรูปแบบ ไม่ว่าจะเป็นตารางหรือข้อความธรรมดา ขึ้นอยู่กับการกำหนดของ HTML และในปัจจุบันนี้ ด้วย XML จะมีการให้รายละเอียดของเนื้อหาเอกสารที่เรียกว่า Document Type Definition (DTD) ที่ให้รายละเอียดเกี่ยวกับตัวเอกสารว่าจะแสดงหรือซ่อนส่วนไหนของเอกสารบ้าง ซึ่ง DTD จะเป็นส่วนที่เพิ่มเติมสำหรับ XML ถ้าหากมีการส่งข้อมูลในรูปแบบ DTD ก็จะรู้กันว่าเป็น XML

2.2.1 องค์ประกอบของ XML

XML จะประกอบด้วย 3 ส่วนพื้นฐาน คือ เอกสารข้อมูล (Data document) เอกสารนิยามความหมาย (definition document) และนิยามภาษา (definition language) การใช้งาน XML จำเป็นต้องใช้ร่วมกับ Style Sheet หรือมาตรฐานอื่น ๆ เพราะ XML เพียงแต่กำหนดรูปแบบของแท็กเท่านั้น ไม่ได้กำหนดว่าแท็กจะแสดงผลแบบใด ดังนั้นหากเอาข้อมูลในรูปแบบ XML ไป

แสดงผลในอุปกรณ์ชนิดใดก็ตาม จะต้องกำหนดวิธีแสดงผลของอุปกรณ์นั้นด้วย นอกจากนี้ XML ยังสนับสนุนภาษาต่างชาติ โดยใช้มาตรฐาน ISO 10646 จุดมุ่งหมายของภาษา XML คือ ภาษาต้องเรียบง่าย มีคำสั่งน้อยที่สุด สามารถเขียนด้วยโปรแกรมแก้ไขข้อความ (Text Editor) และสนับสนุนการทำงานร่วมกับ แอพพลิเคชั่นได้หลายชนิด โดยมีส่วนประกอบหลักๆ ดังนี้

1. อิลิเมนต์ (Element)
2. แท็ก (Tag)
3. แอตทริบิวต์ (Attribute)
4. เอ็นติตี้ (Entity)

ตัวอย่าง XML มีลักษณะดังนี้

```
<book>
 <chap number="1">
     Text for Chapter 1
 </chap>
 <chap number="2">
     Text for Chapter 2
 </chap>
</book>
```

จากตัวอย่างโครงสร้าง เมื่อพิจารณาจะเห็นได้ว่า มีรูปแบบของจำนวนบทเรียนอยู่ 2 บท ในแต่ละบทมีข้อความอยู่บางส่วน เช่น บทที่ 1 มีข้อความ Text for Chapter 1 ที่อยู่ระหว่าง element chap จะมี attribute ชื่อ number โดยบทที่ 1 ใช้ number ="1" เป็นข้อมูลของ attribute สามารถเรียกใช้และเข้าใจได้ด้วยมาตรฐานเดียวกัน ทำให้สามารถเรียกใช้และเข้าใจได้ง่าย

XML เป็นภาษาที่ให้ความชัดเจนในการให้รายละเอียดเกี่ยวกับข้อมูล และการเปลี่ยนแปลงข้อมูลโดยแอพพลิเคชันบนเว็บและใช้ฟอร์มที่ยืดหยุ่นได้ตามมาตรฐาน HTML หรือ Hyper Text Markup Language ได้เปิดโลกแห่งการแสดงข้อมูลต่างๆ มานำเสนอ ส่วน XML จะทำให้การทำงานกับข้อมูลโดยตรงที่เสริมกับการทำงานของ HTML [10]

2.2.2 ข้อดีของ XML

จุดเด่นของ XMLคือ สามารถดูเอกสารได้ง่าย สะดวก และได้ผลดีเหมือน HTML เป็นการเน้นความกะทัดรัด เข้าใจง่าย และได้ประโยชน์กว้างขวาง สนับสนุนการประยุกต์ใช้กับงานต่าง ๆ และสนับสนุนโปรแกรมประยุกต์ต่าง ๆ เหมาะกับงานทางด้านการวิเคราะห์เอกสาร การผลิตเอกสาร การแลกเปลี่ยน และการแสดงผล XML สามารถเขียนได้ง่าย สามารถให้ผู้ใช้อื่นร่วมใช้ได้โดยไม่ต้องอาศัยโปรแกรมหรือเครื่องมือช่วยแปล

การเขียน XML ทำได้ตั้งแต่การใช้ Text editor ทั่วไป และไม่ต้องการเครื่องมือที่ซับซ้อน ซึ่งเป็นมาตรฐานที่กำหนดแล้วใช้งานได้ทันที โดยที่บราวเซอร์และอุปกรณ์ต่าง ๆ พร้อมใช้งานร่วมกัน รูปแบบการเขียนโครงสร้างข้อกำหนดเป็นไปตามหลักการ คือ เมื่อเขียนแล้วต้องสามารถ

ใช้โปรแกรมแปลภาษาได้ง่าย โดยทั่วไปเขียนในรูปแบบ BNF (Baches Normal Form) ได้ ใช้เป็นตัวควบคุมข้อมูล (Meta data) จึงเป็นแนวทางในการขนส่งข้อมูล และสร้างการเชื่อมโยงระหว่างแอปพลิเคชั่นได้ง่าย มีการสนับสนุน UNICODE ทำให้ใช้ได้หลากหลายภาษา และผสมกันได้หลากหลายภาษา สามารถดึงเอกสาร XML มาใช้งานได้ง่าย และใช้ร่วมกับโปรแกรมประยุกต์อื่นๆ ได้ง่ายเช่นกัน ช่วยทำให้เกิดการรับส่งข้อมูลแบบ EDI (Electronic Data Interchange) โดยทำให้แนวทางการเชื่อมโยงและสร้างความเป็นเอกสารหรือมาตรฐานระหว่างองค์กร ช่วยในการขนส่งข้อมูลไปยังปลายทางเพื่อให้แปลความหมายและใช้งานได้อย่างเต็มประสิทธิภาพ

2.3 ระบบการค้นหา (Search Engine)

ระบบการค้นหา (Search Engine) เป็นเครื่องมือการค้นหาข้อมูลผ่านอินเตอร์เน็ตรูปแบบหนึ่ง ที่สามารถค้นหาข้อมูลผ่านอินเตอร์เน็ตได้ โดยการกรอกข้อมูลคำค้นหาที่ต้องการค้นหา เข้าไปที่ช่อง Search Box เพื่อได้ข้อมูลที่ต้องการค้นหา ข้อมูลจะถูกแสดงออกมาเพื่อให้เราสามารถเลือกรายการข้อมูลที่ตรงกับความต้องการมากที่สุด เพื่อนำไปใช้งาน โดยลักษณะการแสดงผลของระบบการค้นหา (Search Engine) นั้น จะทำการแสดงผลแบบเรียงอันดับตามความสำคัญของคำค้นหา ให้เป็นผลการค้นหาผ่านหน้าจอคอมพิวเตอร์

2.3.1 ประเภทของระบบการค้นหา (Search Engine)

2.3.1.1) แบบอาศัยการจัดเก็บข้อมูลเป็นหลัก (Crawler Based Search Engines) เป็นเครื่องมือค้นหาแบบอาศัยการบันทึกข้อมูล และจัดเก็บข้อมูลเป็นหลัก มีความแม่นยำ และการประมวลผลการค้นหาสามารถทำได้อย่างรวดเร็ว จึงทำให้มีบทบาทในการค้นหาข้อมูลมากที่สุดในปัจจุบัน

2.3.1.2) แบบอ้างอิงใน "ชุดคำสั่งเมตะ" (Meta Search Engine) ระบบค้นหาที่ใช้หลักการในการค้นหาโดยอาศัย Meta Tag ในภาษา HTML ซึ่งมีการประกาศชุดคำสั่งต่าง ๆ เป็นรูปแบบของ Tex Editor ด้วยภาษา HTML ที่ใช้สำหรับประกาศข้อมูลสำคัญต่างๆ ของเอกสารหน้าเพจนั้นๆ เช่น ชื่อผู้พัฒนา คำค้นหา เจ้าของเว็บไซต์ หรือ บล็อกที่เป็นคำอธิบายของเว็บหรือบล็อกที่ต้องการสื่อให้ผู้อื่นเข้าใจจุดประสงค์ของเว็บไซต์นั้นๆ

2.3.2 ประโยชน์ของระบบการค้นหา (Search Engine)

เครื่องมือที่ช่วยในการค้นหา (Search Engine) มีประโยชน์อย่างมากต่อผู้ใช้งานอินเตอร์เน็ต เนื่องจากข้อมูลข่าวสารบนโลกอินเตอร์เน็ตมีมากมาย และเมื่อผู้ใช้ต้องการข้อมูลสารสนเทศใดๆ จำเป็นต้องมีเครื่องมือที่ช่วยในการค้นหา เพื่อให้ได้มาซึ่งข้อมูลสารสนเทศที่ผู้ใช้งานต้องการ

ประโยชน์ของระบบค้นหาจะขึ้นอยู่กับ ความสามารถในการค้นหาข้อมูลที่ต้องการได้สะดวกรวดเร็ว และถูกต้อง สามารถค้นหาข้อมูลแบบเจาะลึกได้ เช่น หนัง รูป หนังสือ เป็นต้น รวมถึงสามารถรองรับการค้นหาหลายภาษา

2.3.3 การทำงานของระบบการค้นหา (Search Engine)

ประโยชน์ของระบบค้นหา (Search Engine) นั้น สำหรับผู้ใช้โดยทั่วไปจะใช้ประโยชน์ในการค้นหาสารสนเทศที่ต้องการในอินเตอร์เน็ตได้อย่างรวดเร็วและสะดวกขึ้น สำหรับองค์กรธุรกิจที่ต้องการให้เว็บไซต์ของตนมีผู้มาเยี่ยมชมมากๆ จะต้องวางแผนพัฒนาเว็บโดยคำนึงถึงระบบว่าทำอย่างไรให้ที่อยู่ของเว็บ ปรากฏในฐานข้อมูลของระบบค้นหาด้วย เพราะในแต่ละระบบค้นหา จะมีซอฟต์แวร์ที่คอยเก็บรวบรวมสารสนเทศของเว็บไซต์ที่เปิดใหม่ เพื่อปรับปรุงฐานข้อมูลให้ทันสมัยอยู่เสมอ ซึ่งควรออกแบบให้เว็บไซต์ประกอบด้วยข้อความที่สำคัญที่ผู้ใช้ทั่วไปจะใช้เป็นคำค้นเพื่อเข้าสู่เว็บไซต์ขององค์กรได้

เว็บไซต์ที่เป็นระบบค้นหา (Search Engine) จะแสดงวัตถุประสงค์เพื่อค้นหาข้อมูลให้กับผู้ใช้อย่างชัดเจนและเริ่มทำงานทันทีเมื่อมีผู้ใช้ใส่คำค้นลงไป ค้นหาตามดัชนี หาคำที่เก็บไว้ในฐานข้อมูลอย่างรวดเร็ว ไม่ว่าจะค้นพบข้อมูลนั้นหรือไม่ จะแสดงผลลัพธ์บนหน้าจอให้ผู้ใช้ภายในไม่กี่วินาที ซึ่งในแต่ละระบบค้นหา (Search Engine) จะมีการสร้างฐานข้อมูลและปรับปรุงข้อมูล 2 วิธีดังนี้

a) รวบรวมข้อมูลเอง โดยใช้ซอฟต์แวร์อัตโนมัติ (Intelligent Software) เป็นโปรแกรมชื่อว่าสไปเดอร์ (Spider) หรือครอว์เลอร์ (Crawler) หรือโรบอท (Robots) ซึ่งจะออกไปเยี่ยมเว็บไซต์ต่างๆ และเก็บรวบรวมข้อมูลสารสนเทศต่างๆ เช่น URL (Universal Resource Locator) ของเว็บไซต์ที่เปิดใหม่ ชื่อที่เกี่ยวข้องกับเว็บไซต์ สารสนเทศที่ที่ปรากฏอยู่บนเว็บไซต์ เป็นต้น และจะเก็บข้อมูลกลับมายังฐานข้อมูลแม่ข่ายเว็บในเครื่องแม่ข่ายเว็บของระบบค้นหา (Search Engine)

b) ผู้สร้างเว็บไซต์เก็บรวบรวมข้อมูลให้ โดยผู้สร้างแต่ละเว็บส่ง URL และคำอธิบายโดยย่อเกี่ยวกับเว็บไซต์นั้น ส่งไปให้ระบบค้นหา (Search Engine) เอง เว็บไซต์ที่ต้องการให้ปรากฏเป็นชื่อต้นๆ ในการแสดงผลของระบบค้นหา (Search Engine) ต่างๆ ได้ โดยผู้เขียนเว็บไซต์สามารถใช้ตัวเมตาแท็ก (meta tag) เป็นตัวช่วยจัดอันดับเว็บไซต์ให้ปรากฏในตอนต้น เมื่อแสดงผลการค้นหาด้วยเมตา แท็กซึ่งเป็นส่วนหนึ่งของภาษา HTML ที่จะปรากฏอยู่ที่บริเวณส่วนต้นของไฟล์ HTML ซึ่งภายในจะบรรจุคำต่างๆที่มีความหมายสำหรับอธิบายเนื้อหาสาระของเว็บไซต์ส่วนนั้น ส่วนนี้เป็นส่วนสำคัญสำหรับในกรณีที่ต้องการให้เว็บไซต์นั้นอยู่ในอันดับต้นๆของการแสดงผลจากการค้นหา

สิ่งที่ควรพิจารณาในการค้นหาข้อมูลบนอินเตอร์เน็ตให้มีประโยชน์และใช้เวลารวดเร็วขึ้น ควรมีการวางแผนก่อนลงมือค้นหา การกำหนดคำค้นที่เกี่ยวข้องกับข้อมูลที่ต้องการหา เช่น ค้นหาข้อมูลเกี่ยวกับการใช้งานโปรแกรมลินุกซ์ (Linux) ควรจะทราบคำที่เกี่ยวข้อง เช่น software, operating system, open operating system เป็นต้น เพื่อจะนำคำที่ใกล้เคียงเล่านี้ไปค้นหาข้อมูลที่ต้องการ และตัดสินใจได้ว่าต้องการข้อมูลอะไรต่อไป ซึ่งจะช่วยให้การค้นหาข้อมูลเร็วขึ้น

2.3.4 ตัวอย่างเว็บไซต์ที่ให้บริการระบบค้นหา (Search Engine)

ปัจจุบันเว็บไซต์บนโลกอินเตอร์เน็ตนั้นมีจำนวนมากมายมหาศาล บรรจุข้อมูลข่าวสารมากมาย เมื่อมีข้อมูลที่มากขึ้น ผู้ใช้สามารถสืบค้นหาข้อมูลที่ต้องการจริง ๆ สามารถทำได้ยาก การใช้วิธีสุ่มไปเรื่อย ๆอาจจะใช้เวลานานในการพบข้อมูลที่ต้องการ ถึงแม้มีข้อมูลทุกอย่างที่ต้องการอยู่บนเว็บ แต่ไม่สามารถเอาออกมาใช้งานได้ ด้วยเหตุนี้ จึงมีการคิดค้นเทคโนโลยีขึ้นมาช่วยให้ผู้ที่ใช้อินเทอร์เน็ตเพื่อค้นหาข้อมูลที่ต้องการ เทคโนโลยีนี้เรียกว่า Search Engine ซึ่งมีผู้ให้บริการมากมายในการค้นหาข้อมูล ผู้ให้บริการหลัก ๆ ที่นิยมใช้นั้น มีหลักการทำงานที่แตกต่างออกไป ขึ้นอยู่กับวัตถุประสงค์ของการใช้งาน ดังนี้

a) Yahoo (www.yahoo.com) เป็น Search engine ที่รู้จักกันดีมาก เพราะเป็นรายแรกๆที่ทำระบบการค้นหาขึ้น ชื่อ yahoo ออกเสียงง่าย จดจำได้ง่าย มีลักษณะการค้นหาในแบบเมนู คือบนจอภาพจะแบ่งหัวข้อต่าง ๆ ออกเป็นหมวดหมู่ใหญ่ ๆ หรือเป็นหัวข้อหลัก ในแต่ละหัวข้อหลักจะประกอบไปด้วยหัวข้อย่อยประกอบอยู่อีกหลายหัวข้อ ผู้ใช้สามารถเจาะลงไปทีละหัวข้อที่เกี่ยวข้องจนกระทั่งไปถึงหัวข้อย่อยที่ต้องการ หรือใส่คำค้นหาลงไปช่องค้นหาแล้วสั่งให้โปรแกรมค้นหาสิ่งนั้นก็ได้ yahoo มีข้อจำกัดในบางเรื่องเมื่อเปรียบเทียบกับระบบการค้นหาใหม่ๆ เนื่องจากมีฐานข้อมูลที่น้อยกว่าแหล่งอื่นๆ ทำให้ผลลัพธ์ในการค้นหาแคบและจำกัดและเทคนิคในการค้นหาข้อมูลมีความคล่องตัวน้อยกว่า คือจะค้นหาข้อมูลตามที่พิมพ์เข้าไปมากกว่าการที่จะค้นหาข้อมูลตามความหมายของคำ

b) Excite (www.excite.com) เป็นระบบการค้นหาที่มีไซต์ต่างๆในฐานข้อมูลมากกว่า 1,500,000 ไซต์ และสามารถค้นหาข้อมูลที่เป็นคำหรือความหมายของคำได้ จะค้นหาข้อมูลจาก www และกลุ่มใหม่เป็นหลัก การที่ Excite มีข้อมูลในฐานข้อมูลจำนวนมาก ทำให้ผลลัพธ์ที่ได้มากตามไปด้วย เทคนิคที่ใช้คือพยายามตีความหมายข้อมูลที่ต้องการค้นหากับข้อมูลแต่ละเว็บไซต์ ว่ามีเว็บไซต์ได้ตรงความหมายกับคำที่ต้องการ ไม่ได้ดูเฉพาะการเปรียบเทียบคำให้เหมือนกันเท่านั้น ทำให้ค้นหาข้อมูลได้ตรงความต้องการมากกว่าการเปรียบเทียบธรรมดา ๆ ที่เคยใช้กันทำให้ได้รับความนิยมมากพอควร นอกจากนี้ Excite ยังมีโปรแกรมคอยปรับปรุงฐานข้อมูลให้ทันสมัยอยู่เสมอที่เรียกว่าสไปเดอร์ ทำให้ข้อมูลในฐานข้อมูลมีความถูกต้องอยู่ตลอดเวลา และสามารถค้นหาข้อมูลแบบตรรกะ โดยใส่คำ AND,OR,NOT และใช้วงเล็บ ( ) เพื่อกำหนดสิ่งที่ต้องการค้นหาให้เจาะจงยิ่งขึ้น รวมทั้งใช้เครื่องหมาย + และ – เพื่อกำหนดว่ารวมหรือไม่รวมข้อมูลใดบ้างได้อีกด้วย

c) Google (www.google.com) เป็นระบบการค้นหาที่มีเทคโนโลยีการค้นหาก้าวหน้ามากที่สุดตัวหนึ่งในอินเตอร์เน็ต มีฐานข้อมูลเชื่อมโยงไว้ในประเทศต่างๆและสามารถใช้ภาษาต่างๆ เป็นคำค้นได้นอกเหนือจากภาษาอังกฤษ โดยจะแบ่งประเภทการค้นหาเป็นหัวข้อไว้ เช่น ค้นหาเว็บไซต์ (Web) รูปภาพ (image) กลุ่มข่าว (Groups) ข่าว(News) หลักการค้นหาข้อมูลของ Google ใช้หลักการพื้นฐาน (Concept Based) เดียวกับ Excite หรือสามารถหาข้อมูลที่ต้องการโดย

ใช้ความหมายของข้อมูล ดังนั้นแม้ว่าข้อมูลบางแห่งจะไม่มีคำที่สั่งให้ค้นหาอยู่เลย แต่ถ้ามีความหมายตรงกันจะถือว่าข้อมูลนั้นตรงกับสิ่งที่ค้นหา ซึ่งถ้าใช้วิธีค้นหาข้อมูลด้วยวิธีเปรียบเทียบคำแบบตรงๆ จะได้ผลลัพธ์ที่แคบกว่าและมีจำนวนข้อมูลที่น้อยกว่า

ผลลัพธ์ที่ได้จากการค้นข้อมูลใน Google จะแสดงเป็นหัวข้อเรียงตามลำดับความใกล้เคียงกับสิ่งที่ต้องการค้นหา พร้อมกับบอกขนาดของเอกสารนั้นด้วยว่ามีขนาดกี่กิโลไบต์ นอกจากนี้ยังแสดงข้อความย่อของแต่ละหัวข้อนั้นประกอบด้วย เพื่อจะได้อ่านดูว่ามีเอกสารใดตรงกับสิ่งที่ต้องการค้นหา

2.3.5 เทคนิคการค้นหาข้อมูลโดยใช้บูลีนโอเปอร์เรเตอร์ [11]

การค้นหาข้อมูลในอินเตอร์เน็ตโดยใช้ระบบค้นหานั้น ผู้ใช้ส่วนใหญ่จะใช้วิธีการค้นหาข้อมูลแบบพื้นฐาน คือ ใช้คำสำคัญในการค้นหาหรือเข้าไปในไดเร็กทอรี (Directory) ที่จัดหมวดหมู่ไว้ให้แล้ว โดยไม่มีการค้นหาแบบพิเศษ แต่ระบบค้นหาแต่ละตัวนั้นจะมีความสามารถให้ผู้ใช้ค้นหาข้อมูลที่แตกต่างกัน บางตัวสามารถให้ผู้ใช้ค้นหาคำได้มากกว่า 1 คำ หรือสามารถให้ผู้ใช้ขจัดคำบางคำที่ไม่ต้องการออกไปได้ ระบบค้นหาหลายๆตัว สามารถนำมาให้ผู้ใช้นำเอาบูลีนโอเปอร์เรเตอร์ (Boolean operator) มาใช้ร่วมค้นหาข้อมูล คือ AND, OR และ NOT ซึ่งเป็นตัวแปรตรรกะมาจำกัดจำนวนข้อมูลที่ต้องการ

2.3.6.1) การใช้ AND การสร้างความสัมพันธ์ระหว่างข้อมูล 2 กลุ่ม จะเกิดกลุ่มใหม่ที่มีลักษณะของข้อมูลเดิมทุกกลุ่ม ข้อมูลกลุ่มใหม่จึงมีขนาดเล็กลง หรือแคบลง ข้อมูลที่ค้นหามาได้จะประกอบด้วยคำทั้งหมดจะถูกนำมาแสดงผล เช่น Computer AND Design ผลลัพธ์ที่ได้ จะได้ข้องมูลที่ต้องมีคำว่า Computer และ Design อยู่ด้วยกันเท่านั้น จึงจะดึงข้อมูลนั้นมาแสดงผล

2.3.6.2) การใช้ OR การสร้างความสัมพันธ์ระหว่างข้อมูล 2 กลุ่ม จะเกิดกลุ่มใหม่ที่มีลักษณะของข้อมูลเดิมจากกลุ่มใดกลุ่มหนึ่งหรือทุกกลุ่ม ดังนั้นข้อมูลใหม่จะมีขนาดใหญ่ขึ้นหรือมีขอบเขตกว้างขึ้น เช่น Computer OR Design หมายความว่า "ให้ค้นหาข้อมูลที่มีคำว่า Computer แต่ต้องไม่มีคำว่า Design มาด้วย" ฉะนั้น ผลลัพธ์ที่ได้เมื่อใช้ข้อความนี้ค้นหาข้อมูลจะแสดงข้อมูลที่มีคำว่า Computer เท่านั้น ถ้าข้อมูลใดมีคำว่า "Design" อยู่ด้วย จะไม่ดึงเอาข้อมูลนั้นมาแสดง

2.3.6.3) การใช้ NOT การสร้างความสัมพันธ์ระหว่างข้อมูล 2 กลุ่ม จะเกิดกลุ่มใหม่ที่ไม่มีลักษณะของข้อมูลเดิมที่มี NOT นำหน้า เช่น เมื่อผู้ใช้ค้นหา animal NOT dog สิ่งที่ Search engine แสดงผลให้คือเว็บไซต์ที่เกี่ยวข้องกับ animal ทั้งหมด ยกเว้นเว็บไซต์ที่เกี่ยวข้องกับ Dog

นอกจากนี้แล้ว Search Engine บางตัวสามารถใช้เครื่องหมาย +, * และ – แทนบูลีนโอเปอร์เรเตอร์ ดังนี้

เครื่องหมาย + หมายความว่า คำใดที่ตามหลังเครื่องหมายนี้ จะต้องไม่มีคำนั้นอยู่ในเว็บไซต์นั้นเหมือนคำว่า AND

เครื่องหมาย - หมายความว่า คำใดที่ตามหลังด้วยเครื่องหมายนี้ จะต้องไม่มีคำนั้นอยู่ในเว็บไซต์นั้น เหมือนกับคำว่า NOT

ตัวอย่างเช่น +Computer - Design ข้อมูลที่จะแสดงออกมา จะต้องมีคำว่า Computer แต่ไม่มีคำว่า Design

เครื่องหมาย * แทน OR เช่น Dog* Cat แทน Dog OR Cat เป็นต้น

2.3.6 การค้นหาข้อมูลด้วยวิธีอื่น ๆ

การค้นหาสารสนเทศนอกจากจะต้องเลือกใช้คำศัพท์แทนความต้องการ และสร้างความสัมพันธ์ระหว่างคำศัพท์ด้วยบูลีนโอเปอร์เรเตอร์แล้ว ยังมีเทคนิคอื่นๆในการใช้คำเพื่อค้นหาสารมนเทศอีก ได้แก่ ระบุระยะห่างระหว่างคำค้นและการตัดคำ เทคนิคดังกล่าวจะช่วยอำนวยความสะดวกและช่วยกำหนดรูปแบบคำที่จะใช้ค้นหาใกล้เคียงกับความต้องการมากที่สุด

2.3.7.1) การระบุระยะห่างระหว่างคำค้น การระบุระยะห่างระหว่างคำค้น (Word Proximity) เป็นเทคนิคที่ใช้การระบุระยะห่างระหว่างคำค้น 2 คำ ที่ปรากฏอยู่ในระบบเดียวกัน นอกจากนี้ยังอาจระบุการเรียงลำดับก่อนหลังของคำค้นได้อีกด้วย การระบุระยะห่างระหว่างคำค้นมักใช้กับการค้นหาสารสนเทศที่เป็นภาษาอังกฤษ เพราะมีการเว้นระยะห่างระหว่างคำที่แน่นอน

a) การใช้ NEAR หรือ (N) ระบุระยะห่างระหว่างคำค้น ผลลัพธ์ที่ได้จากคำค้นนี้จะเรียงลำดับก่อนหลังของคำค้น มีรูปแบบและผลการค้นหา ดังนี้

การใช้คำสั่ง	ผลการค้นหา
marketing(N) target	marketing target
	target margeting

b) การใช้ WITH หรือ (W) ระบุระยะห่างระหว่างคำค้น โดยคำที่ค้นจะต้องเรียงลำดับตามที่กำหนดรูปแบบและผลการค้นหาดังนี้

การใช้คำสั่ง	ผลการค้นหา
marketing(W) target	marketing target

2.3.7.2) การตัดคำ (Truncation) คือการละอักขระบางตัวของคำศัพท์แล้วแทนที่ด้วยเครื่องหมายอื่นที่ Search engine นั้นกำหนด ซึ่งจะใช้เครื่องหมายไม่เหมือนกัน เครื่องหมายที่นิยมใช้ เช่น เครื่องหมายดอกจัน (*) เครื่องหมายคำถาม (?) เครื่องหมายเปอร์เซ็นต์ (%) เป็นต้น โดยทั่วไปมักจะละอักขระท้ายคำ การใช้เทคนิคการตัดคำจะทำให้ค้นหาคำทุกคำที่มีรากศัพท์เดียวกัน

2.4 ครอว์เลอร์ (Crawler)

2.4.1 เว็บครอว์เลอร์ (Web Crawler)

เว็บครอวเลอร์ (Web Crawler) คือ เครื่องมือการค้นหาบนอินเตอร์เน็ตแบบอาศัยการบันทึกข้อมูลและจัดเก็บข้อมูล เป็นโปรแกรมที่ทำการเลือกหาหน้าเว็บโดยอัตโนมัติ ทำการตรวจหาและทำการจัดเก็บข้อมูล หรือเว็บไซต์ต่าง ๆ ในรูปแบบของการทำสำเนาข้อมูล โดยทำการเข้าถึงเว็บและทำการเข้าไปตามลิงค์ของเว็บนั้นเรื่อยๆ โดยไม่ได้มีข้อจำกัดว่าเว็บนั้นจะเกี่ยวข้องข้องกับอะไร ตรงกับความต้องการของการค้นหามากน้อยแค่ไหน

จุดประสงค์หลักของเว็บครอว์เลอร์ คือการเข้าถึงเว็บไซต์ที่มีอยู่ให้มากที่สุด เพื่อเก็บรวบรวมข้อมูลของเว็บไซต์เหล่านั้นให้ได้มากที่สุด พิจารณา รูปที่ 2.4 เว็บครอว์เลอร์มีชื่อเรียกและวิเคราะห์หลักการการค้นหาต่างกัน ขึ้นอยู่กับวัตถุประสงค์และออกแบบ เช่น Ants, Automatic Indexers, Bots โดยมีหน้าหลักการทำงานคล้ายๆ กัน

รูปที่ 2.4 ลักษณะการทำงานของเว็บครอว์เลอร์

2.4.2 องค์ประกอบของเว็บครอว์เลอร์

เว็บครอว์เลอร์ มีองค์ประกอบหลัก 2 ส่วน เพื่อทำงานในการค้นหาและเก็บรวบรวมข้อมูลที่ต้องการและสามารถใช้งานได้ตรงวัตถุประสงค์ ดังนี้

2.4.2.1) ฐานข้อมูล เว็บครอว์เลอร์จะมีฐานข้อมูลเพื่อเก็บข้อมูลเป็นของตัวเอง ที่มีระบบการประมวลผล และการจัดอันดับที่เฉพาะ เป็นเอกลักษณ์ของตนเอง เพื่อใช้ในการเก็บรวบรวมข้อมูลที่ได้จาการค้นหา

2.4.2.2) ซอฟแวร์ เครื่องมือหลักสำคัญที่สุดอีกส่วนหนึ่งสำหรับระบบการค้นหา เนื่องจากต้องอาศัยโปรแกรมเล็ก ๆ ทำหน้าที่ในการตรวจหา และทำการจัดเก็บข้อมูล หน้าเพ็จ หรือเว็บไซต์ต่าง ๆ ในรูปแบบของการทำสำเนาข้อมูล เหมือนกับต้นฉบับทุกอย่าง ซึ่งเราจะรู้จักกันในนามสไปเดอร์ (Spider) โดย Spider แต่ละองค์กรมีลักษณะที่ต่างกันไป เช่น Google, Yahoo, MSN, Live, Search, Technocratic เป็นต้น

2.4.3 โฟกัสครอว์เลอร์ (Focused Crawler) [1]

เครื่องมือการค้นหาบนอินเตอร์เน็ตอีกแบบหนึ่ง ที่ทำการเลือกหาหน้าเว็บโดยอัตโนมัติ ทำการตรวจหา และทำการจัดเก็บข้อมูล หน้าเพจ หรือ เว็บไซต์ต่าง ๆ ในรูปแบบของการทำสำเนาข้อมูล เป้าหมายของโฟกัสครอว์เลอร์ คือ การเลือกหน้าเว็บที่เป็นไปได้มากที่สุดที่ ที่เกี่ยวข้องกับหัวข้อหรือคำที่ผู้ใช้นั้นต้องการค้นหา โดยมีหลักการที่สำคัญ คือ การตัดสินว่าเว็บเพจนั้นมีความเกี่ยวข้องกับคำที่ต้องการค้นหา เป็นการทำงานที่มีจุดหมาย โดยเข้าไปเก็บข้อมูลเฉพาะหัวข้อและข้อมูลบนเว็บไซต์ที่เราสนใจ ซึ่งต่างจากเว็บครอว์เลอร์ที่มีการทำงานโดยการปล่อย Spider ออกไปเก็บหน้าเว็บของเว็บไซต์ต่างๆ ไว้ให้มากที่สุด จึงทำให้เสียเวลาและเปลืองทรัพยากรในการเก็บข้อมูลจากเว็บไซต์ที่ยิ่งเพิ่มมาก ระบบการค้นหาที่มีการใช้งานของโฟกัสครอว์เลอร์ เช่น Google, Yahoo เป็นต้น

โฟกัสครอว์เลอร์ ใช้ TF.IDF Algorithm ในการเลือกค้นหาคำค้นหาเพียงเฉพาะบางคำค้นหาที่ต้องการ แต่มีการจำกัดด้านเงื่อนไขเฉพาะบางอย่างในการเลือกคำค้นหา คือ เลือกคำค้นหาที่เชื่อมโยงและเกี่ยวข้องกับหัวข้อที่เราต้องการค้นหามากที่สุด จากนั้นใช้ Vector Similarity ตรวจสอบระหว่างเว็บเพจและข้อมูลคำค้นหา เพื่อแสดงให้เห็นความเหมือนหรือความแตกต่างของเว็บเพจกับคำค้นหาเพื่อเลือกเว็บไซต์ที่เกี่ยวข้องกับหัวข้อคำค้นหาที่ผู้ใช้ระบุ ซึ่งโฟกัสครอว์เลอร์ มีประสิทธิภาพควรมีความแม่นยำสูง นอกจากนี้ยังสามารถควบคุมเวลาในการค้นหาได้ดีด้วย

2.4.4 TF.IDF

TF.IDF ย่อมาจาก Term Frequency-Inverse Document Frequency เป็นกระบวนการในการตัดคำและการวัดค่าน้ำหนักของข้อมูล โดยการประเมินค่าความสำคัญของคำนั้นๆ โดยการให้ค่าน้ำหนักตามความถี่ของคำหนึ่งๆ ในกลุ่มของเอกสาร โดยเรานำมาประยุกต์ใช้ในการให้ค่าน้ำหนักความสำคัญกับข้อมูลในเว็บไซต์ เพื่อพิจารณาว่าเว็บไซต์ที่ค้นหามีเนื้อหาเกี่ยวกับอะไร เพื่อเลือกเก็บข้อมูลเว็บไซต์ที่ต้องการ

การกำหนดค่าน้ำหนักของข้อมูลในเอกสารนั้น พิจารณาตามสูตร TF.IDF Weight รูปที่ 2.5 โดยกำหนดให้

N แทนจำนวนเอกสารทั้งหมด

df แทนจำนวนเอกสารที่พบทั้งหมด

$$weight(i,j) = \begin{cases} (1 + \log(tf_{i,j})) \log \frac{N}{df_{i,j}}, & if \quad tf_{i,j} \geq 1 \\ 0, & if \quad tf_{i,j} = 0 \end{cases}$$

รูปที่2.5 TF.IDF Weighting Schemes [1]

2.4.5 Vector Space Model

Vector Space Model ใช้ในการตรวจสอบระหว่างเว็บเพจและข้อมูลคำค้นหา เพื่อแสดงให้เห็นความเหมือนหรือความแตกต่างของเว็บเพจกับคำค้นหาของผู้ใช้ เพื่อใช้ในการพิจารณาเลือกเว็บไซต์ที่ตรงกับความต้องการมากที่สุด Vector Space Model อาศัยเวกเตอร์ในการทำงาน โดยจะแทนเอกสารและชุดของ Query ด้วยเวกเตอร์

จากนั้นเปรียบเทียบหาเอกสารที่มีเวกเตอร์ที่คล้ายกับของคำค้นหาของผู้ใช้มากที่สุด โดยการแทนขนาดแต่ละ Dimension ของเวกเตอร์ ด้วย TF-IDF Weighting และทำการเปรียบเทียบความคล้ายกันของ Vector (Similarity Measurement) ด้วยการทำ Inner product หรือ Cosine product

การเลือกพิจารณา Vector Space Model ด้วย Cosine product สามารถจัดการเปรียบเทียบข้อมูลได้มีประสิทธิภาพมากกว่า จากค่าน้ำหนักของคำในฐานข้อมูลของเอกสารทั้งหมด ดังสูตร Cosine product รูปที่ 2.7

กำหนดให้ W_{qi} คือ ค่าน้ำหนักของคำ

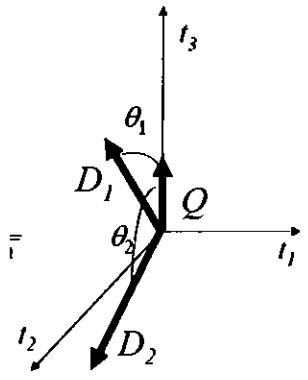
W_i คือ Query Q

W_{di} คือ ค่าน้ำหนักของคำในเอกสาร

$$Sim(Q,D) = \frac{\sum_{i=1}^{n} W_{qi} * W_{di}}{\sqrt{\sum_{i=1}^{n} (W_{qj})^2 * \sum_{i=1}^{n} (W_{di})^2}}$$

รูปที่2.6 Cosine Product [1]

สามารถแสดงความสัมพันธ์ของ Vector Space Model ของ Cosine Product ในรูปของกราฟได้ดังนี้



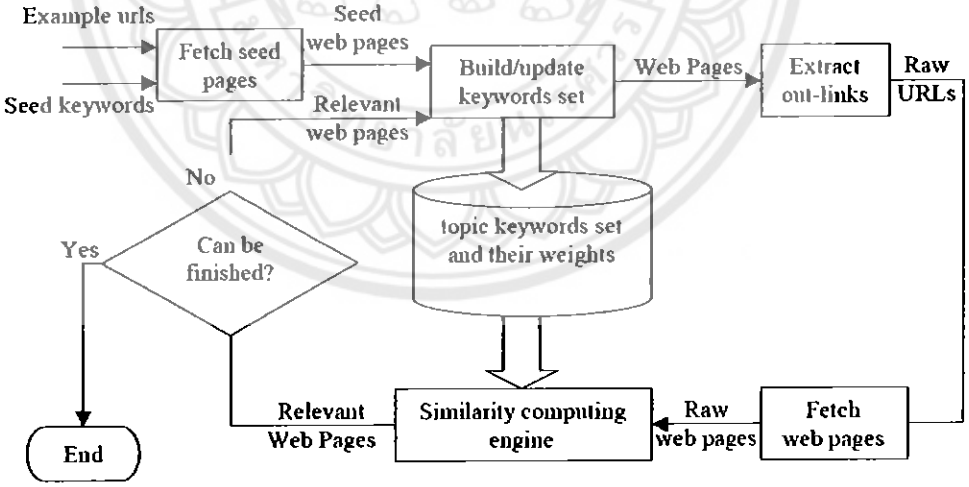
รูปที่2.7 Cosine similarity measure [1]

2.4.6 โครงสร้างของโฟกัสครอวเลอร์

โฟกัสครอวเลอร์ (Focused Crawler) เป็นเครื่องมือการค้นหาบนอินเตอร์เน็ตโดยอาศัยการบันทึกข้อมูลและจัดเก็บข้อมูล ทำการเลือก ตรวจหา และทำการจัดเก็บข้อมูล หน้าเว็บไซต์โดยการทำสำเนาข้อมูล โดยมีโครงสร้างมาจาก 4 ระบบงานย่อย ในการจัดการกับหน้าเว็บในการค้นหาและเก็บรวบรวมข้อมูลที่ต้องการ ดังนี้

1) Seed pages fetching sub-system
2) Topic keywords generating sub-system
3) Similarity computing engine
4) Spider

2.4.6.1) Focused Crawler working process เป็นกระบวนการทำงานของโฟกัสครอวเลอร์ โดยมีเครื่องมือจะทำการค้นหาและสำรวจเว็บไซต์ตาม Keyword ที่ต้องการ แล้วการจัดการค่าน้ำหนักกับคำค้นหาในหน้าเว็บใหม่ๆ แล้วรวบรวมเก็บในฐานข้อมูล จากนั้นทำการบันทึก URL เพื่อนำมาเรียกใช้ในขั้นตอนของการเปรียบเทียบความคล้ายคลึงของคำค้นหาและข้อมูลเว็บไซต์ ในฐานข้อมูล จากนั้นจะได้เว็บไซต์ที่ต้องการ ระบบจะทำการเรียกใช้ URL ที่ต้องการมาแสดงให้ผู้ใช้ตรงกับความต้องการและมีความคล้ายคลึงมากที่สุด ดังรูปที่ 2.8



รูปที่ 2.8 กระบวนการทำงานของ Focused Crawler [1]

2.4.6.2) Seed pages fetching sub-system เป็นกระบวนการค้นหาหน้าเว็บที่ต้องการ นำมาเก็บในฐานข้อมูลนำหน้าเว็บที่ต้องการมาเก็บไว้ในฐานข้อมูล โดยพิจารณาตาม Keyword ที่ต้องการ

2.4.6.3) Topic keywords generating sub-system เป็นการจัดการค่าน้ำหนักให้กับคำในข้อมูลในเว็บไซต์ เพื่อหาความเกี่ยวข้องของเว็บไซต์ เริ่มจากการเรียกหน้าเว็บไซต์ที่เกี่ยวข้อง แล้ว

นำมานับความถี่ของคำด้วย TF.IDF Algorithm แล้วจะได้ข้อมูลว่าเว็บนั้นๆ เกี่ยวข้องกับอะไรเพื่อใช้ในการเปรียบเทียบกับคำค้นหาต่อไป แล้วนำไปเก็บไว้ในฐานข้อมูล

2.4.6.4) Similarity Computing Engine เป็นการเปรียบเทียบความคล้ายคลึงกันของคำค้นหากับข้อมูลในฐานข้อมูล โดยแต่ละเว็บเพจใช้วิธีการวิเคราะห์ด้วย Vector Space Model เมื่อได้การวิเคราะห์ที่ต้องการแล้ว ทำการเรียก URL ของเว็บไซต์มาแสดงเพื่อให้ผู้ใช้ทำการเลือกข้อมูลที่ต้องการ

ครอวเลอร์ ที่มีการใช้ TD.IDF Weighting สามารถใช้สืบค้นได้ถูกต้องมากขึ้น แต่ไม่สามารถใช้เพียงแค่ค่าความถี่ของข้อมูลในเว็บไซต์เพียงเท่านั้นที่จะสามารถทำให้ได้การสืบค้นที่ดีและตรงกับความต้องการ

ในการใช้ TF.IDF ในการให้ค่าน้ำหนักของกลุ่มข้อมูลในหัวข้อคำค้นหา เป็นเหมือนการแทนค่าความถี่ของคำนั้นๆที่มีทั้งหมดในเอกสารทั้งหมดและใช้ Vector Space Model ในการตรวจสอบหาเว็บเพจที่มีความเกี่ยวข้องกับคำค้นหามากที่สุด เพื่อบริการแก่ผู้ใช้ แต่อย่างไรก็ตาม ยังไม่สามารสร้างระบบที่สามารถทำงานได้สมบูรณ์ เพื่อที่จะสามารถประเมินผลของการค้นหาตามคำค้นหา เพราะการให้ค่าน้ำหนักของข้อมูลในเว็บเพจนั้น ไม่สามารถได้ค้นหาและเปรียบเทียบข้อมูลที่ตรงกับคำค้นหาที่เราต้องการค้นหาได้เสมอไป

2.5 ฐานข้อมูล (Database)

ฐานข้อมูล (Database) คือระบบที่รวบรวมข้อมูลไว้ในที่เดียวกัน ซึ่งประกอบไปด้วย แฟ้มข้อมูล (File) ระเบียน (Record) และ เขตข้อมูล (Field) และถูกจัดการด้วยระบบเดียวกัน โปรแกรมคอมพิวเตอร์จะเข้าไปดึงข้อมูลที่ต้องการได้ อย่างรวดเร็ว ซึ่งอาจเปรียบฐานข้อมูลเสมือนเป็น electronic filing system การเข้าถึงข้อมูลในฐานข้อมูลจึงจำเป็นต้องมีระบบการจัดการฐานข้อมูลมาช่วยเรียกว่า database management system (DBMS) ซึ่งจะช่วยให้ผู้ใช้สามารถจัดการกับข้อมูล ตามความต้องการได้ในหน่วยงานใหญ่ๆอาจมีฐานข้อมูลมากกว่า 1 ฐานข้อมูลเช่น ฐานข้อมูลบุคลากร ฐานข้อมูลลูกค้า ฐานข้อมูลสินค้า เป็นต้น [2]

2.5.1 ระบบจัดการฐานข้อมูล

ซอฟต์แวร์สำหรับจัดการฐานข้อมูลนั้น โดยทั่วไปเรียกว่า ระบบจัดการฐานข้อมูล หรือ ดีบีเอ็มเอส (DBMS - Database Management System) สถาปัตยกรรมซอฟต์แวร์ของดีบีเอ็มเอสอาจมีได้หลายแบบ เช่น สำหรับฐานข้อมูลขนาดเล็กที่มีผู้ใช้คนเดียว บ่อยครั้งที่หน้าที่ทั้งหมดจะจัดการด้วยโปรแกรมเพียงโปรแกรมเดียว ส่วนฐานข้อมูลขนาดใหญ่ที่มีผู้ใช้จำนวนมากนั้น ปกติจะประกอบด้วยโปรแกรมหลายโปรแกรมด้วยกัน และโดยทั่วไปส่วนใหญ่จะใช้สถาปัตยกรรมแบบรับ-ให้บริการ (client-server)

โปรแกรมส่วนหน้า (front-end) ของดีบีเอ็มเอส จะเกี่ยวข้องเฉพาะการนำเข้าข้อมูล การตรวจสอบ และการรายงานผลเป็นสำคัญ ในขณะที่โปรแกรมส่วนหลัง (back-end) ซึ่งได้แก่ โปรแกรมให้บริการ เป็นชุดของโปรแกรมที่ดำเนินการเกี่ยวกับการควบคุม การเก็บข้อมูล และการตอบสนองการร้องขอจากโปรแกรมส่วนหน้า โดยปกติแล้วการค้นหาและการเรียงลำดับ จะดำเนินการโดยโปรแกรมส่วนให้บริการ รูปแบบของระบบฐานข้อมูล มีหลากหลายรูปแบบ นับตั้งแต่การใช้ตารางอย่างง่าย ที่จัดเก็บในแฟ้มข้อมูลแฟ้มเดียว ไปจนกระทั่งฐานข้อมูลขนาดใหญ่มาก ที่มีระเบียนหลายล้านระเบียน ซึ่งเก็บในห้องที่เต็มไปด้วยดิสก์ไดรฟ์ หรืออุปกรณ์หน่วยเก็บข้อมูลอิเล็กทรอนิกส์รอบข้าง (peripheral) อื่น ๆ

2.5.2 การออกแบบฐานข้อมูล

การออกแบบฐานข้อมูล (Designing Databases) มีความสำคัญต่อการจัดการระบบฐานข้อมูล (DBMS) เนื่องจากข้อมูลที่อยู่ภายในฐานข้อมูล จะต้องศึกษาถึงความสัมพันธ์ของข้อมูล โครงสร้างของข้อมูล การเข้าถึงข้อมูล และกระบวนการที่โปรแกรมประยุกต์จะเรียกใช้ฐานข้อมูล สามารถแบ่งวิธีการสร้างฐานข้อมูลได้ 3 ประเภท ดังนี้

2.5.2.1) รูปแบบข้อมูลแบบลำดับขั้นหรือโครงสร้างแบบลำดับขั้น (Hierarchical data model) วิธีการสร้างฐานข้อมูลลักษณะนี้ ได้รับความนิยมมาก ในการพัฒนาฐานข้อมูลบนเครื่องคอมพิวเตอร์ขนาดใหญ่และขนาดกลาง โดยที่โครงสร้างข้อมูลจะสร้างรูปแบบเหมือนต้นไม้ (Tree) โดยความสัมพันธ์เป็นแบบ One- to –Many

2.5.2.2) รูปแบบข้อมูลแบบเครือข่าย (Network data Model) ฐานข้อมูลแบบเครือข่ายมีความคล้ายคลึงกับฐานข้อมูลแบบลำดับชั้น ต่างกันที่โครงสร้างแบบเครือข่าย จะมีการติดต่อแบบ Many-to-one หรือ Many-to-many กล่าวคือ Child node อาจมี Parent มากกว่าหนึ่ง สำหรับตัวอย่างฐานข้อมูลแบบเครือข่าย คือ การข้อมูลของห้องสมุด ซึ่งรายการจะประกอบด้วย ชื่อเรื่อง ผู้แต่ง สำนักพิมพ์ ที่อยู่ ประเภท

2.5.2.3) รูปแบบความสัมพันธ์ข้อมูล (Relation data model) เป็นลักษณะการออกแบบฐานข้อมูล โดยจัดข้อมูลให้อยู่ในรูปของตารางที่มีระบบคล้ายแฟ้ม โดยที่ข้อมูลแต่ละแถว จะแทนเรคอร์ด (Record) ส่วนข้อมูลในแนวดิ่งจะแทนคอลัมน์ (Column) ซึ่งเป็นขอบเขตของข้อมูล โดยที่ตารางแต่ละตารางที่สร้างขึ้น จะเป็นอิสระ ดังนั้นผู้ออกแบบฐานข้อมูลต้องมีการวางแผนถึงตารางข้อมูลที่จำเป็นต้องใช้ เช่น ระบบฐานข้อมูลบริษัทแห่งหนึ่ง ประกอบด้วย ตารางประวัติพนักงาน ตารางแผนกและตารางข้อมูลโครงการ แสดงประวัติพนักงาน ตารางแผนก และตารางข้อมูลโครงการ เป็นต้น

2.5.3 การออกแบบฐานข้อมูลเชิงสัมพันธ์

การออกแบบฐานข้อมูลในองค์กรขนาดเล็ก เพื่อตอบสนองความต้องการของผู้ใช้งานอาจเป็นเรื่องที่ไม่ยุ่งยาก เนื่องจากระบบและขั้นตอนการทำงานภายในองค์กรไม่ซับซ้อน ปริมาณข้อมูลที่มีก็ไม่มาก และจำนวนผู้ใช้งานฐานข้อมูลมีไม่มาก หากทว่าในองค์กรขนาดใหญ่ ซึ่งมีระบบและขั้นตอนการทำงานที่ซับซ้อน รวมทั้งมีปริมาณข้อมูลและผู้ใช้งานจำนวนมาก การออกแบบฐานข้อมูลจะเป็นเรื่องที่มีความละเอียดซับซ้อน และต้องใช้เวลาในการดำเนินการนาน ทั้งนี้ ฐานข้อมูลที่ได้รับการออกแบบอย่างเหมาะสมจะสามารถตอบสนองต่อความต้องการของผู้ใช้งานภายในหน่วยงานต่าง ๆ ขององค์กรได้ ซึ่งจะทำให้การดำเนินงานขององค์กรมีประสิทธิภาพดียิ่งขึ้น

Relational Database เป็นกลุ่มของข้อมูลที่ถูกเขียนอยู่ในรูปแบบของตาราง ซึ่งจะแบ่งออกเป็นข้อมูลที่อยู่ในลักษณะแนวนอน (Row) เรียกว่า Record และ ข้อมูลที่อยู่ในลักษณะแนวตั้ง (Column) เรียกว่า Attribute ซึ่งแต่ละตารางจะมีความสัมพันธ์ซึ่งกันและกัน ซึ่งมีส่วนประกอบดังนี้

1 .Entity เป็นสิ่งที่อ้างอิงถึงสิ่งต่างๆ ที่เราสนใจไม่ว่าจะเป็น บุคคล สถานที่ สิ่งของ ซึ่งแต่ละความสนใจจะถูกแยกออกเป็น Attribute ต่างๆที่เป็นรายละเอียดของสิ่งนั้นๆ

2. Attribute เป็นข้อมูลที่แสดงรายละเอียดของ Entity เช่น Attribute ของลูกค้าที่ประกอบไปด้วย ชื่อ ที่อยู่ เพศ เบอร์โทร และอื่นๆ ซึ่งเป็นการบอกรายละเอียดของลูกค้าคนนั้นๆ

3. Record เป็นการนำ Attribute หลายๆ Attribute มารวมกันเพื่อเป็นข้อมูลที่เจาะจงถึงสิ่งใดสิ่งหนึ่ง เช่น ข้อมูลลูกค้าก็ประกอบไปด้วย Attribute ชื่อ ที่อยู่ เพศ เบอร์โทร ซึ่งจะเป็นสิ่งที่เฉพาะของลูกค้าคนนั้น

4. Table เป็นการนำ Record หลายๆ Record มารวมกันเป็น Table เช่น การที่มีลูกค้าหลายๆคน ก็จะนำมารวมกันเป็น Table เดียวกัน

5. Relationships เป็นความสัมพันธ์ของ Entity ซึ่งอยู่ในลักษณะต่างๆ เช่น ลูกค้าสามารถสั่งซื้อสินค้าได้หลายชนิด พนักงานหลายคนสามารถเพิ่มสินค้าได้หลายชนิด ซึ่งความสัมพันธ์จะสามารถแบ่งออกได้ 3 ชนิด ดังนี้

One To One Relationships เป็นความสัมพันธ์ที่หนึ่ง Record ในตารางสามารถมีความสัมพันธ์กับอีกหนึ่ง Record ในอีกตารางเท่านั้น ตารางสินค้ากับตารางจำนวนสินค้า

One To Many Relationships เป็นความสัมพันธ์ที่หนึ่ง Record ในตารางสามารถมีความสัมพันธ์กับอีกหนึ่งหลาย Record ในอีกตาราง เช่น ลูกค้าสามารถที่จะมีใบสั่งซื้อสินค้าได้หลายใบ แต่ใบสั่งซื้อสินค้าสามารถที่จะมีลูกค้าได้แค่เพียงคนเดียว

Many To Many Relationships เป็นความสัมพันธ์หลาย Record ในตารางสามารถมีความสัมพันธ์กับอีกหนึ่งหลาย Record ในอีกตาราง เช่น ใบสั่งซื้อสินค้ามีสินค้าได้หลายชนิด และสินค้าชนิดนั้นสามารถมีได้ในหลายใบสั่งซื้อ

2.4.4 การออกแบบฐานข้อมูลในระดับตรรกะ

การออกแบบฐานข้อมูลในระดับตรรกะ หรือในระดับแนวความคิด เป็นขั้นตอนการออกแบบความสัมพันธ์ระหว่างข้อมูลในระบบโดยใช้แบบจำลองข้อมูลเชิงสัมพันธ์ ซึ่งอธิบายโดยใช้แผนภาพแสดงความสัมพันธ์ระหว่างข้อมูล (E-R Diagram) จากแผนภาพ E-R Diagram นำมาสร้างเป็นตารางข้อมูล (Mapping E-R Diagram to Relation) และใช้ทฤษฎีการ Normalization เพื่อเป็นการรับประกันว่าข้อมูลมีความซ้ำซ้อนกันน้อยที่สุด

2.6 ภาษาสืบค้น (Query Language)

การจัดเก็บข้อมูลสามารถจัดเก็บได้หลายรูปแบบ ซึ่งแต่ละรูปแบบมีภาษาที่ใช้ในการสืบค้นข้อมูลที่แตกต่างกันออกไป ซึ่งการใช้งานฐานข้อมูลนั้น จะสามารถเลือกข้อมูลได้อย่างเหมาะสามจะขึ้นอยู่กับการเลือกรูปแบบการค้นหาที่แตกต่างกันออกไป ดังต่อไปนี้

2.6.1 ภาษาสืบค้นข้อมูลเชิงโครงสร้าง (Structure Query Language: SQL)

SQL หรือ Structure Query Language เป็นชุดคำสั่งที่ใช้จัดการข้อมูลในฐานข้อมูล ชุดคำสั่ง SQL นิยมใช้มากในระบบฐานข้อมูลซึ่งอยู่ในรูปแบบของตารางที่มีความสัมพันธ์กัน หรือที่เรียกว่า Relational Database ดังที่กล่าวในหัวข้อ 2.4.3 ชุดคำสั่งที่ใช้ในการจัดการกับฐานข้อมูลที่สำคัญๆ มีดังนี้

1. Select เป็นคำสั่งสำหรับการดึงข้อมูลจากฐานข้อมูล SQL มาแสดง โดยสามารถที่จะกำหนดเงื่อนไขสำหรับการแสดง โดยใช้คำสั่ง WHERE เพื่อดึงข้อมูลเฉพาะที่ต้องการใช้งาน ซึ่งมีรูปแบบดังนี้

Select *คอลัมน์ที่เลือก* From *ตารางที่ต้องการอ้างถึง* Where *เงื่อนไขสำหรับการเลือกแสดงผล*

2. Update เป็นคำสั่งสำหรับการแก้ไขข้อมูลจากฐานข้อมูล ซึ่งจะเป็นการระบุเป็นคอลัมน์ที่ต้องการแก้ไข ซึ่งอาจจะมีเงื่อนไขสำหรับข้อมูลที่ต้องการจะแก้ไขโดยใช้คำสั่ง WHERE ซึ่งมีรูปแบบดังนี้

Update *ตารางที่อ้างถึง* Set [*ชื่อคอลัมน์=ข้อมูลที่ต้องการแก้ไข*] Where *เงื่อนไขสำหรับการเลือกแก้ไข*

3. Delete เป็นคำสั่งสำหรับการลบข้อมูลซึ่งเป็นการลบออกทีละแถว (Row) ซึ่งจะมีการระบุเงื่อนไขในการลบโดยคำสั่ง WHERE ซึ่งมีรูปแบบดังนี้

Delete From *ชื่อตารางที่อ้างถึง* Where *เงื่อนไขสำหรับการลบ*

4. Insert เป็นคำสั่งสำหรับการเพิ่งข้อมูลลงตารางโดยการเพิ่มทีละแถว (Row) ซึ่งสามารถที่จะระบุคอลัมน์ที่ต้องการที่จะเพื่อข้อมูลในแถวๆนั้นได้ ซึ่งมีรูปแบบดังนี้

Insert Into *ชื่อตารางที่อ้างถึง* [*ชื่อคอลัมน์ที่ต้องการเพิ่มข้อมูลในแถวนั้น*] Values [*ค่าที่ต้องการใส่ให้คอลัมน์นั้น*]

2.6.2 เอกซ์พาท (XPATH)

XPATH เป็นภาษาที่ใช้งานร่วมกับข้อมูล XML เพื่อใช้ในการดึงข้อมูลออกจากเอกสาร XML โดยการระบุ path ที่ต้องการแสดง สามารถระบุเงื่อนไขตามที่ต้องการได้ ตัวอย่างเช่น

"Name" เป็นการเข้าถึง node ของ Name โดยจะทำการแสดงทุกๆ child node ของ Name

"/Name" เป็นการเข้าถึง node ของ Name ที่อยู่ถัดจาก root node

"//Name" เป็นการเข้าถึง node ของ Name ทุกๆตัวในไฟล์ XML

"@age" จะเป็นการเข้าถึง node ต่างๆที่มี attribute ที่ชื่อว่า cost

การใช้เงื่อนไขในการระบุ path

"//Name [@age>21]" เป็นการเข้าถึง node Name ที่มี attribute ของ age มากกว่า 21

2.6.3 เอกซ์ควิรี (XQuery)

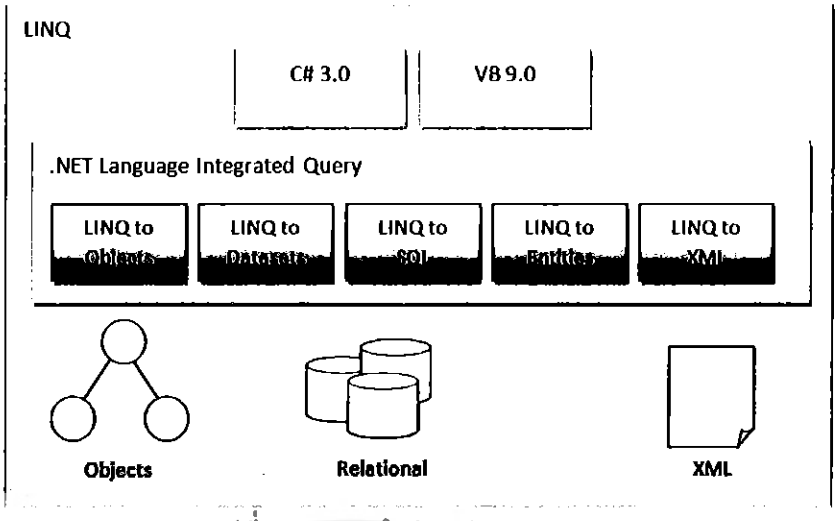
XQuery นั้นถูกพัฒนาโดยมีพื้นฐานมาจากภาษา SQL ซึ่งใช้งานได้ง่ายและเป็นที่ยอมรับจึงทำให้ XQuery สามารถทำความเข้าใจและใช้งานได้ง่ายเช่นเดียวกัน โดย XQuery เป็นภาษาสืบค้นสำหรับ XML ในรุ่นหลัง ซึ่งได้รับอิทธิพลมาจากภาษาสืบค้นสำหรับ XML ในรุ่นแรกๆ และมาตรฐานอื่นๆ ที่เกี่ยวข้อง ซึ่ง XQuery จะใช้แนวคิดของ path expression ในการระบุไปยังข้อมูลตำแหน่งต่างๆ ในเอกสารที่กำลังสนใจในโครงสร้างที่ซับซ้อนของเอกสาร โดย path expression นั้นจะอ้างอิงจากมาตรฐานของ XPath

โครงสร้างการทำงานของ XQuery จะประกอบไปด้วยลำดับของประโยค FOR - LET - WHERE - RETURN ซึ่งได้รับอิทธิพลมาจากภาษา SQL ที่เป็นลำดับประโยค SELECT - FROM – WHERE โดยประโยค FOR จะระบุไปยังข้อมูลที่กำลังสนใจโดยอาศัย path expression และนำข้อมูลนั้นมากำหนดให้กับตัวแปร ในแต่ละรอบของการทำงาน

2.6.4 Language Integrated Query (LINQ) [4]

LINQ หรือ Language Integrated Query เป็นภาษาที่ไมโครซอฟต์สร้างขึ้น เพื่อช่วยให้การสืบค้นข้อมูล ซึ่งอยู่ในรูปแบบที่หลากหลาย เป็นภาษาที่ใช้ดึงข้อมูลจากแหล่งข้อมูลต่างๆ

ข้อมูลถูกจัดเก็บอยู่ในรูปแบบที่หลากหลาย เช่น Text File, XML File หรือข้อมูลที่อยู่ในรูปแบบของตาราง การที่เราจะต้องแยกภาษาสำหรับการดึงข้อมูลนั้น เป็นสิ่งที่ไม่สะดวกเพราะผู้ใช้งานจะต้องเรียนรู้ในหลายๆภาษา ไมโครซอฟต์พัฒนา LINQ ขึ้นเพื่อทำให้การสืบค้นข้อมูลนั้นทำได้กับข้อมูลในหลายชนิดแต่ให้เพียงภาษาเดียว LINQ ไม่ได้เข้ามาทำหน้าที่แทนภาษา SQL แต่ LINQ จะทำหน้าที่เข้ามาเสริมเติมเต็มส่วนที่ SQL ไม่สามารถทำได้



รูปที่ 2.9 แสดงถึงส่วนประกอบ LINQ

2.6.4.1) LINQ to Object ทำหน้าที่สอบถามข้อมูลที่เก็บอยู่ในรูปแบบของ Object ต่างๆ เช่น เก็บใน Array, ใน Class ต่างๆ เป็นต้น ไม่มีความซับซ้อน

2.6.4.2) LINQ to SQL, LINQ to Datasets, LINQ to Entities ทำหน้าที่สอบถามข้อมูลที่เก็บในรูปแบบด้านฐานข้อมูล ใช้ในการติดต่อกับ Relational Database

2.6.4.3) LINQ to XML ใช้ในการติดต่อกับ XML Document ทำหน้าที่สืบค้นข้อมูลที่เก็บอยู่ในรูปแบบของ XML โดยปกติแล้วในการสืบค้นข้อมูลจากไฟล์ XML จะต้องใช้ภาษามาตรฐานซึ่งจะเป็นการระบุ Path และเงื่อนไขข้อมูลที่ต้องการ แต่สำหรับ LINQ to XML สามารถที่จะสืบค้นข้อมูลหรือทำการแก้ไขข้อมูลได้โดยไม่จำเป็นต้องใช้ XPATH

2.7 Ontology

Ontology เป็นการนิยามหรือกำหนดรูปแบบโครงสร้างของสิ่งที่สนใจให้มีความหมายตามขอบเขตขององค์ความรู้ ซึ่ง Ontology นี้ได้ถูกกล่าวถึงในหลายองค์กร โดยเฉพาะปัญญาประดิษฐ์นั้นมีการใช้งานมานานแล้ว โดยในปัจจุบันได้ถูกนำมาประยุกต์ใช้ในมาตรฐานของการออกแบบจำลองโครงสร้างของ eXtensible Markup Language (XML) และการนิยามรูปแบบแนวคิดของโครงสร้างของฐานข้อมูล

Ontology เป็นลักษณะภาษาที่นำมาใช้บรรยายโครงสร้างและความสัมพันธ์ของระบบผ่านโหนดแบบลำดับชั้น (Hierarchies) ในปัจจุบันได้กำหนดภาษามาตรฐานที่ใช้จำลองและออกแบบโครงสร้างของเอกสารเอ็กซ์เอ็มแอล (XML) โดยใช้นิยามแนวคิดให้อยู่ในรูปของกฎ (Role) คลาส (Class) ความสัมพันธ์ระหว่างคลาส (Relation) และคุณสมบัติของคลาส (Properties) แล้วนำเสนอออกมาในรูปของโหนด และความสัมพันธ์แบบลำดับชั้น [7]

ลักษณะของ Ontology คือโครงสร้างลำดับชั้นที่ใช้ในการจำแนกกลุ่มของวัตถุหรือสิ่งที่กำลังสนใจ ตัวอย่างเช่น Ontology การเกษตรอาจจะเริ่มจากผลิตผลการเกษตร

อย่างเช่น ผลิตผลการเกษตร [8]

|- พืช
| |- พืชยืนต้น
| | |- มะม่วง
| | |- มะม่วงแก้ว
| |- พืชล้มลุก
| |- ถั่ว
| |- ข้าว
|- สัตว์

ตัวอย่าง Ontology ข้างต้น อธิบายถึง การจำแนกกลุ่มของพืชที่เป็นผลิตผลการเกษตร ที่บอกถึงพืชประเภทต่างๆเช่น พืชยืนต้น พืชล้มลุก ที่สามารถอธิบาย กลุ่มย่อยที่ได้ออกไปอย่างต่อเนื่องอีก

```
<owl:Class rdf:ID="Wine">
    <rdfs:subClassOf
rdf:resource="&food;PotableLiquid"/>
    <rdfs:subClassOf>
        <owl:Restriction>
        <owl:onProperty
rdf:resource="#madeFromGrape"/>
        <owl:minCardinality
        df:datatype="&xsd;nonNegativeInteger">1</owl:m
        inCardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
        ...
</owl:Class>
```

ตัวอย่าง ของ Ontology ข้างต้นอธิบายถึง ความสัมพันธ์ของกลุ่มอาหารที่ดื่มได้ โดยมีไวน์เป็นซับคลาสของอาหารที่ดื่มได้ ที่มีความสัมพันธ์แบบ OWL แสดงด้วยภาษา XML

การนำ Ontology ไปใช้งานนั้น สามารถทำได้หลายอย่าง นอกจากการทำงานด้านคอมพิวเตอร์และสามารถสร้างประโยชน์อื่นๆมากมาย รวมถึงช่วยในการทำ Classification (จำแนกกลุ่ม) ในงานวิจัยด้านต่างๆด้วย

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

2.7.1 RDF

Resource Description Framework หรือ "RDF" เป็นมาตรฐานที่อิงมาจากภาษา XML แต่มีโครงสร้างรูปแบบหลากหลายกว่า XML ชื่อของ RDF บอกถึงกรอบในการกำหนดและแลกเปลี่ยนข้อมูล Metadata ซึ่งอยู่บนกฎเกณฑ์ดังต่อไปนี้

1. Resource แหล่งข้อมูลคือทุกอย่างที่มี URL มาเกี่ยวข้อง ซึ่งรวมทั้ง WWW แต่ละ Element ของข้อมูล XML ตัวอย่างเช่นระบุเป็น http://www.thaixml.com/RDF/draft.htm เป็นต้น

2. Property คือแหล่งข้อมูลที่มีชื่อเฉพาะและมีคุณสมบัติเป็น Property เช่น ผู้แต่ง หรือ Title

3. Statement ประกอบด้วย Resource Property และค่าของข้อมูล เช่น "ผู้แต่งของ http://www.thaixml.com/essentials/rdf.htm คือ John" เป็นต้น แต่ก็มีวิธีการตรงไปตรงมาในการนำเสนอในรูปแบบของ XML

RDF เป็นมาตรฐานที่อิงพื้นฐานมาจาก XML ใช้อธิบายแหล่งข้อมูลใดๆก็ได้ โดยมาตรฐานจะกำหนดชื่อ tag และโครงสร้างข้อมูล สำหรับอธิบายแหล่งข้อมูลในลักษณะต่างๆ เมื่อใช้มาตรฐานเดียวกันก็จะสะดวกต่อการดึงข้อมูลไปใช้ หรือแลกเปลี่ยนข้อมูลในระหว่างองค์กร

2.7.2 NAICS

NAICS เป็น Ontology แสดงกลุ่มของข้อมูลที่มีความสัมพันธ์กัน แสดงด้วย XMLที่มีความสัมพันธ์กันตามลำดับชั้น ความสัมพันธ์และรายละเอียดอธิบายด้วยหมายเลขกำกับ "NAICSCODE" และข้อความ "NAICSTEXT" อธิบายเป็นลำดับชั้นของข้อมูลตามความสัมพันธ์นั้นๆ นอกจากนี้ยังมีข้อความพิเศษ อธิบายเพิ่มเติมเกี่ยวกับข้อความ ที่มีแบบแผนอย่างเป็นทางการที่อธิบายลำดับชั้นและความสัมพันธ์ระหว่างแหล่งข้อมูลที่แตกต่างกัน ตัวอย่างเช่น

```
      <z:row NAICSCODE='11'
NAICSTEXT='AGRICULTURE,FORESTRY, FISHING AND HUNTING'/>
      <z:row NAICSCODE='111' NAICSTEXT='CROP PRODUCTION'/>
      <z:row NAICSCODE='1111' NAICSTEXT='OILSEED AND GRAIN
FARMING'/>
      <z:row NAICSCODE='11111' NAICSTEXT='SOYBEAN FARMING'
SICCODE='0116' SICTEXT='SOYBEANS'/>
```

ตัวอย่าง NAICS นี้แสดงถึง เป็นความสัมพันธ์ของข้อมูลด้าน AGRICULTURE, FORESTRY, FISHING และ HUNTING ที่เป็นหัวข้อหลักที่แสดงด้วย NAICSCODE หมายเลข 11 ที่เป็นเลข 2 หลัก ที่แสดงว่ากลุ่มข้อมูลนี้จะมีข้อมูลที่เกี่ยวกับ AGRICULTURE, FORESTRY, FISHING และ HUNTING และจะมีข้อมูลที่อธิบายข้อมูลด้านนี้ที่ละเอียดและลึกลงอีก สามารถอธิบายความหมายและแยกข้อมูลที่มีความสัมพันธ์นี้ได้เพิ่มเติม เช่น

NAICSCODE='111' ที่อธิบายถึง CROP PRODUCTION

197 0995 7
ว.ร.
ก 2825
ว553

สามารถแยกกลุ่มที่มีความคล้ายคลึงของข้อมูลต่อไปได้อีก ที่ซึ่งมีความสัมพันธ์กันย่อยลงไปอีก เช่น

```
NAICSCODE='1111' NAICSTEXT='OILSEED AND GRAIN FARMING
```

ตัวเลขที่มากขึ้นแสดงให้เห็นว่า ข้อมูลนั้นๆมีรายละเอียดปีกย่อยได้อีกเรื่อยๆ และตัวเลขที่เหมือนกันแสดงว่าข้อมูลอยู่ในกลุ่มเดียวกันนั่นเอง

2.7.3 OWL

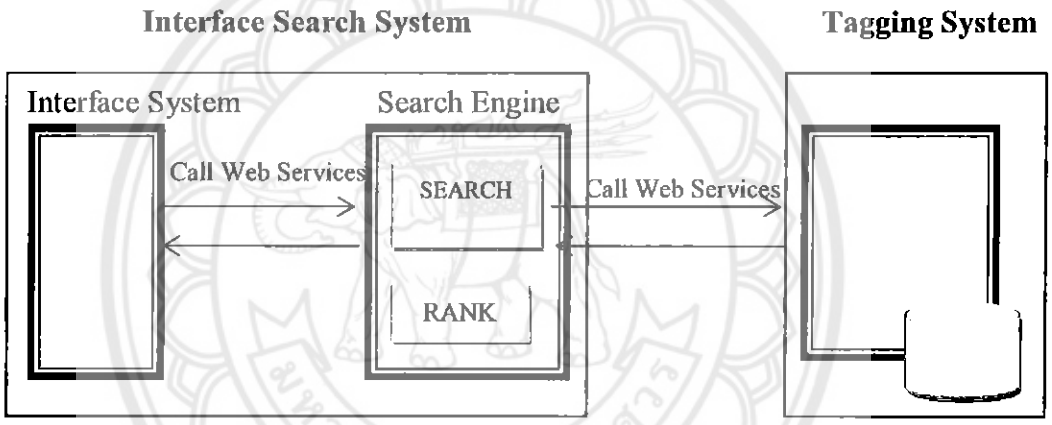
Web Ontology Language หรือ "OWL" ถูกสร้างโดย W3C Web Ontology Working Group (WebOnt) โอดับบลิวแอลถูกพัฒนาขึ้นเพื่อเป็นส่วนขยายต่อจากภาษาอาร์ดีเอฟ (RDF) และสืบทอดมาจากภาษาดีเอเอ็มแอล พลัส ออยล์ (DAML+OIL) ภาษา OWL จัดเป็นองค์ประกอบหนึ่งในงานเว็บเชิงความหมาย (Semantic Web) ที่ใช้ในการบรรยายข้อมูลเชิงความหมาย สามารถกำหนดโครงสร้างข้อมูลในลักษณะลำดับชั้น และอธิบายข้อมูล (Metadata) ที่มีความสัมพันธ์ในระบบฐานข้อมูลได้ รวมทั้งสามารถรองรับการบรรยายข้อมูลเชิงตรรกะ ชนิดข้อมูลและตัวบ่งปริมาณได้ ทำให้ข้อมูลที่ถูกแทนที่นั้นมีความหมายมากยิ่งขึ้น ลักษณะการบรรยายจะอยู่ในรูปของคลาส คุณสมบัติของคลาส และความสัมพันธ์ของคลาส เพื่ออธิบาย entity และความสัมพันธ์ต่าง ๆ ที่เกิดขึ้น[8]

OWL เป็นภาษาที่รวมกันระหว่างข้อความหรือ "Text" และ ข้อความพิเศษ หรือ "Extra Information" ที่เพิ่มเติมเข้ามาเกี่ยวกับข้อความ ที่มีแบบแผนอย่างเป็นทางการที่อธิบายลำดับชั้นและความสัมพันธ์ระหว่างแหล่งข้อมูลที่แตกต่างกัน สร้างอยู่บน RDF และ RDFS ซึ่งประกอบด้วยอนุกรมวิธาน หรือ "Taxonomy" และเซทของกฎที่ได้จากเครื่องที่สามารถสร้างข้อสรุปแบบเชิงตรรกะ หรือ "Logical"

บทที่ 3
การออกแบบระบบ

จากการศึกษาค้นคว้าข้อมูลและทฤษฎีข้างต้น ทำให้สามารถเข้าใจถึงหลักการและวิธีการในการดำเนินงาน ในบทนี้เป็นการนำความรู้ที่ได้ ประยุกต์ใช้กับการจัดการระบบการค้นหาข้อมูล โดยการเปรียบเทียบ Boolean Expression กับแท็กในฐานข้อมูล ผลที่ได้คือรายการของ URL ที่ผู้ใช้ต้องการ ซึ่งมีการจัดอันดับผลของการค้นหา (Ranking) ของข้อมูลที่ได้

3.1 การออกแบบระบบ (System Design)

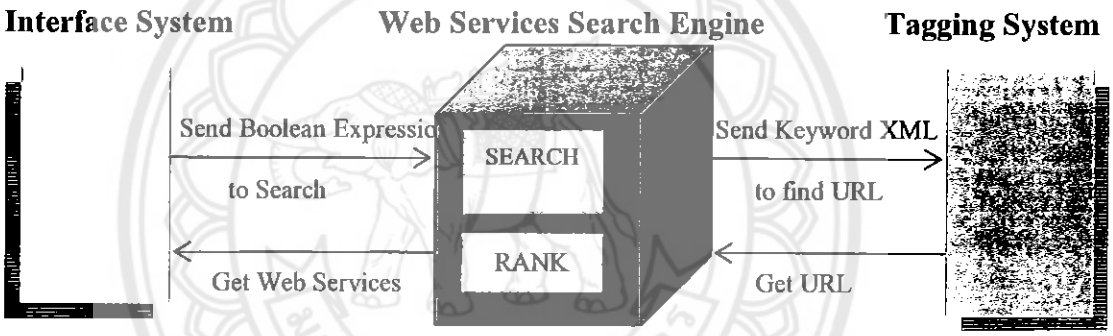


รูปที่ 3.1 แสดงการทำงานของระบบการค้นหาเว็บเซอร์วิส

ระบบการค้นหาเว็บเซอร์วิส (Web services Search Engine) ออกแบบให้มีการทำงานแบบเว็บเซอร์วิสที่สามารถสื่อสารกันด้วย XML ระบบการค้นหาเว็บเซอร์วิสนี้มีการทำงานเชื่อมต่อกับ 2 ระบบการทำงาน ประกอบด้วย ส่วนระบบการค้นหาข้อมูล (Interface Search System) และระบบฐานข้อมูลแท็ก (Tagging System) ซึ่งระบบการค้นหาเว็บเซอร์วิสนี้เป็นส่วนหนึ่งของระบบการค้นหาข้อมูลเป็นส่วนในการค้นหา(Interface Search System) มีการเชื่อมต่อในหน้าสำหรับการค้นหาและระบบข้อมูลแท็กในการค้นหา URL ที่ผู้ใช้ต้องการ ซึ่งมีการออกแบบการทำงานให้สามารถเชื่อมต่อกันในการทำงานได้ ให้ระบบมีประสิทธิภาพในการค้นหาเว็บเซอร์วิสที่ต้องการมีหลักการ ดังรูป 3.1

3.1.1 การทำงานของระบบ (Conceptual Design)

ระบบการค้นหาเว็บเซอร์วิส (Web Services Search Engine) ออกแบบให้สามารถรับ Boolean Expression ที่ได้มาจากการวิเคราะห์การใส่คำค้นหาในส่วนหน้าของหน้าการค้นหา ในส่วนแสดงผล(Interface System) แล้วเปรียบเทียบข้อมูลและเงื่อนไขในประโยคสืบค้น Boolean Expression กับฐานข้อมูลแท็ก ซึ่งเป็นแท็กที่ได้มาจากผู้ให้บริการ(Provider) ที่กำหนดขึ้นจาก NAICS Ontology และแท็กที่กำหนดเองโดยผู้ใช้ (User Defined Tag) จากส่วนระบบฐานข้อมูลแท็ก (Tagging System) ผลที่ได้จากการเปรียบเทียบคือ รายการของการค้นหา URL ของเว็บเซอร์วิส จากนั้นระบบจะทำการวิเคราะห์ผลการค้นหาและทำการจัดอันดับผลของการค้นหา (Ranking) ของบริการของเว็บเซอร์วิสใน Ontology ที่เซอร์วิสนั้นๆ เกี่ยวข้องอยู่ แสดงดังรูป 3.1 ซึ่งมีการแบ่งทำงานออกเป็น 3 ส่วน ดังต่อไปนี้



รูปที่ 3.2 แสดง Conceptual Design

1) Tagging System คือ ส่วนฐานข้อมูลแท็ก เป็นส่วนที่ทำหน้าแท็กข้อมูลของบริการต่างๆ เป็นการกำหนดข้อมูลและรายละเอียดของแต่ละบริการ เพื่อง่ายต่อการทำการค้นหา ระบบนี้ทำการแท็กข้อมูลและเก็บลงในฐานข้อมูลของบริการ เพื่อใช้ในการค้นหา URL ที่ต้องการ ซึ่งแท็กได้มาจากผู้ให้บริการ (Provider) ที่กำหนดขึ้นจาก NAICS Ontology และแท็กที่กำหนดเองโดยผู้ใช้ (User Defined Tag) ระบบฐานข้อมูลแท็ก (Tagging System) ไม่เกี่ยวเนื่องในระบบการค้นหานี้ เป็นระบบงานของระบบการแท็กข้อมูลโดยอาศัยหลักการของออนโทโลจี

2) Interface System คือ ส่วนแสดงผล เป็นส่วนที่ทำหน้าติดต่อกับผู้ใช้โดยตรง โดยผู้ใช้เป็นผู้ใส่คำค้นหาในส่วนหน้าการค้นหา (User Interface) แล้วทำการเลือกข้อมูลที่มีความสัมพันธ์กับคำค้นหา ผู้ใช้มีการเลือกแนวทางความเป็นไปได้ของคำค้นหา ที่ผู้ให้บริการ (Provider) แนะนำเพื่อสร้างประโยค Boolean Expression เพื่อใช้ในการค้นหาข้อมูลเว็บเซอร์วิส ระบบการแสดงผล

(Interface System) ไม่เกี่ยวเนื่องในระบบการค้นหานี้ เป็นระบบงานของระบบการค้นหาเว็บเซอร์วิสแบบอินเตอร์แอกทีฟ

3) Search System ส่วนของการค้นหาข้อมูลเว็บเซอร์วิสที่ต้องการ เป็นส่วนของการทำงานในระบบการค้นหานี้ ประกอบด้วยระบบ Search และ Rank สำหรับทำการค้นหาข้อมูลตามคำค้นหาที่ได้จากส่วนแสดงผล (Interface System) และมีการเชื่อมต่อในการค้นหาข้อมูล URL ที่ต้องการในฐานข้อมูลของระบบฐานข้อมูลแท็ก (Tagging System) แสดงรายละเอียดการทำงานของระบบทั้งสอง ดังนี้

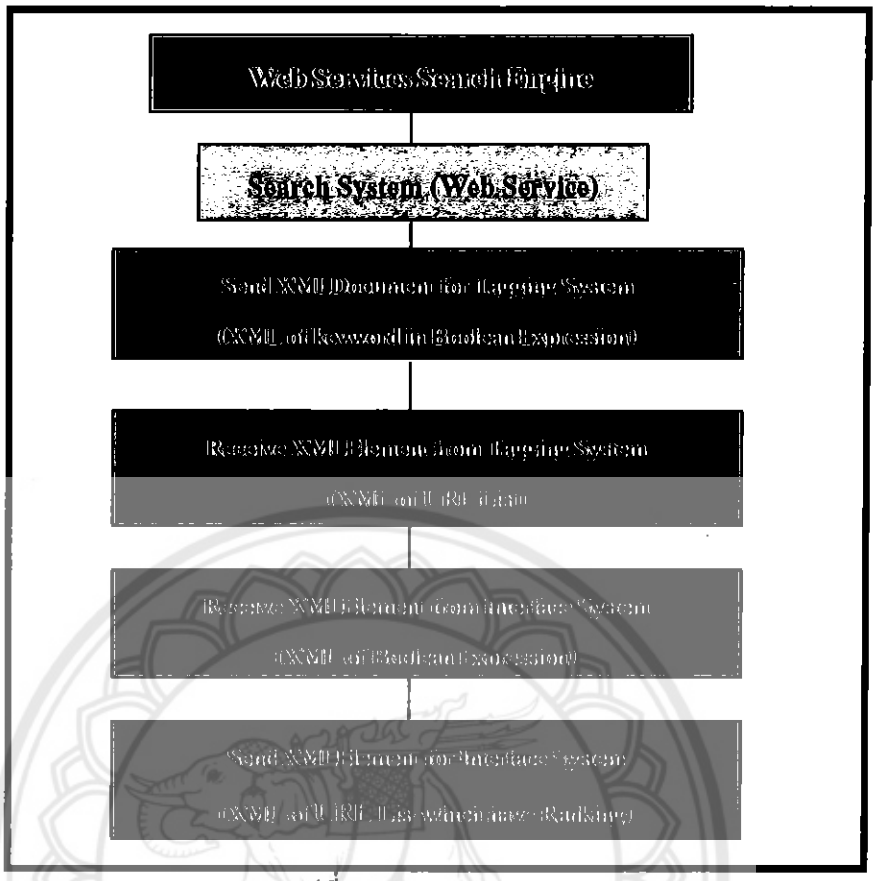
3.1) Search ส่วนของการเปรียบเทียบข้อมูลระหว่างข้อมูลจากฐานข้อมูลและเงื่อนไขของ Boolean Expression ที่มาจากส่วนแสดงผล (Interface System) และแท็กของข้อมูล URL ในฐานข้อมูลที่มีในส่วนฐานข้อมูลแท็ก (Tagging System) เพื่อหาผลลัพธ์เว็บเซอร์วิสที่ผู้ใช้ต้องการแสดงเป็นรายการของ URL โดยยึดความสัมพันธ์ตามเงื่อนไขของ Boolean Expression ซึ่งเป็นส่วนหนึ่งของการทำงานในระบบการค้นหานี้

3.2) Rank ส่วนของการจัดอันดับผลการค้นหา ด้วยการกำหนดค่าน้ำหนักให้กับแต่ละ URL ที่ค้นพบ โดยการพิจารณาตามเงื่อนไขของประโยค Boolean Expression และทำการจัดอันดับผลของการค้นหาตามค่าน้ำหนัก สร้างไฟล์ XML ของรายการ URL ที่ได้จากการค้นหาเพื่อส่งให้ส่วนแสดงผล (Interface System) แสดงข้อมูลแก่ผู้ใช้ ส่วนนี้เป็นส่วนหนึ่งของการทำงานในระบบการค้นหานี้

3.1.2 โครงสร้างของโปรแกรม (Program Structure)

โครงสร้างของโปรแกรม (Program Structure) ในระบบการค้นหาเว็บเซอร์วิส จะแสดงการทำงานของระบบที่เป็นตัวกลางในการสื่อสารและค้นหาข้อมูลระหว่าง 2 ระบบ เพื่อค้นหาบริการเว็บเซอร์วิสที่ผู้ใช้ต้องการ ประกอบด้วย ระบบการแสดงผล (Interface System) และระบบฐานข้อมูลแท็ก (Tagging System) ซึ่งมีการทำงานเกี่ยวเนื่องกันในการค้นหาเว็บเซอร์วิส

ระบบทั้งหมดจะทำงานสื่อสารกัน เป็นการทำงานแบบเว็บเซอร์วิส ที่มีการสื่อสารและแลกเปลี่ยนข้อมูลด้วย XML ระบบการแสดงผล (Interface System) และระบบฐานข้อมูลแท็ก (Tagging System) สามารถเรียกใช้งานระบบการค้นหาได้รวมถึงการรับและส่งข้อมูลที่ต้องการในการทำงานของระบบของแต่ละระบบ แสดงดังรูปที่ 3.3



รูปที่ 3.3 Program Structure

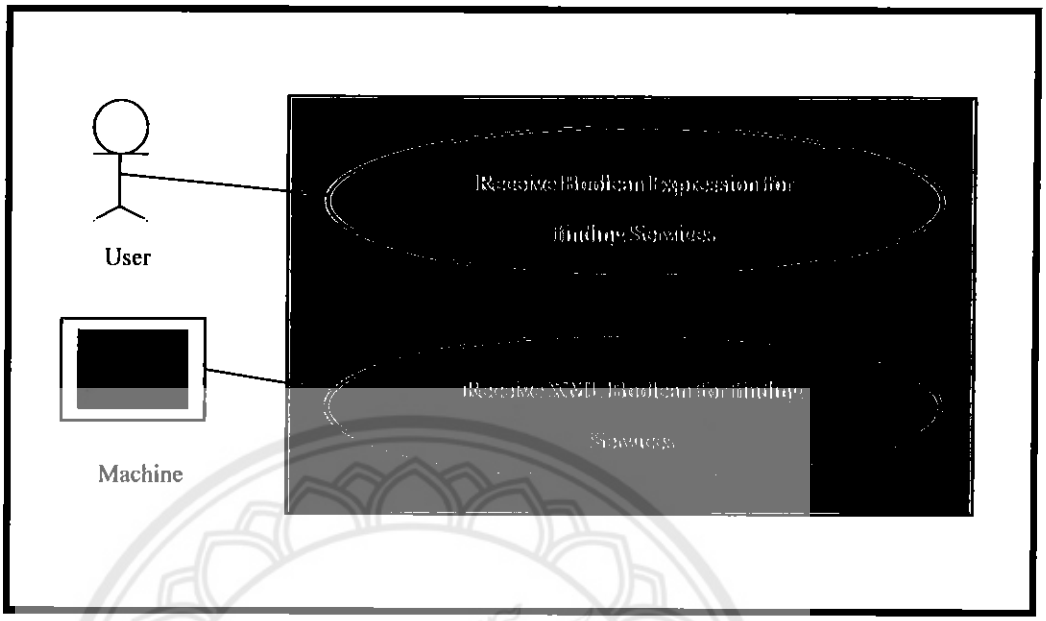
3.1.3 Interface Chart

ในระบบการค้นหาเว็บเซอร์วิสนี้ ทำงานสื่อสารระหว่างระบบอีก 2 ระบบ คือ ส่วนระบบการแสดงผล (Interface System) และส่วนระบบฐานข้อมูลแท็ก (Tagging System) ที่สามารถเรียกใช้งานระหว่างกันในการทำการค้นหาข้อมูล มีการเรียกใช้งานของระบบต่างๆกับระบบการค้นหานี้ ทำงานได้ในหลายส่วน พิจารณาข้อมูล Interface Chart ดังตารางที่ 3.3

ตารางที่ 3.1 ข้อมูล Interface Chart

Path of Searching		
Send XML of Boolean Expression	/	
Receive XML of Boolean Expression		/
Send XML of URL		/
Receive XML of URL List which has Ranking	/	

3.1.4 Use Case Diagrams



รูปที่ 3.4 Use Case Diagram ของระบบ

รูปที่ 3.4 แสดง Use Case Diagram ของระบบ เป็นแผนภาพของระบบที่ผู้ใช้และระบบอื่นๆ สามารถเรียกใช้งานระบบการค้นหานี้ได้ ในระบบการค้นหานี้ ผู้ใช้และMachine อื่นๆ สามารถเรียกใช้งานได้ด้วยการสื่อสารด้วยไฟล์ XML

Machine อื่นๆสามารถใช้ไฟล์ XML ของ Boolean Expression เพื่อสื่อสารข้อมูลในการค้นหาข้อมูลเพื่อค้นหาข้อมูลเว็บเซอร์วิสที่ต้องการ

ส่วนผู้ใช้สามารถใส่ Boolean Expression ผ่านทางระบบแสดงผล (Interface System) ในการค้นหาข้อมูลเว็บเซอร์วิสตามความต้องการ และเมื่อระบบค้นหาเสร็จจะได้ข้อมูลเว็บเซอร์วิสที่ต้องการแสดงผลทางระบบการแสดงผล (Interface System)

3.1.5 Use Case Descriptions

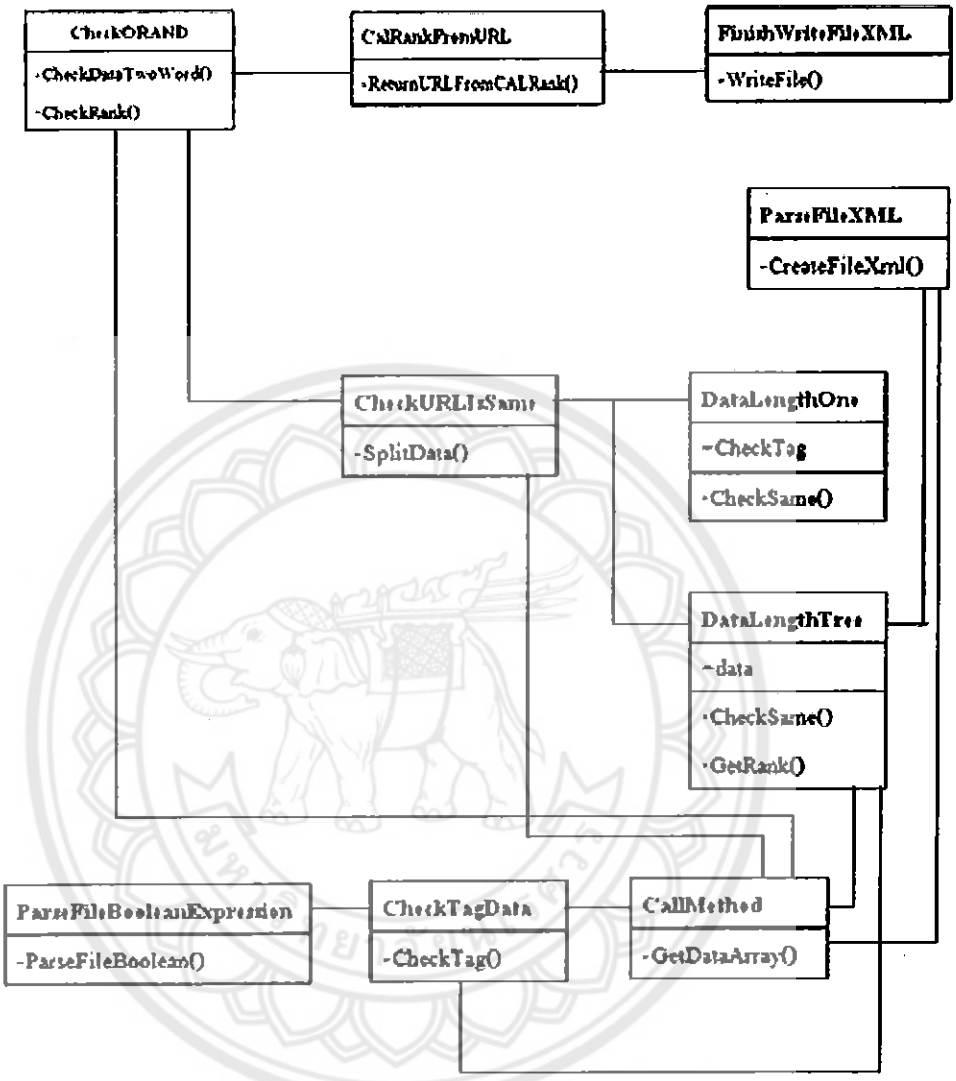
ตารางที่ 3.2 Use Case Description ของผู้ใช้

1. **Name :** รับ Boolean Expression จากหน้าการค้นหาเว็บเซอร์วิสสำหรับค้นหาเว็บเซอร์วิส
2. **Participating Actor :** User (ผู้ใช้)
3. **Entry Condition :** มีประโยค Boolean Expression ที่ได้จากคำค้นหาของผู้ใช้ที่ถูกต้อง
4. **Exit Condition :** ได้รายการของเว็บเซอร์วิสตามเงื่อนไขของคำค้นหาที่ต้องการ
5. **Flow of Even** - ผู้ใช้กรอกคำค้นหาในหน้าค้นหาเว็บเซอร์วิสในส่วนระบบการแสดงผลและระบบแสดงผลทำการสร้าง Boolean Expression - ระบบรับ XML ของ Boolean Expression จากส่วนระบบการแสดงผล วิเคราะห์เงื่อนไขของการค้นหาและทำการค้นหาข้อมูลในส่วนระบบฐานข้อมูลแท็ก วิเคราะห์ข้อมูลและเรียงลำดับผลของการค้นหา และส่งข้อมูลเพื่อการแสดงผล - ผู้ใช้ได้รับรายการของเว็บเซอร์วิสตามเงื่อนไขของคำค้นหาที่ต้องการ ที่มีการเรียงลำดับการค้นหาแล้ว

ตารางที่ 3.3 Use Case Description ของ Machine

1. **Name :** รับ XML Boolean Expression สำหรับค้นหาเว็บเซอร์วิส
2. **Participating Actor :** Machine (Interface System)
3. **Entry Condition :** มี XML ของประโยค Boolean Expression ที่ถูกต้อง
4. **Exit Condition :** ได้ไฟล์ XML ของของเว็บเซอร์วิส ตามเงื่อนไขของBoolean Expression
5. **Flow of Even** - ระบบรับ XML ของ Boolean Expression จากส่วนระบบการแสดงผล วิเคราะห์เงื่อนไขของการค้นหาและทำการค้นหาข้อมูลในส่วนระบบฐานข้อมูลแท็ก ค้นหาข้อมูลและเรียงลำดับผลของการค้นหา - ส่ง XML ของรายการเว็บเซอร์วิส ตามเงื่อนไขของคำค้นหาที่ต้องการ ที่มีการเรียงลำดับการค้นหาแล้ว

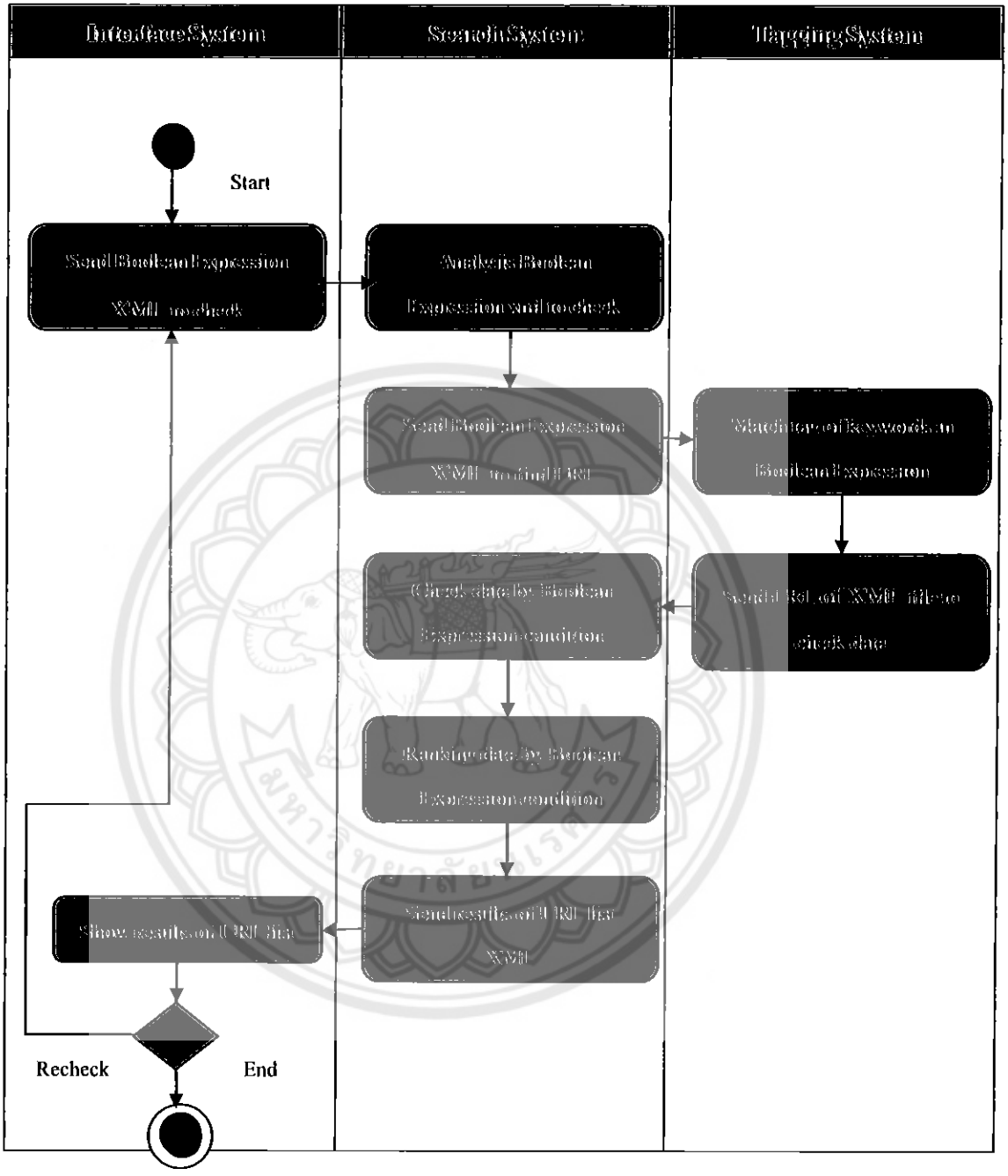
3.1.6 Class Diagram



รูปที่ 3.5 Class Diagram ของระบบ

รูปที่ 3.5 แสดง Class Diagram ของระบบ เป็นแผนภาพของระบบที่ทำงานประสานกันในการค้นหาเว็บเซอร์วิส และมีการเรียกใช้งานเว็บเซอร์วิสอื่นๆในการทำงานร่วมกัน

3.1.7 Activity Diagram



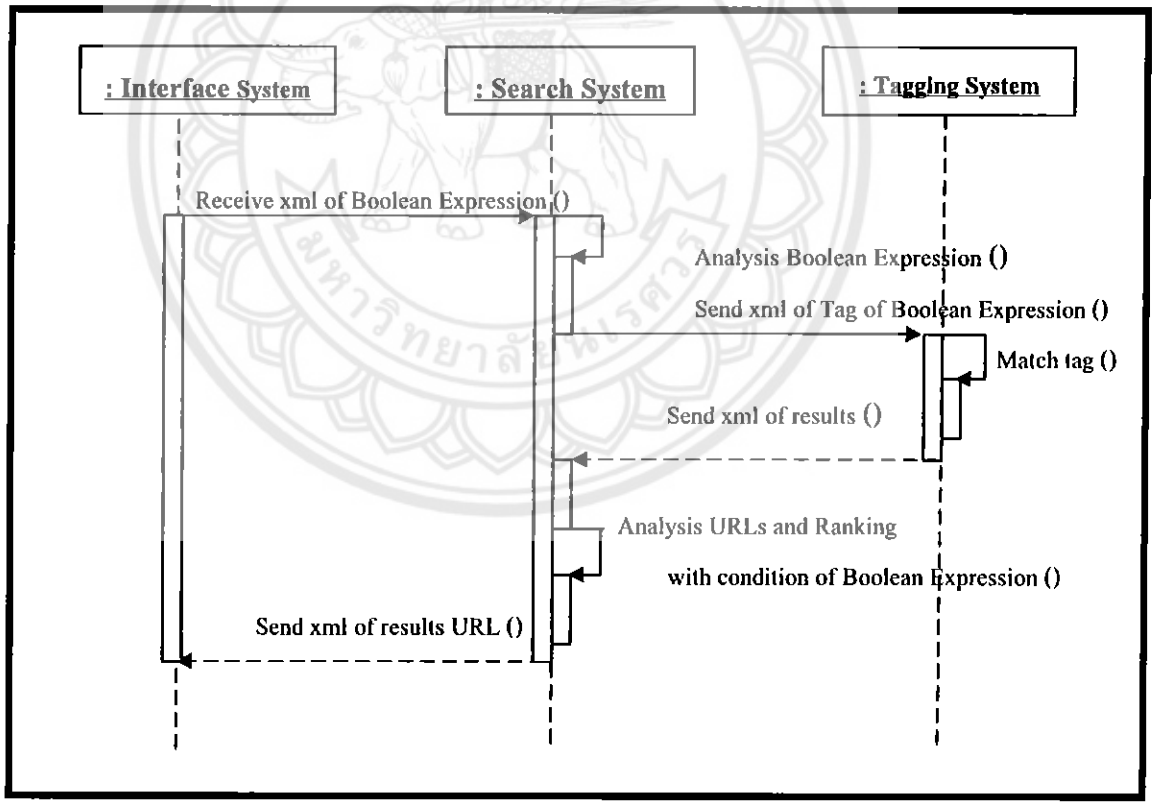
รูปที่ 3.6 Activity Diagram ของระบบ

จากรูปที่ 3.6 แสดง Activity Diagram แผนภาพแสดงการทำงานของระบบทั้งหมด ตั้งแต่เริ่มแรกในการทำงานของระบบการค้นหาเว็บเซอร์วิส มีการสื่อสารด้วยไฟล์ XML เนื่องจากเป็นการทำงานของระบบเว็บเซอร์วิส โดยระบบค้นหานี้เริ่มต้นการทำงานเมื่อมีการส่งไฟล์ XML ของ Boolean Expression จากส่วนระบบการแสดงผล (Interface System) จากนั้นทำการวิเคราะห์

ประโยคสืบค้น Boolean Expression ที่ได้ เพื่อเป็นเงื่อนไขในการค้นหาข้อมูล และส่งไฟล์ XML ที่ได้ของประโยคสืบค้น Boolean Expression ไปในส่วนระบบฐานข้อมูลแท็ก (Tagging System) เพื่อค้นหาข้อมูลที่ต้องการตามข้อมูลในประโยคสืบค้น Boolean Expression

จากนั้นเมื่อมีการค้นหาข้อมูลเรียบร้อยแล้ว ระบบการค้นหาจะรับข้อมูลที่ได้จากการค้นหาในฐานข้อมูลในส่วนระบบฐานข้อมูลแท็ก (Tagging System) กลับมาวิเคราะห์ พิจารณาเปรียบเทียบค้นหาข้อมูลจากเงื่อนไขในประโยค Boolean Expression กับข้อมูลที่ได้จากฐานข้อมูล เพื่อพิจารณาหา URL ที่เป็นไปได้ทั้งหมด ที่ถูกต้องตามเงื่อนไขของ Boolean Expression และกำหนดค่าน้ำหนักของ URL ทั้งหมดตามเงื่อนไขของ Boolean Expression จากนั้นระบบจะทำการจัดอันดับการค้นหา (Ranking) ของรายการ URL ที่มีตามค่าน้ำหนักของข้อมูล พิจารณาค่าน้ำหนักเหล่านั้น และสร้างไฟล์ XML ของผลการค้นหาที่ได้ เพื่อส่งให้แก่ส่วนระบบการแสดงผล (Interface System) เพื่อแสดงผลของการค้นหาบริการเว็บเซอร์วิสผู้ใช้

3.1.8 Sequence Diagram



รูปที่ 3.7 Sequence Diagram ของระบบ

จากรูปที่ 3.7 แสดง Sequence Diagram แผนภาพการทำงานของระบบการค้นหาเว็บเซอร์วิสที่สื่อสารด้วยไฟล์ XML โดยระบบค้นหานี้จะรับไฟล์ XML ของ Boolean Expression จาก

ส่วนระบบการแสดงผล (Interface System) แล้วทำการวิเคราะห์ประโยคสืบค้น Boolean Expression จากนั้นส่งไฟล์ XML ที่ได้ของประโยคสืบค้น Boolean Expression เพื่อค้นหาข้อมูลที่ต้องการในฐานข้อมูลในอีกส่วน ที่ทำงานเป็นส่วนระบบฐานข้อมูลแท็ก (Tagging System) จากนั้นรับข้อมูลที่ค้นหาได้มาวิเคราะห์ พิจารณาเปรียบเทียบค้นหาข้อมูลและกำหนดค่าน้ำหนักของข้อมูลด้วยเงื่อนไขในประโยค Boolean Expression เพื่อพิจารณาหา URL ที่ถูกต้องตามเงื่อนไข ระบบจะจัดอันดับผลที่ได้จากการค้นหา (Ranking) ของรายการ URL ที่มี จากนั้นสร้างไฟล์ XML ของผลการค้นหา เพื่อส่งให้แก่ส่วนระบบการแสดงผล (Interface System) เพื่อแสดงผลของการค้นหาบริการเว็บเซอร์วิสให้แก่ผู้ใช้

3.2 ข้อมูล (Data)

3.2.1 ฐานข้อมูลของระบบฐานข้อมูลแท็ก (Tagging System)

3.2.1.1) Database Design

ระบบฐานข้อมูลแท็ก (Tagging System) เป็นส่วนในการแท็กข้อมูลที่เกี่ยวบริการโดยจากผู้ให้บริการ ในส่วนของระบบฐานข้อมูลแท็ก (Tagging System) จะประกอบด้วยส่วนของหน้าของการแท็กข้อมูล สำหรับให้ผู้ใช้ในการแท็กข้อมูล

ขั้นตอนแรกของการใช้งานนั้น ผู้ใช้จะเลือกบริการที่ต้องการแท็กข้อมูล แล้วทำการศึกษารายละเอียดของบริการนั้นๆ จากนั้นทำการเลือกแท็กจากข้อมูลจากแผนภาพต้นไม้ (Tree view) ของ Ontology NAICS ที่เกี่ยวข้องกับบริการนั้นๆที่แสดงอยู่หรือสามารถใส่ข้อมูลได้เอง (User define tag) ตามความต้องการ จากนั้นระบบจะแสดงแท็กที่มีความหมายที่เกี่ยวข้องกับข้อมูลนั้นๆ ผู้ใช้จะสามารถเลือกแท็กเหล่านั้นได้ตามต้องการ และทำการบันทึกแท็กที่ต้องการเหล่านั้นลงในฐานข้อมูล

WID_URL

[illegible]	[illegible]	[illegible]
S01	http01	Path01
S02	http02	Path02
S03	http03	Path03
…	…	…

Link_Services_and_Message

[illegible]	[illegible]
S01	T01
S02	T01
S03	T02
…	…

Table_Tag_Message

[illegible]	[illegible]	[illegible]
T01	Word01	1
T02	Word02	26
T03	Word03	15
…	…	…

รูป 3.8 รูปแสดงตารางของแท็กในฐานข้อมูลระบบ Tagging System

การแท็กข้อมูลที่เกี่ยวข้องของบริการข้างต้น เป็นการแท็กข้อมูลที่เกี่ยวข้องกับบริการ เป็นการแสดงรายละเอียดและความหมายของบริการนั้นๆ ในการค้นหาข้อมูลจึงมีการพิจารณาแท็กข้อมูลเหล่านี้ ที่ถูกเก็บไว้ในฐานข้อมูลตามความสัมพันธ์ เพื่อใช้ในการค้นหาบริการที่เกี่ยวข้องที่ผู้ใช้ต้องการ ฐานข้อมูลแท็กเหล่านี้ ประกอบด้วย ข้อมูลแท็กและความสัมพันธ์ของบริการต่างๆ เป็น 3 ตารางความสัมพันธ์ ดังรูป 3.8 ประกอบด้วย

1) WID_URL เป็นตารางความสัมพันธ์ของบริการต่างๆ ของไฟล์ WSDL, OWL หรือ OWLS ตารางความสัมพันธ์ประกอบด้วย ข้อมูล URL ของแต่ละบริการ และกำกับด้วยหมายเลขบริการ รวมถึงรายละเอียดของ Directory ที่บริการนั้นเก็บอยู่ เพื่อสามารถเรียกใช้บริการและรายละเอียดของบริการแก่ผู้ใช้

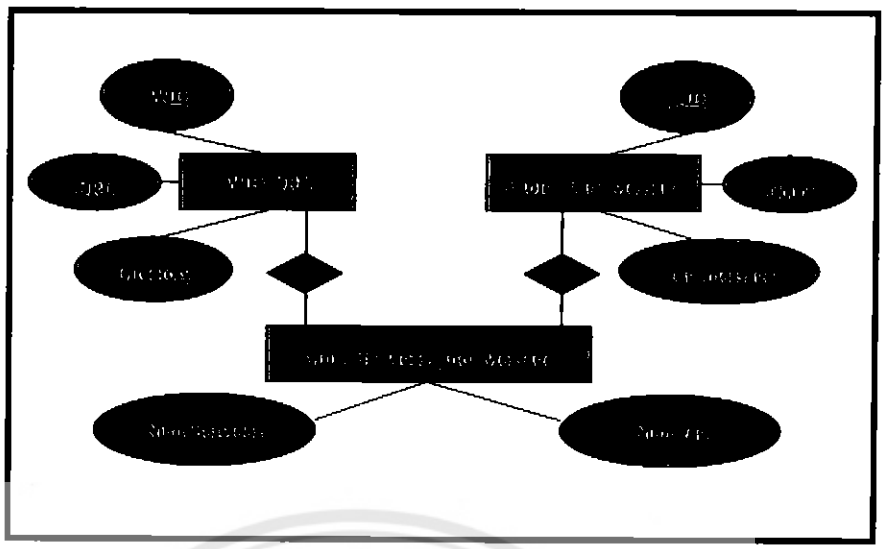
2) Table_Tag_Message เป็นตารางความสัมพันธ์ของข้อมูลที่แท็กได้จากผู้ใช้ทั้งหมด เก็บข้อมูลแท็กที่บันทึกจากผู้ใช้ เพื่อใช้ในการตรวจสอบความสัมพันธ์ของ URL เมื่อต้องการค้นหาข้อมูล ในตารางนี้มีที่มีการนับจำนวนแท็กทั้งหมดและแท็กที่ซ้ำกันที่ผู้ใช้แท็ก เพื่อแสดงให้เห็นว่าแท็กนั้นๆ มีความสัมพันธ์กับบริการนั้นมากน้อยแค่ไหน

3) Link_Services_and_Message เป็นตารางความสัมพันธ์ของหมายเลข ID ของแท็กในตาราง *Table_Tag_Message* กับหมายเลข ID ของบริการ (Services) ที่แสดงได้ว่าบริการไหน มีความสัมพันธ์กับแท็กใดๆบ้าง ซึ่งเป็นความสัมพันธ์แบบ one-to-many ของบริการต่างๆ

รูป 3.9 รูปแสดงตารางและความสัมพันธ์ของแท็กและบริการในฐานข้อมูล

3.2.1.2) ER Diagram

ในฐานข้อมูลข้างต้น ข้อมูลแต่ละ Entiry จะมีความสัมพันธ์ระว่าง Entiry หรือกลุ่มของ ข้อมูลต่างๆ สามารถที่จะเขียนเป็น Entity Relationship Diagram ได้ดังรูปที่ 3.10



รูป 3.10 โครงสร้าง ER – DIAGRAM ของฐานข้อมูลแท็ก

3.2.2 ฐานข้อมูลของระบบการแสดงผล (Interface System)

ระบบการแสดงผล (Interface System) เป็นส่วนของการแสดงผลและติดต่อการใช้งานกับผู้ใช้ มีหน้าเว็บสำหรับการค้นหาเว็บเซอร์วิส ใช้สำหรับเป็นสื่อระหว่างผู้ให้บริการ (Provider) กับผู้ใช้ (User) ในการติดต่อการใช้งานระบบการค้นหา ระบบแสดงผล (Interface System) มี User Query สำหรับผู้ใช้กรอกข้อมูลคำค้นหา เพื่อทำการค้นหาเว็บเซอร์วิสตามต้องการ ดังรูป 3.11

เมื่อผู้ใช้ทำการใส่คำค้นหา ระบบแสดงผล (Interface System) จะทำการวิเคราะห์คำค้นหาเปลี่ยนเป็นข้อมูลที่ใช้สำหรับการค้นหา มีการแสดงรายละเอียดของคำที่เป็นคีย์เวิร์ดที่เกี่ยวข้องกับคำค้นหาที่ผู้ใช้กรอกข้อมูลเข้ามา เพื่อเป็นแนวทางในการเลือกค้นหาบริการเว็บเซอร์วิสที่ต้องการ เมื่อผู้ใช้เลือกคีย์เวิร์ดเหล่านี้ที่เกี่ยวข้องกับคำค้นหาหรืออาจมีความคล้ายคลึงกับคำค้นหาที่ต้องการแล้วระบบจะทำการสร้าง Boolean Expression เพื่อกำหนดเงื่อนไขของการค้นหา ตัวอย่างการวิเคราะห์คำค้นหา ดังนี้

ตัวอย่าง การวิเคราะห์คำค้นหาในส่วนระบบแสดงผล (Interface System)

1. ผู้ใช้กรอกคำค้นหา เช่น Resort has yoga near beach

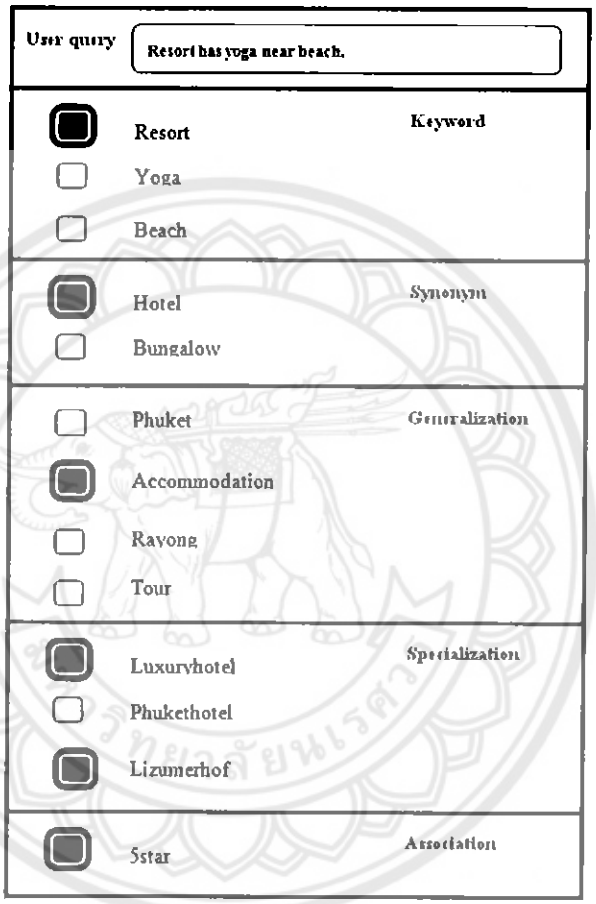
2. ระบบทำการวิเคราะห์ได้คีย์เวิร์ดใหม่ คือ Resort, yoga, beach

3. ระบบทำการแสดงข้อมูลคีย์เวิร์ด เพื่อให้ผู้ใช้เลือกข้อมูลที่เกี่ยวข้องที่ต้องการ เมื่อเลือกคีย์เวิร์ดแล้ว ระบบทำงาน

- หา Synonym ของคีย์เวิร์ดนั้นๆ เช่น Resort มี Synonym คือ Hotel Guess house เป็นต้น
- หาโหนดของ Generalization (โหนดแม่) ซึ่งเป็นซับคลาสของข้อมูลที่เลือก เช่น Phuket เป็น Generalization ของ Resort

- หาโหนดของ Specialization (โหนดลูก) ซึ่งเป็นซับคลาสของข้อมูลที่เลือก เช่น yoga Resort เป็น Specialization ของ Resort

- หา Association (any or none) ของข้อมูลที่เลือก ซึ่งเป็นความสัมพันธ์ของข้อมูล 2 ข้อมูล ที่มีความสัมพันธ์ เช่น Hotel และ Luxuryhotel มีความสัมพันธ์กันด้วย 5star



รูป 3.11 Search Interface

4. จากนั้นแสดงข้อมูลเหล่านั้นเป็นรายการของสำหรับผู้ใช้เลือกข้อมูลที่เกี่ยวข้อง ที่ต้องการ สำหรับเป็นแนวทางของข้อมูลในการค้นหา ซึ่งจะทำการค้นหาข้อมูลที่เกี่ยวข้องข้างต้นในทุกๆข้อมูลที่ผู้ใช้เลือก

5. จากนั้นระบบทำการวิเคราะห์ข้อมูลเหล่านี้ ที่ผู้ใช้เลือกมาสร้างประโยคสำหรับสืบค้น Boolean Expression เพื่อเป็นเงื่อนไขในการค้นหาของระบบ

บทที่ 4
การพัฒนาและใช้งานระบบ

จากการออกแบบระบบที่ได้จากการศึกษาค้นคว้าข้อมูลและทฤษฎี สามารถสร้างระบบการค้นหาที่สามารถแสดงผลการค้นหา และสามารถทำงานแบบเว็บเซอร์วิสสำหรับการค้นหาบริการที่ต้องการได้ บทนี้เป็นการแสดงการทำงานของระบบที่ได้จากการออกแบบ โดยการสร้างระบบการค้นหาเว็บเซอร์วิสจากการออกแบบระบบข้างต้น ด้วยวิธีการเปรียบเทียบ Boolean Expression กับข้อมูลแท็กในฐานข้อมูล เพื่อค้นหาบริการเว็บเซอร์วิสที่ต้องการ

4.1 Hardware และ Software Requirement

4.1.1 Hardware Requirement

ตารางที่ 4.1 แสดงรายการของ Hardware Requirement

Hardware	Specification
Windows	Window XP ,Vista , 7
Process	Intel Pentium 4 and above
Memory	256 MB and above
Hard disk	10 GB and above
Networking	10/100 Internet

4.1.2 Software Requirement

ตารางที่ 4.2 แสดงรายการของ Software Requirement

Software Requirement	Purpose
1. Microsoft visual studio 2008 professional edition VS2008ProEdition90dayTrialENUX1435622.iso (http://www.microsoft.com)	-พัฒนา Platform ในการเขียนโปรแกรมภาษา C# และรัน Coding - ใช้สร้าง ASP.NET Web Services Application
2.SQLite Expert Personal SQLiteExpertSetup.exe (http://www.sqliteexpert.com)	- การสร้างฐานข้อมูล

4.2 อินพุต เอาต์พุต และเซ็ตข้อมูล

4.2.1 ข้อมูลอินพุต

4.2.1.1) อินพุตจากระบบการแสดงผล (Interface System)

อินพุตของระบบการค้นหาเว็บเซอร์วิสจากส่วนระบบการแสดงผลมีการรับอินพุตเป็น Boolean Expression ประกอบด้วยประโยค Boolean Expression ที่มีความยาวไม่จำกัด และข้อมูลคำที่เกี่ยวข้องกับคำค้นหารวมถึงเงื่อนไขของข้อมูลในการค้นหา ไฟล์ XML ที่รับมานั้นมาจากระบบการแสดงผล (Interface System)

```
<?xml version="1.0" encoding="utf-8"?>
<booleanExpression ID="(airport or transportfacility) or (yoga or relaxation) or
(skiing and asAccommudation) or (resort or hotel and hotelcentral and
jugendherbergeinnsbruck)">
<Data>
<requestWord>airport</requestWord>
<generalization>transportfacility</generalization>
</Data>
<Data>
<requestWord>yoga</requestWord>
<generalization>relaxation</generalization>
</Data>
<Data>
<requestWord>skiing</requestWord>
<association>hasAccommudation</association>
        <associationNode>
        <Asso01>skiingresortaxamerlizum</Asso01>
        <Asso02>skiingresortsattelbergbahn</Asso02>
        </associationNode>
</Data>
</booleanExpression>
```

รูปที่ 4.1 ไฟล์ XML อินพุตของ Boolean Expression ของระบบจากส่วนระบบการแสดงผล

ประโยค Boolean Expression จากส่วนระบบการแสดงผล (Interface System) มาจากการวิเคราะห์คำค้นหาของผู้ใช้ในหน้าการค้นหาเว็บเซอร์วิส (User Interface) ไฟล์ XML ของประโยค Boolean Expression นี้ ประกอบด้วย XML ของเป็นประโยคเงื่อนไขของ Boolean Expression และข้อมูลของคำค้นหาเหล่านั้น ใช้ในการพิจารณาเงื่อนไขในการค้นหาและวิเคราะห์หาค่าน้ำหนักของข้อมูล แสดงอินพุตของระบบจากส่วนระบบการแสดงผล (Interface System) ดังรูป 4.1

4.2.1.2) อินพุตจากระบบการฐานข้อมูลแท็ก (Tagging System)

อินพุตของระบบของระบบการค้นหาเว็บเซอร์วิสจากระบบการฐานข้อมูลแท็ก (Tagging System) รับอินพุตเป็นไฟล์ XML ของข้อมูล URL ที่ได้จากการเปรียบเทียบคำค้นหาที่มาจาก Boolean Expression กับแท็กในฐานข้อมูลแท็ก (Tagging System) เพื่อใช้ URL เหล่านี้ในการ

วิเคราะห์ URL ที่เกี่ยวข้องกับคำค้นหาของระบบ แสดงส่วนอินพุตของระบบของระบบการค้นหาเว็บเซอร์วิสจากส่วนฐานข้อมูลแท็ก (Tagging System) ดังรูป 4.2

```
<?xml version="1.0" encoding="utf-8"?>
<ListURL>
  <url>http://127.0.0.1/services/1.0/1personbicycle4wheeledcar_price_service.owls</url>
  <url>http://127.0.0.1/services/1.0/1personbicyclecar_price_Kohlservice.owls</url>
  <url>http://127.0.0.1/services/1.0/1personbicyclecar_price_TheBestservice.owls</url>
  <url />
  <Title />
</ListURL>
```

รูปที่ 4.2 ไฟล์ XML อินพุตของ URL ที่ได้จากระบบการฐานข้อมูลแท็ก

4.2.2 ข้อมูลเอาต์พุต

4.2.1.2) เอาต์พุตที่ส่งให้แก่ระบบการฐานข้อมูลแท็ก (Tagging System)

เอาต์พุตที่ส่งให้แก่ระบบการฐานข้อมูลแท็ก (Tagging System) เป็น XML ของคำที่ต้องการค้นหาที่มาจากประโยค Boolean Expression ที่ได้จากส่วนระบบการแสดงผล (Interface System) ส่งไปเพื่อทำการค้นหาข้อมูล URL ที่เกี่ยวข้องกับคำค้นหาจาก Boolean Expression ในฐานข้อมูลแท็ก (Tagging System) เพื่อใช้ URL เหล่านี้ในการวิเคราะห์ URL ที่เกี่ยวข้องกับคำค้นหาของระบบ ส่วนเอาต์พุตของระบบแสดงดังรูป 4.3

```
<?xml version="1.0" encoding="utf-8" ?>
<ListURL>
<Data>Production_Value</Data>
</ListURL>
```

รูปที่ 4.3 ไฟล์ XML เอาต์พุตของ URL ที่ได้จากระบบการฐานข้อมูลแท็ก

4.2.1.1) เอาต์พุตที่ส่งให้แก่ระบบการแสดงผล (Interface System)

เอาต์พุตที่ส่งให้แก่ระบบการแสดงผล (Interface System) เป็น XML รายการของ URL ที่มีการวิเคราะห์ข้อมูลด้วยเงื่อนไขใน Boolean Expression แล้ว มีการเรียงลำดับผลการค้นหาที่ได้จากการพิจารณาค่าน้ำหนักของ URL เพื่อใช้ในการแสดงผลการค้นหาที่ผู้ใช้ต้องการ

ไฟล์ XML ข้อมูลนี้ เก็บเว็บเซอร์วิสที่ได้จากการค้นหา ตามเงื่อนไข Boolean Expression ที่เป็นเงื่อนไขการค้นหาของคำค้นหาของผู้ใช้ เพื่อแสดงผลการค้นหาของบริการของเว็บเซอร์วิสเหล่านี้ในส่วนระบบการแสดงผล (Interface System) แสดงดังรูป 4.4

```
<?xml version="1.0" encoding="utf-8"?>
<ListURL>
  <url>http://127.0.0.1/services/1.0/1personbicycle4wheeledcar_price_service.owls</url>
  <url>http://127.0.0.1/services/1.0/1personbicyclecar_price_Kohlservice.owls</url>
  <url>http://127.0.0.1/services/1.1/author_bookrecommendedprice_service.owls</url>
  <url>http://127.0.0.1/services/1.1/4wheeledcar1personbicycle_price_service.owls</url>
  <url>http://127.0.0.1/services/1.1/3wheeledcaryear_price_service.owls</url>
  <url>http://127.0.0.1/services/1.1/_coffeeteareport_service.owls</url>
  <url>http://127.0.0.1/services/1.1/_author_CompJservice.owls</url>
  <url>http://127.0.0.1/services/1.0/book_authorbook-type_service.owls</url>
  <url>http://127.0.0.1/services/1.0/_pricecamera_Wallmartservice.owls</url>
  <url>http://127.0.0.1/services/1.0/cdplayer_recommendedprice_service.owls</url>
  <url>http://127.0.0.1/services/1.1/autobicycle_maxprice_service.owls</url>
</ListURL>
```

รูปที่ 4.4 ไฟล์ XML เอาต์พุตของผลการค้นหาที่ส่งให้แก่ระบบการแสดงผล

4.2.3 เซ็ตข้อมูล

เมื่อมีการใช้แท็กในระบบแท็กข้อมูล มีการใช้งานเซอร์วิสจากบางกลุ่มข้อมูลในการทดสอบและทดลองการทำงานระบบการการค้นหาทั้งในส่วนของระบบการค้นหาและระบบอื่นๆ อีก 2 ระบบ เพื่อทดลองการใช้งานในการแท็กและการค้นหาบริการ เซ็ตข้อมูลที่ใช้ในกระบวนการการทำงานมีดังนี้

- OWLS-TC (โดยการใช้ Test Collection ของ OWL-S และ NAICS ของ Test Collection นี้)

  OWLS-TC4_PDDL.zip

  (http://www.semwebcentral.org/frs/?group_id=89&release_id=380)

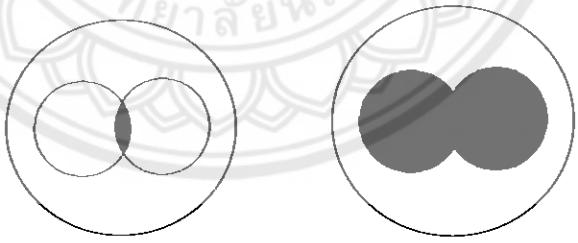
4.3 ระบบการค้นหา (Search System)

4.3.1 ประโยคสืบค้น Boolean Expression

การค้นหาด้วยการใช้บูลีนโอเปอร์เรเตอร์นั้น สามารถช่วยในการสืบค้นข้อมูลได้ดีอีกวิธีหนึ่ง ดังกล่าวในหัวข้อ 2.3.5 Boolean Expression เป็นผลลัพธ์ที่เกิดจากนิพจน์ทางคณิตศาสตร์ ประโยคสืบค้น Boolean Expression ในระบบการค้นหานี้ Boolean Expression หมายถึง ประโยคผลลัพธ์ที่เกิดจากนิพจน์ที่ได้จากการวิเคราะห์คำที่เกี่ยวข้องกับคำค้นหาของผู้ใช้ ได้จากหน้าการค้นหาข้อมูลในส่วนระบบการแสดงผล (Interface System) ประโยคที่ได้จะประกอบด้วยข้อมูลและเงื่อนไขในการค้นหา ซึ่งเป็นสัญลักษณ์ที่สัมพันธ์กับข้อความในประโยคสืบค้น Boolean Expression เงื่อนไขที่ใช้ในการพิจารณา ประกอบด้วยสัญลักษณ์ 2 ลักษณะ คือ AND (∪) และ OR (∩) ซึ่งเป็นเงื่อนไขในการค้นหาบริการให้ได้ตรงกับคำค้นหาของผู้ใช้ ดังนี้

ตารางที่ 4.3 สัญลักษณ์เงื่อนไข Boolean Expression

สัญลักษณ์	ตัวอย่างรูปแบบที่ใช้งาน	ลักษณะที่ค้นหา
AND (∪)	resort and hotel	พบทั้ง resort และ hotel เท่านั้น
OR (∩)	resort or hotel	พบ resort และ hotel หรือทั้งสองอย่าง



a. b.

รูปที่ 4.5 เงื่อนไขของ Boolean Expression ในการดึงข้อมูล

a. AND (∪) b. OR (∩)

AND สร้างความสัมพันธ์ระหว่างข้อมูล 2 กลุ่ม จะเกิดกลุ่มใหม่ที่มีลักษณะของข้อมูลเดิมทุกกลุ่ม ข้อมูลกลุ่มใหม่จึงมีขนาดเล็กลง หรือแคบลง ข้อมูลที่ค้นหามาได้จะประกอบด้วยคำทั้งหมดจะถูกนำมาแสดงผล เช่น Computer AND Design ผลลัพธ์ที่ได้ จะได้ข้องมูลที่ต้องมีคำว่า Computer และ Design อยู่ด้วยกันเท่านั้น จึงจะดึงข้อมูลนั้นมาแสดงผล เป็นต้น

OR การสร้างความสัมพันธ์ระหว่างข้อมูล 2 กลุ่ม จะเกิดกลุ่มใหม่ที่มีลักษณะของข้อมูลเดิมจากกลุ่มใดกลุ่มหนึ่งหรือทุกกลุ่ม ดังนั้นข้อมูลใหม่จะมีขนาดใหญ่ขึ้นหรือมีขอบเขตกว้างขึ้น เช่น Computer OR Design หมายความว่า "ค้นหาข้อมูลที่มีคำว่า Computer และ Design " ฉะนั้นผลลัพธ์ที่ได้เมื่อใช้ข้อความนี้ค้นหาข้อมูลจะแสดงข้อมูลทั้งหมดที่เกี่ยวข้องกับ Computer หรือ Design ถ้าไม่เกี่ยวข้องจะไม่ดึงเอาข้อมูลนั้นมาแสดง เป็นต้น

การใช้งานเงื่อนไขของ Boolean Expression นั้นจะพิจารณาคำจากผู้ใช้เลือก ระบบนำมาวิเคราะห์ใช้งานร่วมกับเงื่อนไข Boolean Expression ที่มี คำจากผู้ใช้เลือกนั้นมาจากการวิเคราะห์ข้อมูลคำค้นหาของระบบที่แสดงให้ผู้ใช้เลือกเป็นแนวทางในการค้นหาข้อมูล ซึ่งมีหลายประเภทของข้อมูล ดังตารางที่ 4.4 ระบบจะเลือกใช้เงื่อนไข Boolean Expression ในแต่ละตัวของข้อมูลแตกต่างกันออกไป เนื่องจากความสำคัญและความเป็นไปได้ที่ข้อมูลที่ทำการค้นหามีความแตกต่างกัน การเลือกใช้เงื่อนไขที่แตกต่างกันจะทำให้สามารถค้นหาข้อมูลที่ตรงกับความต้องการของผู้ใช้มากที่สุด

ตารางที่ 4.4 ความหมายและการใช้งานเงื่อนไขของ Boolean Expression

[illegible]	[illegible]	[illegible]
Keyword	Keyword ที่ได้จากข้อมูลที่ผู้ใช้กรอกโดยตรง	-
Synonym	คำที่มีความหมายคล้ายคลึงกับ Keyword	OR
Generalization	โหนดแม่ของ Keyword ใน NAICS	OR
Specialization	โหนดลูกของ Keyword ใน NAICS	AND
Association	ความสัมพันธ์ของโหนดที่พิจาณา	AND

ตัวอย่าง การใช้งาน ความหมายและการใช้งานเงื่อนไขของ Boolean Expression

- Resort เป็นข้อมูล Keyword
- Hotel เป็นข้อมูล Synonym
- Accommodation เป็นข้อมูล generalization
- Lizumehof เป็นข้อมูล specialization
- Luxuryhotel เป็นข้อมูล specialization
- 5star เป็นความสัมพันธ์ Association ของข้อมูล hotel และข้อมูล Luxuryhotel

ดังนั้น ประโยค Boolean Expression มีลักษณะดังนี้

Resort or Hotel or Accommodation and Lizumahof and Luxuryhotel and 5star

ความหมายของ Boolean Expression นี้คือ ค้นหาข้อมูลในฐานข้อมูลแท็ก โดยกำหนดเงื่อนไขว่า ข้อมูลนั้นต้องประกอบด้วยข้อมูลแท็กที่เป็น Resort ด้วยเสมอ อาจมี Hotel หรือ Accommodation หรือ Lizumehof อย่างใดอย่างหนึ่งหรือทั้งหมดได้ และต้องมี Luxuryhotel และ 5star ด้วยเสมอ

ในการสร้างประโยค Boolean Expression นั้น จำเป็นต้องสร้างตามการเลือกข้อมูลของ Keyword ของผู้ใช้เป็นสำคัญ กล่าวคือเมื่อเลือก Keyword ใด Keyword หนึ่งจะมีการสร้าง Boolean Expression สำหรับ Keyword นั้นๆเท่านั้น จากนั้นจะนำข้อมูลที่เกี่ยวข้องทั้งหมดในทุก Keyword มารวมกันโดยใช้ความสัมพันธ์ OR

ตัวอย่าง ความสัมพันธ์ของ Boolean Expression ในหลาย Keyword

(Resort or Hotel or Accommodation and Lizumahof and Luxuryhotel and 5star) or (Beach or Phuket or Sport) or (Yoca or PrinceOfSongklaUniversity and BookStore)

นั่นหมายความว่า ข้อมูลที่ได้จะมาจากในหลายๆ Boolean Expression ที่รวมกัน ดังนั้นข้อมูลที่ได้จะมีความหลากหลายมากขึ้นในการตอบสนองความต้องการของผู้ใช้

การสร้างประโยคสืบค้น Boolean Expression ต้องคำนึงถึงความต้องการของผู้ใช้มากที่สุด คือสามารถแสดงผลลัพธ์ได้ตรงความต้องการและสามารถตอบสนองความต้องการของผู้ใช้ได้อย่างรวดเร็ว และถูกต้อง

4.3.2 การค้นหาข้อมูลโดยการเปรียบเทียบแท็กข้อมูล

ระบบการค้นหาเว็บเซอร์วิส มีหลักการค้นหาข้อมูลโดยการเปรียบเทียบข้อมูลแท็กในฐานข้อมูลซึ่งผู้ให้บริการ (Provider) กำหนดขึ้นให้กับเว็บเซอร์วิส ซึ่งแท็กนั้นๆมาจาก Ontology NAICS และแท็กที่กำหนดเองโดยผู้ใช้ (User Defined Tag)

ในการทำการเปรียบเทียบข้อมูลนั้น จำเป็นต้องมีการรับข้อมูลประโยค Boolean Expression ที่มีความชัดเจนเพียงประโยคเดียวในการทำการค้นหาในหนึ่งครั้ง เพื่อวิเคราะห์หาเงื่อนไขที่เป็นไปได้ทั้งหมดที่จะพิจารณา และสามารถทำการเปรียบเทียบตารางฐานข้อมูลได้อย่างถูกต้องตามหลักการทำงานของระบบ ซึ่งมีหลักการทำงานหลักๆ ดังนี้

1) ส่งไฟล์ XML เพื่อร้องขอให้ระบบฐานข้อมูลแท็ก (Tagging System) ในการค้นหาบริการที่เกี่ยวข้องกับข้อมูลคำค้นหาของประโยค Boolean Expression โดยการส่งไฟล์ XML ของประโยค Boolean Expression เพื่อทำการค้นหาข้อมูลมาใช้งานในระบบ

2) รับข้อมูลที่ได้จากการค้นหาแล้วที่ได้จากในระบบฐานข้อมูลแท็ก (Tagging System) จะแสดงเป็นไฟล์ XML ของรายการ URL ที่เกี่ยวข้องกับชุดข้อมูลของคำค้นหาแต่ละตัว นำข้อมูลพิจารณาตามเงื่อนไขของ Boolean Expression ตามความสัมพันธ์ในตางที่ 4.4 ได้ข้อมูล

ความสัมพันธ์ของ URL แต่ละตัว ที่มีเงื่อนไขแตกต่างกันออกไปตามความหมายในประโยค Boolean Expression

ตัวอย่าง

http://127.0.0.1/service/1.1/userRomanticnovel_price_service.owls

Resort or Hotel or Accommodation or Lizumahof and Luxuryhotel and 5star

ความหมาย คือ http://127.0.0.1/services/1.1/userRomanticnovel_price_service.owls มีแท็กที่พบ คือ เงื่อนไข Boolean Expression ของ Resort or Hotel or Accommodation or Lizumahof and Luxuryhotel and 5star

ตัวอย่าง

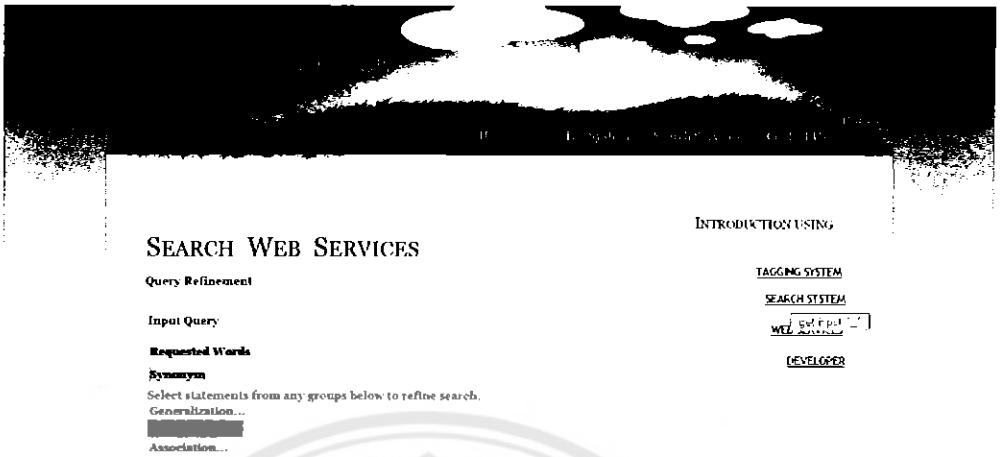
http://127.0.0.1/services/1.0/vehicle_price_service.owls
http://127.0.0.1/services/1.0/towncountry_hotel_service.owls

Resort or Hotel and Luxuryhotel and 5star

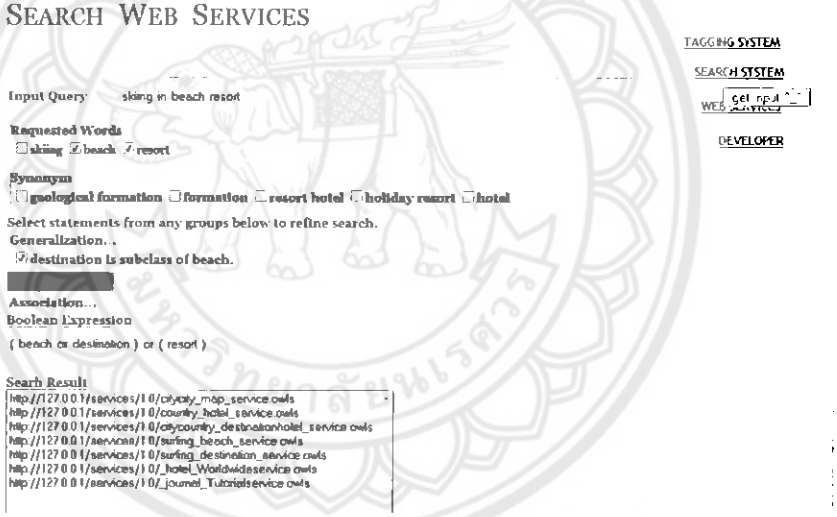
ความหมาย คือ http://127.0.0.1/services/1.0/vehicle_price_service.owls และ http://127.0.0.1/services/1.0/towncountry_hotel_service.owls มีแท็กที่พบ คือ เงื่อนไข Boolean Expression ของ Resort or Hotel and Luxuryhotel and 5star

ข้อมูล URL ที่พบอาจมีหลากหลายในประโยค Boolean Expression หนึ่งๆและอาจซ้ำกันในหลาย Boolean Expression เช่นกัน อาจเนื่องด้วยข้อมูลที่ซ้ำกันหรือการกำหนดแท็กให้กับบริการนั้นๆ ดังนั้นต้องมีการเลือกใช้ข้อมูลที่ถูกต้องและตรงกับความต้องการของผู้ใช้มากที่สุด จึงต้องมีการจัดลับดับของข้อมูลที่มีความสำคัญมากที่สุดที่อาจกล่าวได้ว่าจะตรงกับความต้องการของผู้ใช้มากที่สุด นั่นคือการกำหนดค่าน้ำหนักให้กับข้อมูล โดยพิจารณาว่า ข้อมูล URL ที่พบแท็กมากที่สุดอาจหมายความได้ว่าข้อมูลนั้นต้องมีค่าน้ำหนักที่มากนั่นเอง ดังจะกล่าวในหัวข้อ 4.4 การจัดลำดับผลของการค้นหา

4.3.3 การทำงานสื่อสารของเว็บเซอร์วิสในระบบการค้นหาเว็บเซอร์วิส

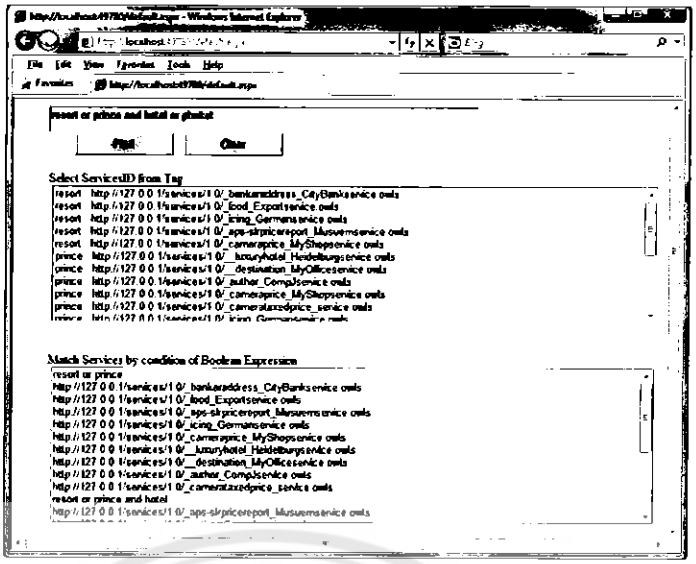


รูปที่ 4.6 หน้า Interface สำหรับการค้นหาเว็บเซอร์วิส



รูปที่ 4.7 ตัวอย่างการค้นหาเว็บเซอร์วิส โดยการกรอกข้อมูลคำค้นหาและแสดงผลการค้นหา

รูปที่ 4.6 และ 4.7 แสดงหน้า Interface สำหรับการค้นหาเว็บเซอร์วิสและหน้า Interface ที่ตัวอย่างการค้นหาเว็บเซอร์วิส โดยการกรอกข้อมูลคำค้นหาและแสดงผลการค้นหา โดยแสดงข้อมูลที่มีการรวมทั้ง 3 ระบบ ที่ใช้ในการค้นหา มีการค้นหาโดยการกรอกคำค้นหาที่ต้องการและมีการแสดงผลของข้อมูลที่ได้จากการค้นหาแสดงแก่ผู้ใช้



รูปที่ 4.8 ตัวอย่างการค้นหาข้อมูลที่ใช้ในระบบ แสดงข้อมูลตามเงื่อนไขของ Boolean Expression

รูปที่ 4.9 ตัวอย่างระบบเว็บเซอร์วิสในการค้นหาข้อมูล แสดงเซอร์วิสที่ใช้ ผลการค้นหา และน้ำหนักของข้อมูลตามเงื่อนไขของ Boolean Expression

รูปที่ 4.8 แสดงตัวอย่างการค้นหาข้อมูลที่ใช้ในระบบการค้นหา แสดงข้อมูลตามเงื่อนไขของ Boolean Expression โดยการเปรียบเทียบข้อมูลจากฐานข้อมูลที่ได้ทั้งหมดกับเงื่อนไขของ Boolean Expression ในทุกข้อมูลที่ต่างกัน เพื่อนำไปใช้ตรวจสอบความถูกต้องก่อนนำไปรวมกับระบบจริง รูปที่ 4.9 แสดงส่วนระบบค้นหาที่ใช้สำหรับการค้นหาเว็บเซอร์วิส ที่มีการเชื่อมต่อกับ

เว็บเซอร์วิสของระบบฐานข้อมูลแท็ก มีการรับและส่งไฟล์ XMLของระบบในการการสื่อสารในข้อมูลระหว่างเว็บเซอร์วิสเพื่อทำการค้นหาข้อมูล แสดงผลของข้อมูลที่ได้จากการค้นหาตามเงื่อนไขของ Boolean Expression

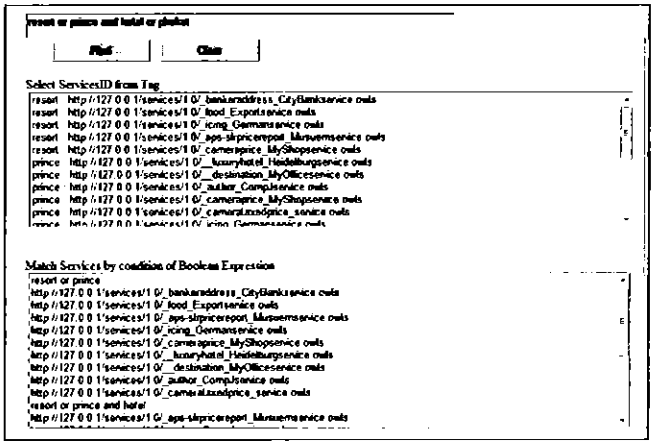
4.3.4 Pseudo code ในการทำงานของระบบค้นหาเว็บเซอร์วิส

การเปรียบเทียบแท็กกับข้อมูลคำค้นหาใน Boolean Expression นั้น ระบบจะทำการ Query ข้อมูลจากฐานข้อมูล ดังรูป 4.10 และแสดงข้อมูลที่ได้จากการ Query ดังรูป 4.11

```
CONNECT PathofDATA FROM DATABASE("//TAG.db")//USE SQLITE DATABASE
DO SELECT * FROM Table_TagMessage WHERE Tag_Message Like "%datacheck"%
  SELECT id_TAG
        DO SELECT * FROM Link_Service_and_Message WHERE Num_tag =id_TAG
            SELECT id_Services
                  DO SELECT * FROM WID_URL WHERE ID=id_Services
                     SELECT Url
                          DO checkData = Url
                          ENDDO
                  ENDDO
        ENDDO
        DO return checkData
        ENDDO
ENDDO
```

รูปที่ 4.10 Pseudo code การ Query ข้อมูลจากฐานข้อมูล

```
<?xml version="1.0" encoding="utf-8"?>
<ListURL>
  <url>http://127.0.0.1/services/1.0/1personbicycle4wheeledcar_price_service.owls</url>
  <url>http://127.0.0.1/services/1.0/1personbicyclecar_price_Kohlservice.owls</url>
  <url>http://127.0.0.1/services/1.0/1personbicyclecar_price_TheBestservice.owls</url>
  <url />
  <Title />
</ListURL>
```



รูปที่ 4.11 ผลของข้อมูลที่ได้จากการ Query ข้อมูลจากฐานข้อมูล

อินพุตที่รับเข้ามาจากส่วนแสดงผล (Interface System) ที่ข้อมูลนั้นได้จากการวิเคราะห์คำค้นหาที่ได้จากผู้ใช้ ในการค้นหาแต่ละครั้ง Boolean Expression ที่ได้จะแตกต่างกันออกไป ก่อนการทำงานของระบบจะมีการกำหนดข้อมูลที่ได้จาก Boolean Expression ที่เข้ามา เพื่อระบบสามารถทำงานอย่างรวดเร็วและถูกต้อง มีกำหนดข้อมูลในการใช้ทำงาน โดยมีการกำหนดการอ่านข้อมูล แสดง Pseudo code ดังรูป 4.12

```
XmlNodeList fnames = xdoc.GetElementsByTagName("Data")
REPEAT WHILE codition
      dataCheck = dataCheck.Replace("<requestWord>", "?")
      dataCheck = dataCheck.Replace("</requestWord>", "")
      dataCheck = dataCheck.Replace("<generalization>", " or ")
      dataCheck = dataCheck.Replace("</generalization>", "")
      dataCheck = dataCheck.Replace("<specialization>", " and ")
      dataCheck = dataCheck.Replace("</specialization>", "")
      dataCheck = dataCheck.Replace("<synonym>", " or ")
      dataCheck = dataCheck.Replace("</synonym>", "")
      dataCheck = dataCheck.Replace("<association>", " and +")
      dataCheck = dataCheck.Replace("</association>", "")
      dataCheck = dataCheck.Replace("<Asso1>", ",")
      dataCheck = dataCheck.Replace("</Asso1>", "")
      dataCheck = dataCheck.Replace("<Asso2>", ",")
      dataCheck = dataCheck.Replace("</Asso2>", "")
      dataCheck = dataCheck.Replace("<associationNode>", "")
      dataCheck = dataCheck.Replace("</associationNode>", "")
      GetData.Add(dataCheck)
```

รูปที่ 4.12 Pseudo code การอ่านไฟล์อินพุต Boolean Expression

```
Checkdata(datacheck[])
      IF datacheck have Only FirstData
            THEN  Data[] = datacheck
      ENDIF
      ELSE IF
            REPEAT WHILE codition
            IF have next datacheck[]
                  THEN CONPARE data with FirstData
                        IF FirstData = datacheck
                              THEN not KEEP
                        ENDIF
                        ELSR IF FirstData != datacheck
                              THEN Data[] = datacheck
                        ENDIF
            ENDIF
      ENDIF
      DO return Data[]
      ENDDO
```

รูปที่ 4.13 Pseudo code การตรวจสอบข้อมูลที่ซ้ำกัน

เมื่อทำการดึงข้อมูลบริการที่ได้จากระบบฐานข้อมูลแท็กแล้ว จะได้ข้อมูลที่มากและมีโอกาสที่ข้อมูลจะซ้ำกันมาก จึงต้องมีการพิจารณาตัดข้อมูลที่ซ้ำกันออก เพื่อการใช้งานข้อมูลที่เป็นประโยชน์เท่านั้น แสดง Pseudo code รูป 4.13

จากนั้นทำการเปรียบเทียบข้อมูลของแท็กด้วยเงื่อนไข Boolean Expression นั้นคือ AND ($\cap$) และ OR ($\cup$) แสดง Pseudo code รูปที่ 4.14 โดยทำการเลือกพิจารณาค้นหา Boolean Expression ทีละคู่ตามเงื่อนไข

การพิจารณาตามเงื่อนไขของ Boolean Expression จะได้เซตของ URL ที่แตกต่างกันออกไปในแต่ละเงื่อนไขที่ทำการพิจารณา ยิ่งมี Boolean Expression ยาวมากเท่าไหร่ ข้อมูลผลการค้นหายิ่งมากขึ้นหรือน้อยลงขึ้นอยู่กับเงื่อนไขใน Boolean Expression เช่นกัน

```
CheckOrAnd (datacheckBoolean[])
      REPEAT WHILE codition
      IF Relation = OR
            THEN  KEEP sameURL REPEAT WHILE codition
                  KEEPData = URL + WeigthOR
      ENDIF
      ELSE IF Relation = AND
            THEN KEEP notsameURL REPEAT WHILE codition
                  KEEPData = URL + WeigthAND
      ENDIF
      DO RECHECK by next Data and Relation with KEEPData[]
            DO return Data[]
            ENDDO
      ENDDO
```

รูปที่ 4.14 Pseudo code การตรวจสอบเงื่อนไข OR และ AND

4.4 ระบบจัดลำดับการค้นหา (Ranking System)

4.4.1 การจัดลำดับผลของการค้นหา (Ranking)

การจัดลำดับผลของการค้นหา (Ranking) เป็นกระบวนการในการจัดเรียงเว็บเพจที่มีความสำคัญเพื่อแสดงความแม่นยำและความสำคัญของข้อมูล รวมถึงการพิจารณาข้อมูลให้ตรงกับความต้องการของผู้ใช้ เป็นการเรียงลำดับของข้อมูลตามค่าน้ำหนักและความสัมพันธ์ของข้อมูลตามเงื่อนไขของ Boolean Expression ที่มีค่าน้ำหนักมากที่สุดลดลงเรื่อยๆตามค่าความสำคัญ

การจัดลำดับผลของการค้นหา (Ranking) ของระบบการค้นหาเว็บเซอร์วิสนี้ พิจารณาจากเงื่อนไขของ Boolean expression โดยทำการให้ค่าของน้ำหนักของความสำคัญของ URL นั้นๆ ซึ่งค่าน้ำหนักนั้น อยู่ระหว่าง 0 ถึง 1 ค่าน้ำมากหมายความได้ว่าความเป็นไปได้ที่ URL นั้นๆจะตรงกับความต้องการของผู้ใช้มากที่สุด หลักการพิจารณาให้ค่าน้ำหนักของ Keyword ใน Boolean expression เป็นไปดังนี้

ตารางที่ 4.5 ค่าน้ำหนักที่ใช้พิจารณาตามเงื่อนไขของ Boolean Expression ของแต่ละข้อมูล

Keyword	ค่าน้ำหนัก
Keyword	1
Synonym	0.5
Generalization	0.5
Specialization	0.5
Association	ค่าจาก Similarity Hierarchy

ทำการคำนวณหาค่าน้ำหนักตามเงื่อนไขของ Boolean Expression จะได้ URLหลายตัวที่มีเงื่อนไขแตกต่างกันออกไป และมีค่าน้ำหนักที่แตกต่างกันออกไปตามเงื่อนไขเช่นกัน

ตัวอย่าง การคำนวณค่าน้ำหนัก

Resort ∪ Beach ∪ Adventure ∩ Safari ∩ has activity

Resort = keyword	ค่าน้ำหนักเท่ากับ	1
Beach =Synonym	ค่าน้ำหนักเท่ากับ	0.5
Adventure = Generalization	ค่าน้ำหนักเท่ากับ	0.5
Safari = Specialization	ค่าน้ำหนักเท่ากับ	0.5
has activity = Association	ค่าน้ำหนักได้จากค่า	Similarity Hierarchy

ผลที่ได้จากการคำนวณ มีค่าเท่ากับ 0.182 (หัวข้อ 4.4.2)

ดังนั้นค่าน้ำหนักของของ Boolean Expression นี้คือ

(1 x 1) + (1 x 0.5) + (1 x 0.5) + (1 x 0.5) + (1 x 0.128) = 2.628

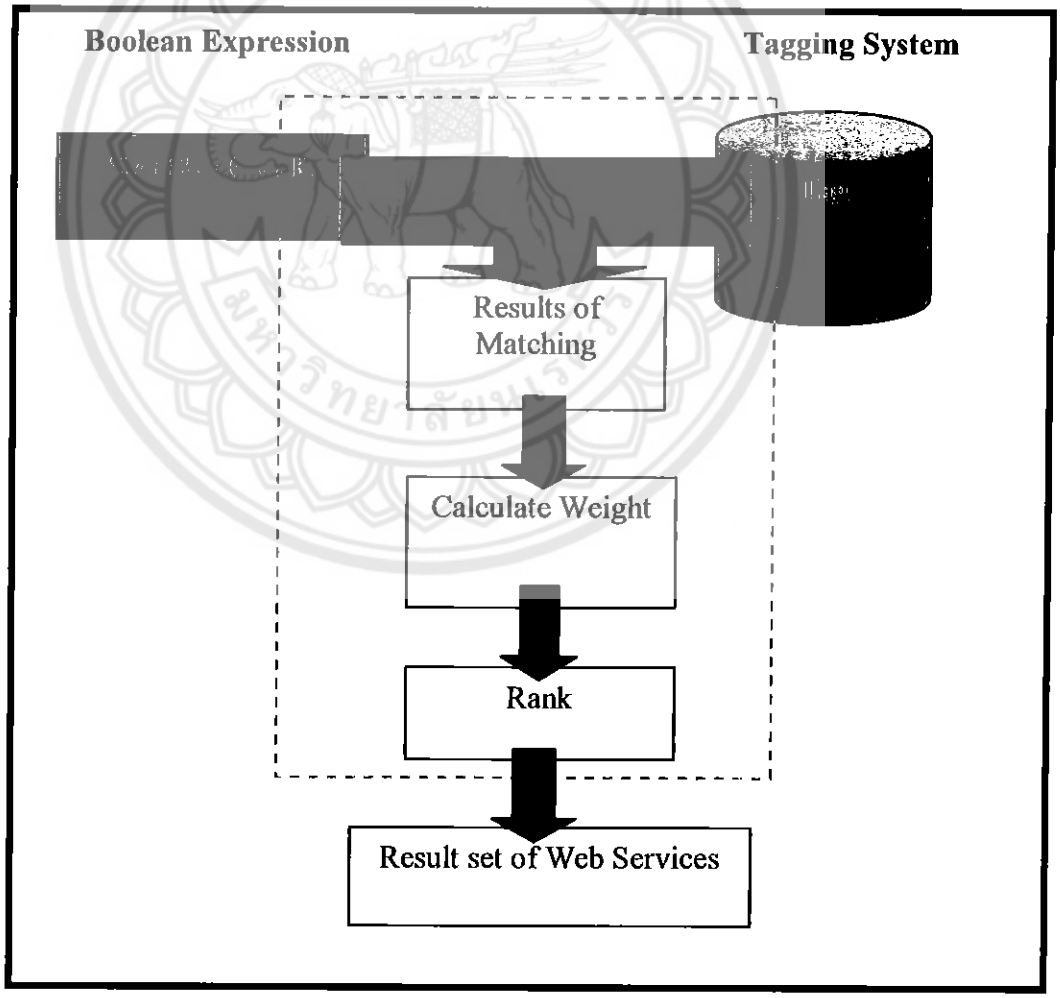
จากนั้นทำการหารค่าน้ำหนักของข้อมูลที่ได้ด้วยค่าของจำนวน Keyword ใน URL ที่มากที่สุดที่พบ Keyword จะได้ค่าน้ำหนักใหม่ ที่มีค่าอยู่ระหว่าง 0-1 เพื่อใช้ในการเรียงลำดับข้อมูล

ดังนั้นค่าน้ำหนักใหม่ที่ได้คือ **0.628/5 = 0.477121**

หมายเหตุ : กำหนดให้ ค่า Keyword ใน URL ที่มากที่สุดที่พบ มีค่าเท่ากับ 5

ดังนั้นค่าน้ำหนักของ URL นี้จากเงื่อนไขของ Boolean expression นี้ มีค่า **0.477121**

จากนั้นทำการหาค่าน้ำหนักในของทุกเงื่อนไขของ Boolean expression แล้วเรียงลำดับค่าของค่าน้ำหนักของ URL จากมากไปน้อย ค่าน้ำหนักที่ใช้ในการจัดลำดับนั้น จะได้ เซตของผลของการทำ Ranking เป็น URL ของเซอร์วิสที่มีความสัมพันธ์กับ Boolean expression สำหรับผู้ใช้เลือกไฟล์ของ URL ที่เก็บไฟล์ OWLS และไฟล์ WSDL ตามต้องการ แสดงการทำงานดังรูป 4.15



รูป 4.15 รูปแสดงการทำงานของระบบ Searching และ Ranking

4.4.2 Similarity Thought Document Hierarchy [11]

หลักการพิจารณาค่า Similarity มีหลักการในหลายวิธี ค่าที่ได้แตกต่างการไปตามหลักการวิเคราะห์ วิธี Similarity Thought Document Hierarchy พิจารณาหาค่า Similarity ด้วยการพิจารณาระดับของ Ontology นำมาวิเคราะห์หาความสัมพันธ์ของข้อมูล เหมาะสมกับการหาความสัมพันธ์ของโหนด 2 โหนดในความสัมพันธ์ของ Association ที่พิจารณา

Similarity Hierarchy เป็นอีกวิธีการหนึ่งที่สามารถหาความสัมพันธ์ความเหมือนของข้อมูลใน Ontology ได้ดีและแม่นยำ เนื่องจากความสัมพันธ์ของลำดับชั้นใน Ontology ที่บ่งบอกว่าในลำดับชั้นแบบแผนภาพต้นไม้นั้น Parent node ย่อมมีความสัมพันธ์กับ Child node เนื่องจาก Child node เป็นซับคลาสย่อยของ Parent node ดังนั้นข้อมูลจึงมีความสัมพันธ์กัน อาจมีมากหรือน้อยขึ้นอยู่กับระยะทางของโหนดนั้นๆ และความหมายโดยเฉพาะของแต่ละโหนดนั้นเอง

การหา Similarity แบบ Similarity Hierarchy นั้น เป็นการหาความสัมพันธ์ของข้อมูลที่ต้องการใน Ontology ใดๆ ที่สนใจโดยการพิจารณาโหนดร่วมที่น้อยที่สุด (The lower the common Parent node) ของ 2 ข้อมูล โดยกำหนดให้

$F(d_i,d_j)$ แทนโหนดร่วมที่น้อยที่สุด

$L(x)$ คือระดับของโหนด X

L_{Total} แทนระดับที่มากที่สุดในลำดับชั้นของออนโทโลจี

ซึ่งความคล้ายคลึงกันของ 2 ข้อมูลนั้นพิจารณาจากสูตร

$$S(d_i,d_j) = (L(F(d_i,d_j) - 1)/(L_{Total} - 1))$$

รูปที่ 4.16 สูตรการหาค่า Similarity Hierarchy

หมายเหตุ : $S(d_i,d_j) = 1$ และ $S(d_i,d_j) = 0$ ถ้า $F(d_i,d_j) = root$

หลักการนี้สามารถหาความคล้ายคลึงกันของ 2 ข้อมูลใน Ontology เพื่อได้ Similarity ของข้อมูลที่พิจารณา เพื่อใช้เป็นค่าน้ำหนักของ Keyword ใน Association เนื่องจาก Association เป็นความสัมพันธ์ของข้อมูลใน *2* ข้อมูลใน Ontology การหาค่าน้ำหนักด้วยวิธีนี้จะเป็นประโยชน์และได้ข้อมูลที่มีความเที่ยงตรงมากขึ้น

4.4.3 Pseudo code ของการทำงานของระบบ Ranking

เมื่อระบบทำการค้นหา URL ตามเงื่อนไขของ Boolean Expression นั้น ระบบจะทำการคำนวณหาค่าน้ำหนักของ URL นั้นๆไปด้วยพร้อมกัน กล่าวคือ พิจารณาตามเงื่อนไขของ Boolean Expression ที่พิจารณาลักษณะเดียวกับการพิจารณาค้นหา URL ที่เป็นไปได้นั้น

ระบบจะทำการคำนวณหาค่าน้ำหนักโดยพิจารณาข้อมูล ตารางที่ 4.5 พิจารณา Pseudo code ของการทำงาน รูปที่ 4.17 ในส่วนการกำหนดค่าน้ำหนักของแต่ละ URL ซึ่งเปลี่ยนแปลงตามเงื่อนไขที่ต่างกันออกไปในแต่ละรอบที่พิจารณาเงื่อนไข โดยทำการพิจารณาทีละคู่ตามเงื่อนไข เลือกข้อมูลที่ตรงกับเงื่อนไขของ Boolean Expression ซึ่งยิ่งข้อมูลคำค้นหามาก ค่าน้ำหนักจะมีค่ามากเช่นเดียวกัน

```
IF DATA.Length = 1
      THEN  Data[] = Url + RankValue

ELSE IF DATA.Length = 3
      FIND URL DATA.Length 1
      FIND URL DATA.Length 2
      CHECK Relation
            DO CHECK DATA BY RELATION REPEAT WHILE codition
               DataLIST[] = Url + RankValue
            ENDDO
      ENDCHECK

ELSE IF DATA.Length > 3
      FIND URL DATA.Length 1
      FIND URL DATA.Length 2
      CHECK Relation
            DO CHECK Data by Relation REPEAT WHILE codition
               DataLIST[] = Url + RankValue
            ENDDO
      CHECK next and Relation
            FIND next URL
            CHECK next Relation
            DO CHECK Data with dataLIST[] REPEAT WHILE codition
                  DataLIST[] = Url + RankValue
            ENDDO
      ENDCHECK
ENDIF
```

รูปที่ 4.17 Pseudo code การตรวจสอบเงื่อนไข OR AND

ข้อมูลที่เป็นความสัมพันธ์แบบ Association การพิจารณาหาค่าน้ำหนักจะแตกต่างไปจากการพิจารณาในข้อมูลอื่นๆ เนื่องจาก Association เป็นความสัมพันธ์ของ 2 โหนดข้อมูลที่มีความสัมพันธ์เฉพาะใน Ontology จึงพิจารณาความสัมพันธ์เพื่อหาค่าน้ำหนักที่ของข้อมูลตัวอื่นๆ ดังที่กล่าวหัวข้อที่ 4.4.2 Pseudo code แสดงการทำงาน ดังรูปที่ 4.18

```
Check()
      IF FirstData.Length < SecondData.Length
            THEN index = FirstData.Length
      ELSE  index = SecondData.Length
      ENDIF

      REPEAT WHILE condition
      IF FirstData[X] = SecondData[X]
            THEN COUNT level FirstData
      ELSE X = index - 1
      ENDIF
      DO (level.Length - 1) / TotalLevel-1)
      ENDDO
```

รูปที่ 4.18 Pseudo code ในการหาค่าความสัมพันธ์ใน Similarity Hierarchy ของ Association

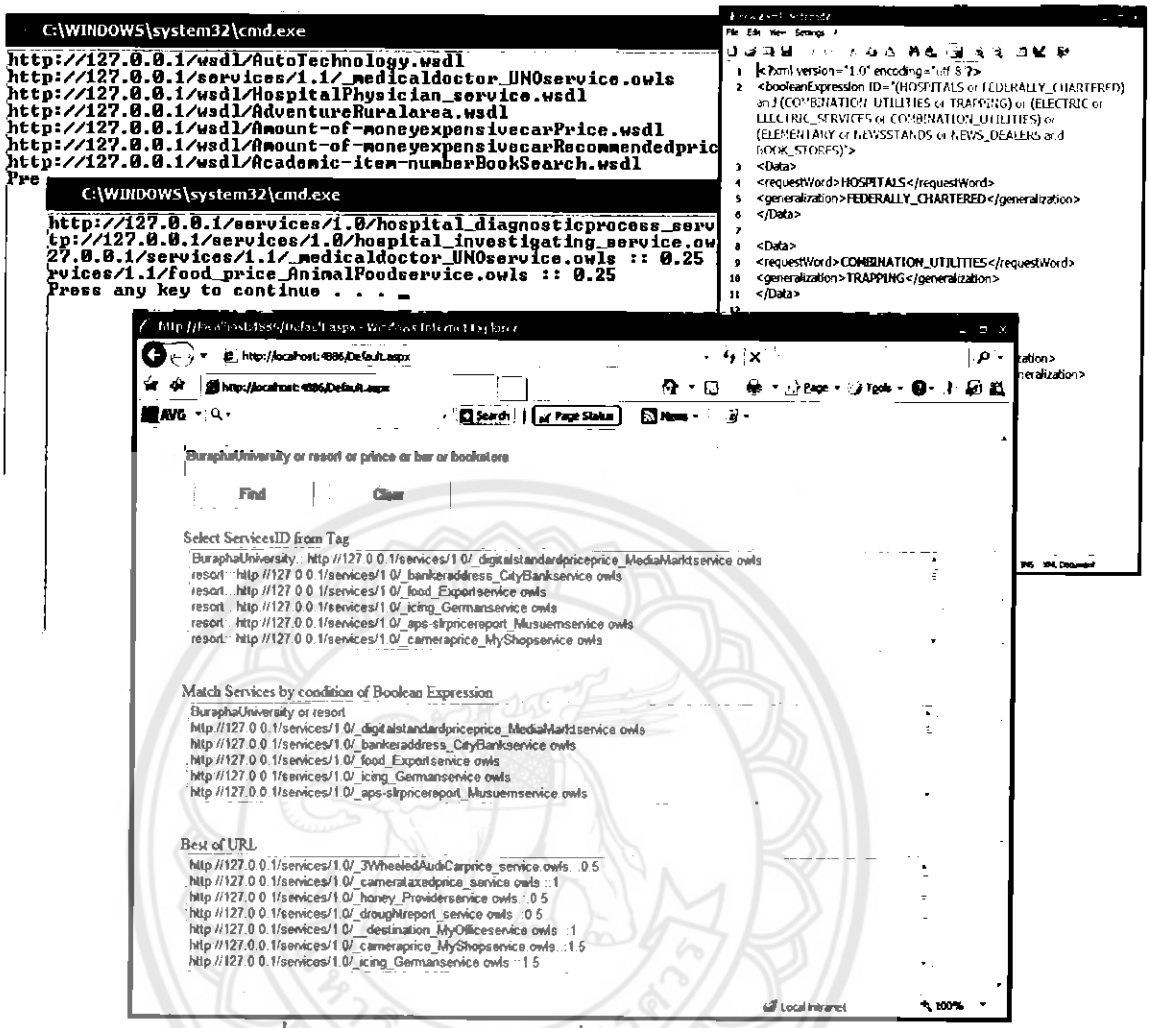
เมื่อทำการคำนวณข้อมูลหาค่าความสัมพันธ์ใน Similarity Hierarchy ของ Association ได้ตามการทำงานแล้ว จะได้ค่าน้ำหนักที่เป็นส่วนของ Association เพื่อพิจารณาตามทำเงื่อนไขร่วมกับค่าน้ำหนักตัวอื่นๆ ในการหาค่าน้ำหนักรวมของข้อมูลต่อไป

จากนั้นจะเป็นขั้นตอนสุดท้ายในการเรียงลำดับข้อมูล โดยทำการรวมค่าน้ำหนักที่ได้ของข้อมูลทั้งหมด ให้อยู่ระหว่างค่า 0-1 โดยการหารเฉลี่ยแต่ละ URL ด้วยจำนวน Keyword ที่พบมากที่สุด พิจารณาหัวข้อ 4.4.1 ทำการจัดลำดับการค้นหาจากข้อมูลที่มีค่าน้ำหนักมากที่สุดไปหาค่าน้ำหนักที่น้อยที่สุดตาม Pseudo code ดังรูป 4.19

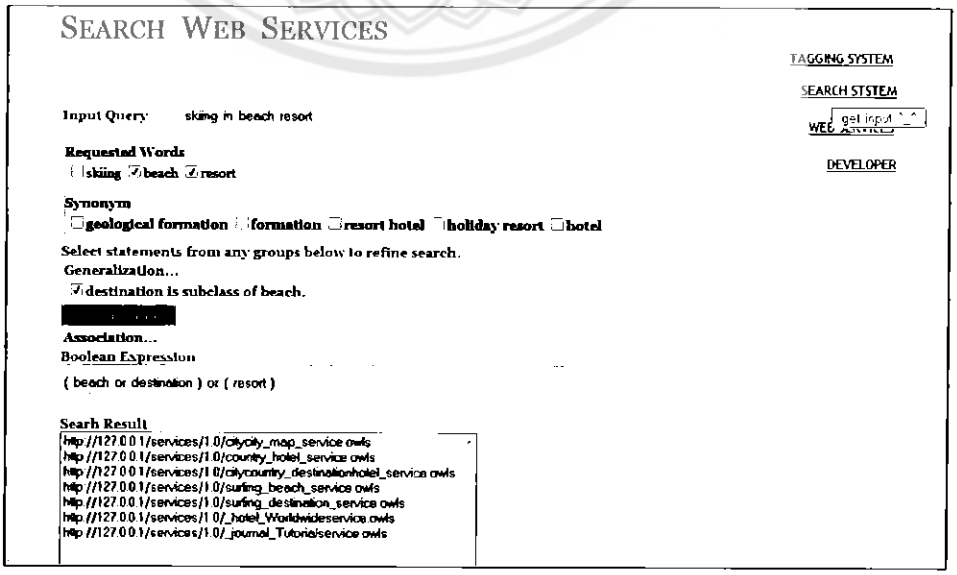
```
Rank(URLData[])
      COUNT keyword of AllURLData
      IF GetMax < URLkeyword REPEAT WHILE codition
         GetMax = URLkeyword
      ENDIF

      CHECK Weight of Url
            URLWeight/GetMax REPEAT WHILE codition
            DO ORDERBY all URLWeight
            KEEP URL and URLWeight
            ENDDO
      ENDCHECK
      DO CREATE XML file of Data
      ENDDO
```

รูปที่ 4.19 Pseudo code การเรียงลำดับข้อมูล

จากนั้นระบบจะทำการสร้างไฟล์ XML ที่ได้ของรายการ URL ที่ได้จากการค้นหาตามเงื่อนไขของ Boolean Expression และมีการจัดลำดับผลการค้นหาเรียบร้อยแล้ว แสดงดังรูป 4.20 เพื่อให้ระบบในส่วนการแสดงผลแสดงข้อมูลบริการของเว็บเซอร์วิสแก่ผู้ใช้



รูปที่ 4.20 ผลการค้นหา URL ที่มีการจัดลำดับการค้นหาแล้ว



รูปที่ 4.21 ตัวอย่างการค้นหาเว็บเซอร์วิส โดยการกรอกข้อมูลคำค้นหาและแสดงผลการค้นหา

บทที่ 5
ผลการทดลอง

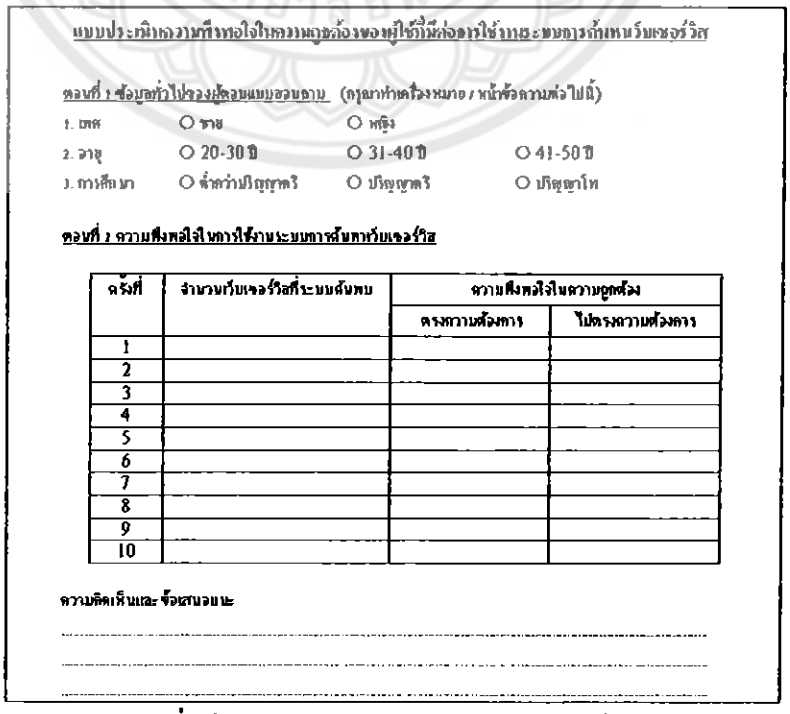
5.1 การทดสอบระบบค้นหา (Search System)

การทดสอบระบบค้นหาเป็นการวัดประสิทธิภาพด้านการทำงานของระบบ ในการทำงานการค้นหาเว็บเซอร์วิส เพื่อเป็นการทดสอบให้ได้ระบบที่สามารถตอบสนองความต้องการของผู้ใช้ในการค้นหาข้อมูลได้มีความคล่องตัวในการใช้งานร่วมกับระบบอื่นๆในการทำงานให้มีประสิทธิภาพได้

5.1.1 วิธีการทดสอบระบบการค้นหาเว็บเซอร์วิส

สำหรับการทดสอบประสิทธิภาพของระบบค้นหาเว็บเซอร์วิส(Search System) ใช้การทดสอบใน 2 ประเด็น คือ ด้านความถูกต้องของข้อมูลในการค้นหาและความเร็วในการทำงานในการค้นหาเว็บเซอร์วิส เพื่อวัดความถูกต้องของข้อมูล URL ที่สามารถค้นหาได้รวมถึงวัดความรวดเร็วในการตอบสนองความต้องการในการค้นหาเว็บเซอร์วิส

5.1.1.1) การวัดประสิทธิภาพการทำงานด้านความถูกต้องของข้อมูลในการค้นหา สำหรับการทดสอบประสิทธิภาพของระบบค้นหาเว็บเซอร์วิสด้านความถูกต้องของข้อมูลที่ต้องการ ใช้การทดสอบในรูปแบบการสอบถามด้วยแบบสอบถาม โดยการประเมินการใช้งานโดยผู้ใช้ เพื่อวัดประสิทธิภาพด้านความถูกต้องของการทำงานของระบบ



แบบประเมินความพึงพอใจในความถูกต้องของผู้ใช้ที่มีต่อการใช้งานระบบการค้นหาเว็บเซอร์วิส

ตอนที่ 1 ข้อมูลทั่วไปของผู้ตอบแบบสอบถาม (กรุณาทำเครื่องหมาย / หน้าข้อความต่อไปนี้)

1. เพศ ○ ชาย ○ หญิง

2. อายุ ○ 20-30 ปี ○ 31-40 ปี ○ 41-50 ปี

3. การศึกษา ○ ต่ำกว่าปริญญาตรี ○ ปริญญาตรี ○ ปริญญาโท

ตอนที่ 2 ความพึงพอใจในการใช้งานระบบการค้นหาเว็บเซอร์วิส

ครั้งที่	จำนวนเว็บเซอร์วิสที่ระบบค้นพบ	ความพึงพอใจในความถูกต้อง	
		ตรงความต้องการ	ไม่ตรงความต้องการ
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

ความคิดเห็นและข้อเสนอแนะ

.....

.....

.....

รูปที่ 5.1 แบบสอบถามที่ใช้ในการวัดประสิทธิภาพการทำงานด้านความถูกต้องในการค้นหา

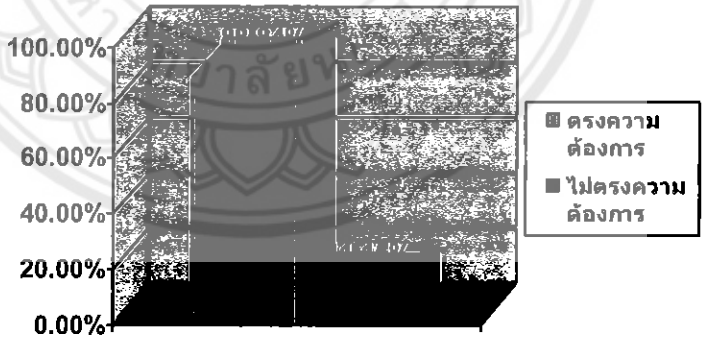
การประเมินการทำงานของระบบนั้น ได้ทำการตรวจสอบความพึงพอใจในการใช้งานของผู้ใช้และผลของการค้นหาจากหน้าการค้นหา โดยพิจารณา URL ที่ได้จากการค้นหา ว่ามีผลตรงกับความต้องการมากน้อยเพียงใด พิจารณาตามผลการทดสอบ แสดงแบบทดสอบในการทดสอบผลการค้นหา ดังรูป 5.1

5.1.1.2) การวัดประสิทธิภาพการทำงานด้านความเร็วในการค้นหา สำหรับการทดสอบประสิทธิภาพด้านความเร็วในการค้นหาของระบบค้นหาเว็บเซอร์วิส ใช้การทดสอบในรูปแบบการสุ่มเวลาที่ใช้ในการค้นหาจากระบบการค้นหาจำนวนทั้งสิ้น 10 ครั้ง เพื่อทำการวิเคราะห์หาค่าความเร็วเฉลี่ยที่ใช้ในการค้นหา เพื่อวัดประสิทธิภาพการทำงานของระบบ

5.1.2 ผลการทดสอบ

ตารางที่ 5.1 ผลการทดสอบการวัดประสิทธิภาพ
การทำงานด้านความถูกต้องในการค้นหา

[illegible]	[illegible]	
[illegible]	[illegible]	[illegible]
278	247	31
100%	88.84%	11.15%



รูปที่ 5.2 แผนภูมิผลการทดสอบการวัดประสิทธิภาพการทำงานด้านความถูกต้องในการค้นหา

ตารางที่ 5.1 และรูปที่ 5.2 แสดงระดับความพึงพอใจของผู้ใช้ในการใช้งานของระบบการค้นหา ในด้านความถูกต้องของการค้นหาข้อมูลเว็บเซอร์วิส ในการให้บริการของระบบ โดยทำการวัดประสิทธิภาพและความพึงพอใจในข้อมูลที่ถูกต้องจากผู้ใช้งานจำนวน 10 คน โดยวัดข้อมูล URL ที่พบ ในการใส่คำค้นหาของผู้ใช้ ว่าตรงกับความต้องการมากน้อยเพียงใด ซึ่งจะแสดง

ประสิทธิภาพในการค้นหาให้เห็นได้เป็นอย่างดี ข้อมูลแสดงให้เห็นว่า ผลของการค้นหามีความถูกต้องตามระดับความพึงพอใจของผู้ตอบแบบสอบถาม มีเพียงแค่ 11.15% ที่เห็นว่าข้อมูลไม่ตรงกับความต้องการ แต่ข้อมูลที่ตรงความต้องการมีถึง 88.84% แสดงให้เห็นอย่างชัดเจนว่าข้อมูลที่ได้ตรงกับความต้องการของผู้ใช้งาน

ตารางที่ 5.2 ผลการทดสอบประสิทธิภาพการทำงาน
ด้านความเร็วในการค้นหา

ครั้งที่	เวลาที่ใช้ในการประมวลผลการค้นหา (วินาที)
1	0.53
2	0.96
3	1.02
4	1.04
5	1.56
6	0.96
7	2.03
8	0.65
9	1.32
10	1.13
เฉลี่ย	1.02

ตารางที่ 5.2 แสดงระดับความเร็วในการค้นหาแต่ละครั้งในการค้นหา ในการค้นหาเว็บเซอร์วิส โดยทำการวัดประสิทธิภาพความเร็วในการค้นหาจำนวน 10 ครั้ง โดยวัดด้วยการจับเวลาในการทำงาน ผลที่ได้จะมีความเร็วของการทำงานที่ต่างกันไป และระดับความเร็วของการค้นหาเฉลี่ยประมาณ 1.02 วินาที

5.1.3 สรุปผลการทดสอบ

จะเห็นได้ว่าประสิทธิภาพของการค้นหาเว็บเซอร์วิสนั้นสามารถค้นหาได้ดีตามความหมายของประโยค Boolean Expression ที่มาจากผู้ใช้กรอกในหน้าการค้นหา ระบบนี้สามารถใช้แทนระบบการค้นหาอื่นๆได้ มีความสะดวกสบาย รวดเร็วเพียงพอสำหรับการค้นหาเว็บเซอร์วิสที่ต้องการ สะดวกและรวดเร็วมากกว่าการค้นหาข้อมูลด้วยตนเอง และมีข้อมูลที่หลากหลาย มีการ

เรียงลำดับตามความสำคัญ ข้อมูลมีความถูกต้องแม่นยำพอสมควร ซึ่งข้อมูลที่ได้จากการค้นหานั้นอาจขึ้นอยู่กับความหมายของแท็กในฐานข้อมูลและการเลือกข้อมูลคำค้นหาสำหรับการค้นหาด้วย การทดสอบประสิทธิภาพของการค้นหาเว็บเซอร์วิสนี้ จึงสรุปได้ว่าประสิทธิภาพในการค้นหาอยู่ในระดับที่ดี สามารถนำข้อมูลเว็บเซอร์วิสที่ได้ไปใช้งานได้

บทที่ 6
บทสรุป

6.1 สรุปผลการดำเนินการ

โครงงานนี้พัฒนาขึ้นเพื่อศึกษาและทดลองเกี่ยวกับหลักการทำงานของระบบการค้นหาเว็บเซอร์วิส ซึ่งมุ่งเน้นไปที่การค้นหาข้อมูลบริการเว็บเซอร์วิสที่ผู้ใช้ต้องการ แล้วแสดงผลให้ได้ตรงตามความต้องการที่ผู้ใช้งานต้องการ โดยการใช้เงื่อนไขของ Boolean Expression ที่ได้จากการวิเคราะห์คำค้นหาของผู้ใช้เข้ามาช่วยในการค้นหา เพื่อเปรียบเทียบคำค้นหากับข้อมูลในฐานข้อมูลที่ได้จากแท็กที่ผู้ให้บริการเว็บเซอร์วิส (Provider) ที่กำหนดขึ้น โดยแท็กนั้นได้มาจาก NAICS Ontology และแท็กที่กำหนดเองโดยผู้ใช้ (User Defined Tag) จาก NAICS Ontology เพื่อให้ได้ URL ของบริการที่ต้องการ โดยข้อมูลที่ได้ถูกต้องและตรงความต้องการมากที่สุด แล้วทำการแสดงผลออกมาโดยเรียงลำดับข้อมูลตามลำดับความสำคัญ

การพัฒนาระบบการค้นหาเว็บเซอร์วิส สามารถสรุปได้ว่าระบบการค้นหาเว็บเซอร์วิส ที่พัฒนาขึ้นสามารถค้นหาข้อมูลเว็บเซอร์วิสได้จึง และสามารถทำได้ดี ตรงกับความต้องการของผู้ใช้งานในการค้นหา และผลของการค้นหาจะขึ้นอยู่กับ Boolean Expression ที่ใช้ในการค้นหาและข้อมูลของแท็กที่มีในฐานข้อมูล สามารถใช้แทนการค้นหาได้ดีเมื่อเทียบกับระบบอื่นๆ

โครงงานนี้เป็นโปรแกรมที่พัฒนาขึ้นโดยใช้ Microsoft Visual C# .NET และ ASP.NET เป็นเว็บเซอร์วิสที่พัฒนาร่วมกันใน 3 ระบบ คือ ระบบแสดงผล (Interface System) ระบบฐานข้อมูลแท็ก (Tagging System) และระบบค้นหา (Search System) เพื่อสร้างระบบการค้นหาเว็บเซอร์วิสที่สามารถตอบสนองต่อการค้นหาได้

6.2 ปัญหาและอุปสรรคในการพัฒนา

6.2.1 ผลของการทำงานระบบการค้นหาเว็บเซอร์วิส แปรผันกับคำที่ใช้เลือกในการค้นหาและข้อมูลของแท็กที่มีในฐานข้อมูล หากแท็กที่ได้จากฐานข้อมูลมีความหมายไม่ตรงกับบริการนั้นๆ ตามที่ผู้ใช้ต้องการ การค้นหาจะได้บริการที่ไม่ถูกต้องเท่าที่ควร

6.2.2 องค์ประกอบและสิ่งแวดล้อมของฐานข้อมูลและ Boolean Expression ส่งผลต่อคุณสมบัติของการค้นหาที่ได้ เช่น ความหมายของแท็ก จำนวนข้อมูลคำค้นหาที่ผู้ใช้เลือก ความยาวของประโยค Boolean Expression เป็นต้น ซึ่งหากผู้ใช้เลือกข้อมูลมากเกินไปทำให้ Boolean Expression ยาวมากขึ้น ทำให้การเปรียบเทียบข้อมูลและค่าน้ำหนักจะมากขึ้นส่งผลต่อความหมายของข้อมูลที่สืบค้นได้ ทำให้ผลการค้นหาที่ได้เปลี่ยนตามไป

6.3 ข้อเสนอแนะ

6.3.1 ศึกษาการพัฒนาระบบด้วย Microsoft Visual C# .NET ให้มีความละเอียดกว่านี้

6.3.2 พัฒนาระบบการค้นหาเว็บเซอร์วิสให้รองรับคำค้นหาที่หลากหลายและเจาะจงในการค้นหาเว็บเซอร์วิสได้

6.4 แนวทางในการพัฒนาต่อในอนาคต

6.4.1 สามารถประยุกต์ใช้กับการสืบค้นอื่น ๆ นอกจากการค้นหาเว็บเซอร์วิสได้

6.4.1 สามารถพัฒนาการสืบค้นข้อมูลเว็บเซอร์วิสด้วยภาษาไทยและภาษาอื่นๆได้

6.4.1 สามารถสร้างฐานข้อมูลด้วยการเก็บข้อมูลด้วยครอวเลอร์ได้

เอกสารอ้างอิง

[1] Yang Y., and Wang H.,**Implementation of Focused Crawler,** HKUST, Hongkong.2000

[2] Raghu R.,and Johanes G.**Database management system.**,McGrawHill,pp.926-966.2003

[3] Nurulhuda F.,Zeti D.,Nuzulha K., and Rozanna D.**Development of Appointment Scheduling Agent Using Distributed Constraint Satisfaction (DisCS),**Universiti Teknologi Malaysia.2008

[4] Webmaster.**"LinqtoSQL".**[http://greatfriends.biz/webboards/msg.asp?id=50209].2009

[5] Yoo Jung .**ONTOLOGY LEARNING FOR THE SEMANTIC DEEP WEB,** New Jersey Institute of Technology].2008

[6]Webmaster. **"เว็บเซอร์วิส และพื้นฐานการสร้างเว็บเซอร์วิส".**[http://truehits.net/faq/webmaster/webservice/].2008

[7] Webmaster.**"Ontology คืออะไร".**[http://tewwss.multiply.com/journal/item/1/1].2011

[8] Webmaster. **"OWL".** [http.multiply.com/journal/item/1/2]. 2011

[9] Webmaster.**"Ontology".**[http://guru.google.co.th/guru/thread?tid=679ae93eef7c5fb5].2011

[10] Webmaster.**"XML".**[http://guru.google.co.th/guru/thread?tid=456ed0bfbb5a00a1]. 2011

[11] Webmaster.**"Boolean operator".**[http://kanjanachonfunfunhp.spaces.live.com/blog/cns!8DF845673B2A42A9!538.entry].2009

ประวัติผู้เขียนโครงงาน

ชื่อ นางสาวกฤษณา ไชยยา

ภูมิลำเนา 39 ม. 6 ต.ธารทอง อ.พาน จ.เชียงราย 57250

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนแม่ลาววิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
  สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

E-mail : **maruko2001@hotmail.com**

ชื่อ นายศราวุฒิ โยชนะ

ภูมิลำเนา 8 ม.5 ต.วังตะแบก อ.พรานกระต่าย จ.กำแพงเพชร 62110

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนพรานกระต่ายพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4
  สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
  มหาวิทยาลัยนเรศวร

E-mail : **nonlablaw@hotmail.com**