การพัฒนาระบบสืบค้นเชิงความหมายด้วยวิธีอ็อบเจกต์ออนโทโลยีแมปปิง

นายสุทธิรักษ์ แสงจันทร์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2555

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย
บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)
are the thesis authors' files submitted through the Graduate School.

# DEVELOPMENT OF SEMANTIC SEARCH USING OBJECT-ONTOLOGY MAPPING

Mr. Sutthirak Sangchun

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2012

Copyright of Chulalongkorn University

| | |
|---|---|
| หัวข้อวิทยานิพนธ์ | การพัฒนาระบบสืบค้นเชิงความหมาย<br>ด้วยวิธีอ็อบเจกต์ออนโทโลยีแมปปิง |
| โดย | นายสุทธิรักษ์ แสงจันทร์ |
| สาขาวิชา | วิทยาศาสตร์คอมพิวเตอร์ |
| อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก | รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี |

---

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

…….. คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

…… ประธานกรรมการ
(รองศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา)

…….. อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี)

…….. กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ธนารัตน์ ชลิดาพงศ์)

…… กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.วีระ บุญจริง)

สุทธิรักษ์ แสงจันทร์ : การพัฒนาระบบสืบค้นเชิงความหมายด้วยวิธีอ็อบเจกต์ออนโทโลยีแมปปิง. (DEVELOPMENT OF SEMANTIC SEARCH USING OBJECT-ONTOLOGY MAPPING) อ. ที่ปรึกษาวิทยานิพนธ์หลัก: รศ.ดร.พรศิริ หมื่นไชยศรี, 70 หน้า.

ระบบค้นหาโดยส่วนใหญ่ดำเนินการโดยทำการจับคู่ระหว่างคำค้นและข้อมูลจากระบบฐานข้อมูล โดยตรวจสอบจากตัวอักษรและความหมายว่าเหมือนหรือแตกต่างกันหรือไม่ ซึ่งวิธีการดังกล่าวนี้ทำให้ระบบสามารถค้นหาได้เพียงข้อมูลที่มีความสัมพันธ์ทางภาษาได้เท่านั้น แต่จะไม่สามารถค้นหาข้อมูลในกรณีที่คำค้นและข้อมูลไม่มีความสัมพันธ์กันทางภาษาแต่มีความสัมพันธ์กันในองค์ความรู้อื่นได้ ซึ่งวิธีการที่จะทำให้ระบบจะสามารถค้นหาข้อมูลในลักษณะดังกล่าวได้นั้น จะต้องอาศัยข้อมูลเพิ่มเติมมาช่วยในอธิบายความสัมพันธ์ระหว่างคำค้นและข้อมูลสำหรับการค้นหา วิทยานิพนธ์นี้จึงขอนำเสนอวิธีการพัฒนาระบบค้นหาเชิงความหมายด้วยวิธีการอ็อบเจกต์ออนโทโลยีแมปปิงหรือโอโอเอ็ม ซึ่งมีองค์ประกอบสำหรับแก้ไขปัญหาในการนำออนโทโลยีมาใช้ในการพัฒนาระบบค้นหาเชิงความหมายและสอดคล้องแนวทางในการพัฒนาโปรแกรมเชิงวัตถุ นอกจากนั้นแล้วผู้วิจัยได้นำวิธีการโอโอเอ็มไปพัฒนาเป็นระบบค้นหาข้อมูลสิ่งมีชีวิตประเภทสัตว์สะเทินน้ำสะเทินบกของประเทศไทย (ไทส์) ที่สามารถค้นหาข้อมูลของสิ่งมีชีวิตได้ทั้งจากชื่อทางวิทยาศาสตร์หรือชื่อสามัญ สามารถจัดกลุ่มความสัมพันธ์ทางด้านชีววิทยาของสิ่งมีชีวิต รวมถึงเพิ่ม ปรับปรุง แก้ไขข้อมูลของออนโทโลยีที่ใช้ในระบบค้นหาเพื่อรองรับการนำองค์ความรู้ในการจัดจำแนกสิ่งมีชีวิตใหม่ๆ มาร่วมใช้ในการค้นหาในอนาคต

ในส่วนของการประเมินผล ผู้วิจัยได้ทำการประเมินผลวิธีการโอโอเอ็มใน 2 ด้าน คือ ทางด้านประสิทธิผลและประสิทธิภาพ โดยด้านประสิทธิผลพบว่า วิธีการโอโอเอ็มสามารถให้ผลการค้นคืนได้อย่างถูกต้องและในส่วนของประสิทธิภาพจะพบว่าวิธีการโอโอเอ็มจะมีประสิทธิภาพสูงขึ้นเมื่อนำไปพัฒนาร่วมกับองค์ประกอบอื่นในการพัฒนาซอฟต์แวร์ เช่น การเก็บข้อมูลถูกประมวลผลด้วยวิธีการโอโอเอ็มเรียบร้อยแล้วไว้ในแคชของระบบ เพื่อเพิ่มความเร็วในการเรียกใช้ ข้อมูลในครั้งต่อไป

ภาควิชา.....วิศวกรรมคอมพิวเตอร์.....ลายมือชื่อนิสิต.....
สาขาวิชา..วิทยาศาสตร์คอมพิวเตอร์..... ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก.....
ปีการศึกษา. 2555.....

## 5271443621 : MAJOR COMPUTER SCIENCE

KEYWORDS : SEMANTIC WEB / ONTOLOGY / ANNOTATION

SUTTHIRAK SANGCHUN : DEVELOPMENT OF SEMANTIC SEARCH USING OBJECT-ONTOLOGY MAPPING. ADVISOR : ASSOC. PROF. PORNSIRI MUENCHAISRI, 70 pp.

The algorithm of most searching systems simply matches between input search queries and a collection of data in the databases by comparing text similarity. Nowadays, new knowledge has been continuously and immensely established. Therefore, definition and relationship of vocabularies vary in each field of knowledge. As a result, the searching algorithm using only text similarity comparison cannot find other relevant information of the searching word. This research introduces a method to develop a searching system capable of matching the searching word and its related definitions by applying Object-Ontology Mapping (OOM) approach. Researcher applies OOM to develop the system called Thailand Amphibian Information System (TAIS). TAIS assists users to effectively search all the Amphibians found in Thailand. The system searching can be done by either specifying scientific or common names of any Amphibian. The results from TAIS significantly show the relationship between creatures in the group and Biology classification. In addition, the ontology used along with database searching can be either added or modified in order to support new knowledge of Biology classification in the future.

In the evaluation, researcher evaluates effectiveness and efficiency of the OOM approach. The effectiveness evaluation shows that OOM approach has the correct result and effectiveness will increase by using the other software components such as cache.

Department : Computer Engineering Student's Signature .....

Field of Study : Computer Science Advisor's Signature .....

Academic Year : 2012

# กิตติกรรมประกาศ

ข้าพเจ้าขอขอบพระคุณ รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี อาจารย์ที่ปรึกษาวิทยานิพนธ์ฉบับนี้ที่เสียสละเวลาในการคำปรึกษา คำแนะนำ ข้อเสนอแนะต่างๆ จนกระทั้งทำให้วิทยานิพนธ์เล่มนี้สามารถสำเร็จลงได้

ขอขอบพระคุณรองศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา ประธานกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.ธนารัตน์ ชลิดาพงศ์ และ รองศาสตราจารย์ ดร.วีระ บุญจริง กรรมการสอบวิทยานิพนธ์ที่ให้กรุณาให้คำแนะนำต่างๆ ที่ทำให้วิทยานิพนธ์ฉบับนี้มีความถูกต้องและสมบูรณ์เพิ่มเติมมากยิ่งขึ้น

ขอขอบพระคุณคณาจารย์ทุกท่านที่ช่วยอบรมสั่งสอน ให้ความรู้ในเรื่องต่างๆแก่ข้าพเจ้าตลอดระยะเวลาการศึกษาที่ผ่านมานี้

ขอขอบพระคุณคุณพ่อ คุณแม่ และผู้มีพระคุณทุกท่านที่คอยให้กำลังใจ และสนับสนุนให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงลงได้

ข้าพเจ้าหวังเป็นอย่างยิ่งว่าวิทยานิพนธ์ฉบับนี้จะเป็นประโยชน์ต่อผู้นำไปศึกษาหรือผู้ที่สนใจ หากมีข้อผิดพลาดประการข้าพเจ้าขอน้อมรับไว้เพื่อนำไปปรับให้วิทยานิพนธ์ฉบับนี้มีความสมบูรณ์มากยิ่งขึ้นต่อไป สำหรับความดีที่ข้าพเจ้าได้รับจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้แก้ผู้มีพระคุณแก่ข้าพเจ้าทุกๆท่าน

# สารบัญ


หน้า

บทคัดย่อภาษาไทย..... ง

บทคัดย่อภาษาอังกฤษ..... จ

กิตติกรรมประกาศ..... ฉ

สารบัญ..... ช

สารบัญตาราง..... ญ

สารบัญภาพ..... ฎ

บทที่

**1 บทนำ**..... **1**

1.1 ความเป็นมาและความสำคัญของปัญหา..... 1

1.2 วัตถุประสงค์ของการวิจัย..... 2

1.3 ขอบเขตของการวิจัย..... 2

1.4 ขั้นตอนการวิจัย..... 3

1.5 ประโยชน์ที่คาดว่าจะได้รับ..... 3

1.6 ผลงานตีพิมพ์..... 4

**2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง**..... **5**

2.1 ทฤษฎีที่เกี่ยวข้อง ..... 5

2.1.1 ออนโทโลยี (Ontology) ..... 5

2.1.2 อาร์ดีเอฟ (RDF)..... 5

2.1.3 โอดับเบิลยูแอล (OWL)..... 5

2.1.4 สปาร์เคิล (SPARQL)..... 6

2.1.5 อาปาเชเจนา (Apache Jena)..... 6

2.1.6 จาวา (Java)..... 7

2.1.7 แอนโนเตชัน (Annotation)..... 8

2.1.8 สปริงเฟรมเวิร์ก (Spring Framework)..... 8

2.1.9 กูเกิลแอฟเอนเจิน (Google App Engine)..... 9

2.1.10 ผลการค้นคืน (Recall)..... 10

2.1.11 ความถูกต้องจากการค้นคืน (Percision)..... 10

บทที่ หน้า

2.2 งานวิจัยที่เกี่ยวข้อง..... 10
2.2.1 วิธีการสร้างโดเมนของออนโทโลยีโดยใช้รีซอร์สโมเดลและเจนา (An Approach of Domain Ontology Construction based on Resource Model and Jena) ..... 10
2.2.2 การรวมการประมวลผลอาร์ดีเอฟในโปรแกรมภาษา (Integration of RDF Processing into a Programming Language)..... 12
2.2.3 การวิจัยเรื่องการใช้เจนาในการพัฒนาระบบวิเคราะห์พฤติกรรมความฉลาดในการเรียนรู้เชิงความหมายแบบออนไลน์ (The Research of Using Jena in the Semantic-based Online Learning Intelligent Behavior Analysis System)..... 13
2.2.4 ประโยชน์ในการบำรุงรักษาซอฟต์แวร์ด้วยการเขียนโค้ดแบบ แอนโนเตชันดริเวน (Software Maintainability Benefits From Annotation-Driven Code) 14
**3 การออกแบบวิธีการโอโอเอ็ม**..... **15**
3.1 ออนโทโลยีดาต้าแอนโนเตชัน (Ontology Data Annotation) หรือโอดีเอ..... 16
3.2 ออนโทโลยีดาต้าเบสเมเนจเมนท์ซิสเต็ม (Ontology Database Management System) หรือโอดีบีเอ็มเอส(ODBMS)..... 19
3.3 อ็อบเจกต์แมปเปอร์ (Object Mapper)..... 20
**4 การพัฒนาวิธีการโอโอเอ็ม**..... **23**
4.1 พัฒนาโอดีเอ..... 23
4.2 พัฒนาโอดีบีเอ็มเอส..... 24
4.3 พัฒนอ็อบเจกต์แมปเปอร์..... 27
**5 การใช้งานวิธีการโอโอเอ็ม**..... **28**
5.1 อัพโหลดออนโทโลยีเข้าสู่ระบบ..... 29
5.2 สร้างและทดสอบสปาร์เคิลที่ต้องการนำไปใช้งาน..... 29
5.3 สร้างคลาสสำหรับใช้งาน..... 30
5.4 ระบุโอดีเอภายในคลาส..... 31
5.5 ระบุคลาสพาธและคีย์ของคลาสพาธภายในอ็อบเจกต์แมปเปอร์..... 32
5.6 เรียกใช้งานอ็อบเจกต์ผ่านองค์ประกอบอ็อบเจกต์แมปเปอร์..... 32
**6 การนำวิธีการโอโอเอ็มไปพัฒนาระบบค้นหา**..... **33**

บทที่ หน้า

6.1 อัพโหลดออนโทโลยีเข้าสู่ระบบ..... 33
6.2 สร้างและทดสอบสปาร์เคิลที่จะนำไปใช้งาน..... 35
6.3. สร้างคลาสสำหรับใช้งาน..... 35
6.4. ระบุโอดีเอที่จะใช้ภายในคลาส..... 37
6.5 ระบุคลาสพาธและคีย์ของคลาสพาธภายในอ็อบเจกต์แมปเปอร์..... 38
6.6 เรียกใช้งานอ็อบเจกต์ผ่านองค์ประกอบอ็อบเจกต์แมปเปอร์..... 39
**7 การทดสอบวิธีการโอโอเอ็ม**..... **43**
7.1 ด้านประสิทธิผล (Effectiveness)..... 43
7.2 ด้านประสิทธิภาพ (Efficiency)..... 44
**8 สรุปผลการวิจัยและข้อเสนอแนะ**..... **47**
8.1 ข้อสรุป..... 47
8.2 ข้อเสนอแนะ..... 47
**รายการอ้างอิง**..... **48**
**ภาคผนวก**..... **50**
ภาคผนวก ก ตารางผลการทดลองการวัดประสิทธิผลด้วยวิธีโอโอเอ็ม..... 51
ภาคผนวก ข ตารางผลการทดลองการวัดประสิทธิภาพของวิธีโอโอเอ็ม..... 53
ภาคผนวก ค ซอร์สโค้ดของเดฟฟินิชันแอนโนเตชัน..... 54
ภาคผนวก ง ซอร์สโค้ดของแมปเปอร์แอนโนเตชัน..... 55
ภาคผนวก จ ซอร์สโค้ดของพารามิเตอร์แอนโนเตชัน..... 56
ภาคผนวก ฉ ซอร์สโค้ดของพร็อพเพอร์ตีแอนโนเตชัน..... 57
ภาคผนวก ช ซอร์สโค้ดของวาริเอเบิลแอนโนเตชัน..... 58
**ประวัติผู้เขียนวิทยานิพนธ์**..... **59**

# สารบัญตาราง


| ตารางที่ | | หน้า |
|---|---|---|
| 1 | แสดงตัวอย่างผลการทดลองวัดประสิทธิผลของวิธีการโอโอเอ็ม..... | 43 |
| 2 | แสดงการเปรียบเทียบเวลาที่ใช้ในการประมวลครั้งที่ 1 และครั้งที่ 10..... | 45 |

# สารบัญภาพ


| ภาพที่ | | หน้า |
|---|---|---|
| 1 | แสดงองค์ประกอบของโอโอเอ็ม…….……..…….. | 16 |
| 2 | แสดงวิธีการใช้งานโอดีเอ…….…… | 18 |
| 3 | แสดงการทำงานของอ็อบเจกต์แมปเปอร์…… | 21 |
| 4 | การเรียกใช้งานคลาสอ็อบเจกต์แมปเปอร์…… | 21 |
| 5 | คลาสไดอะแกรมของคลาสอ็อบเจกต์แมปเปอร์…….. | 21 |
| 6 | แผนภาพกิจกรรมการพัฒนาวิธีการโอโอเอ็ม…… | 23 |
| 7 | ตัวอย่างการสร้างคลาสแอนโนเตชัน……. | 24 |
| 8 | เทคโนโลยีที่ใช้ในการพัฒนาโอดีบีเอ็มเอส…….... | 25 |
| 9 | หน้าจอสำหรับเพิ่มหรือแก้ไขออนโทโลยีโดยใช้โอดีบีเอ็มเอส…….. | 25 |
| 10 | หน้าจอสำหรับทดสอบสปาร์เคิลจากโอดีบีเอ็มเอส…….. | 25 |
| 11 | แผนภาพกิจกรรมการใช้งานวิธีการโอโอเอ็ม……..…….. | 28 |
| 12 | การอัพโหลดไฟล์ออนโทโลยีเข้าสู่ระบบ…… | 29 |
| 13 | ทดสอบการสปาร์เคิลเพื่อนำไปใช้ในการพัฒนาระบบ……. | 30 |
| 14 | การสร้างคลาสเพื่อใช้ในระบบค้นหา…….…… | 31 |
| 15 | การระบุโอดีเอภายในคลาสอ็อบเจกต์ของภาษาจาวา…… | 31 |
| 16 | การเรียกใช้งานคลาส Animalผ่านองค์ประกอบอ็อบเจกต์แมปเปอร์……..... | 32 |
| 17 | เทคโนโลยีที่ใช้ในการพัฒนาระบบไทส์……..…….. | 33 |
| 18 | แผนภาพยูสเคสของระบบไทส์…….…….. | 34 |
| 19 | คลาสไดอะแกรมของคลาสสปีชีส์…… | 36 |
| 20 | คลาสไดอะแกรมของคลาสโลคัลเนม……..…….. | 36 |
| 21 | การใช้งานโอดีเอภายในคลาสสปีชีส์…… | 37 |
| 22 | การใช้งานโอดีเอภายในคลาส LocalName …… | 38 |
| 23 | การประกาศสร้างอินสแตนซ์ของคลาสอ็อบเจกต์แมปเปอร์จากสปริงคอนเท็ค... | 39 |
| 24 | การแสดงข้อมูลของสิ่งมีชีวิตจากการสืบค้นด้วยชื่อสามัญด้วยระบบไทส์…..... | 40 |
| 25 | การจัดกลุ่มของสิ่งมีชีวิตด้วยระบบไทส์…… | 41 |
| 26 | กราฟผลการทดสอบการจัดกลุ่มของสิ่งมีชีวิตด้วยวิธีการโอโอเอ็ม…….. | 46 |

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาและความสำคัญของปัญหา

ระบบการค้นหาข้อมูลในปัจจุบันส่วนใหญ่ใช้วิธีการสืบค้นข้อมูล (Information retrieval) โดยมีหลักการคือเมื่อผู้ใช้ระบุคำค้น (Query) เข้าไปในระบบ ระบบจะนำคำค้นของผู้ใช้มาทำการเปรียบเทียบกับข้อมูลที่ถูกจัดเก็บในระบบในฐานข้อมูลว่ามีข้อมูลส่วนใดที่ใกล้เคียงกับคำค้นที่ผู้ใช้ระบุเข้ามาบ้าง โดยระบบจะทำการคำนวณความใกล้เคียงกันระหว่างคำค้นและข้อมูล จากนั้นจะทำการจัดเรียงข้อมูลที่มีความสัมพันธ์กับคำค้นมากไปน้อยคืนกลับไปให้ผู้ใช้ ซึ่งวิธีการดังกล่าวนี้ระบบจะใช้องค์ความรู้ทางด้านภาษาในกระบวนการเปรียบเทียบความใกล้เคียงกันระหว่างข้อมูลแต่ละข้อมูลกับคำค้น ซึ่งด้วยกระบวนการดังกล่าวทำให้ระบบค้นหาข้อมูลไม่สามารถค้นหาข้อมูลในกรณีที่ข้อมูลและคำค้นมีความสัมพันธ์ระหว่างกันในองค์ความรู้อื่นๆได้ ยกตัวอย่างเช่น หากเราต้องการสร้างระบบค้นหา โดยมีข้อมูลในฐานข้อมูลเป็นรายชื่อของสิ่งมีชีวิตทางด้านชีววิทยา เมื่อต้องการค้นหาว่ามีสิ่งมีชีวิตชนิดใดบ้างที่มีความใกล้เคียงกับสิ่งมีชีวิตที่มีชื่อว่า ปา บัวร์เรติ (Paa bourreti) หากใช้วิธีการค้นหาแบบเดิม ระบบจะไม่สามารถค้นหาข้อมูลและให้ผลการค้นคืนที่ได้ เนื่องจากระบบค้นหาใช้องค์ความรู้ทางด้านภาษาไม่สามารถใช้อธิบายได้ว่ามีข้อมูลใดบ้างที่มีความใกล้เคียงกับคำค้นที่มีชื่อว่า ปา บัวร์เรติ ทำให้ระบบค้นหาไม่สามารถค้นหาข้อมูลที่มีใกล้เคียงกันระหว่างสิ่งมีชีวิตดังกล่าวได้

จากปัญหาที่ระบบค้นหาในปัจจุบันยังไม่สามารถสืบค้นข้อมูลในกรณีที่คำค้นและข้อมูลไม่ได้มีความสัมพันธ์กันตามองค์ความรู้ทางภาษาแต่มีความสัมพันธ์กันตามองค์ความรู้อื่นๆได้ ผู้วิจัยจึงขอนำเสนอวิธีการพัฒนาระบบค้นหาเชิงความหมายด้วยวิธีการอ็อบเจกต์ออนโทโลยีแมปปิง (Object-Ontology Mapping) หรือโอโอเอ็ม (OOM) ซึ่งมีองค์ประกอบ 3 ส่วนที่ทำงานร่วมกันคือ องค์ประกอบที่ 1 ออนโทโลยีดาต้าแอนโนเตชัน (Ontology Data Annotation) หรือโอดีเอ (ODA) คือ เซ็ตของจาวาแอนโนเตชันเพื่อประกาศเรียกใช้งานออนโทโลยีจากภายในจาวาซอร์สโค้ด องค์ประกอบที่ 2 ออนโทโลยีดาต้าเบสเมเนจเมนท์ซิสเต็ม (Ontology Database Management System) หรือโอดีบีเอ็มเอส (ODBMS) ระบบหลังบ้านสำหรับการสร้าง ลบ และแก้ไข ข้อมูลของออนโทโลยีที่นำมาใช้ในการพัฒนาระบบค้นหา และองค์ประกอบสุดท้ายคืออ็อบเจกต์แมปเปอร์ (Object Mapper) ทำหน้าที่ดึงข้อมูลของออนโทโลยีที่ถูกเก็บไว้ในโอดีบีเอ็มเอส

เข้าสู่จาวาอ็อบเจกต์ตามที่ประกาศไว้ในซอร์สโค้ดด้วยโอดีเอ ซึ่งเป็นองค์ประกอบที่ออกแบบมาเพื่อให้วิธีการโอโอเอ็มสอดคล้องกับแนวทางในการพัฒนาโปรแกรมเชิงวัตถุ นอกจากนั้นแล้วผู้วิจัยยังได้อธิบายถึงขั้นตอนของการนำวิธีการโอโอเอ็มไปใช้ในการพัฒนาระบบการค้นหาเชิงความหมาย รวมทั้งยังได้นำขั้นตอนดังกล่าวไปพัฒนาเป็นระบบค้นหาเชิงความหมายเพื่อสืบค้นข้อมูลของสิ่งมีชีวิตประเภทสัตว์สะเทินน้ำสะเทินบกของประเทศไทยโดยใช้องค์ความรู้ทางการจัดจำแนกสิ่งมีชีวิตทางด้านชีววิทยา เพื่อทดสอบการใช้งานจริงของวิธีการโอโอเอ็ม อันจะเป็นตัวอย่างของการนำวิธีการดังกล่าวไปใช้ในการพัฒนาระบบค้นหาเชิงความหมายโดยใช้องค์ความรู้อื่นได้ต่อไปในท้ายที่สุดนั่นเอง

## 1.2 วัตถุประสงค์ของการวิจัย

เพื่อพัฒนาวิธีการสร้างระบบค้นหาเชิงความหมาย (Semantic Web) ในการแก้ปัญหาที่ระบบค้นหาไม่สามารถค้นหาข้อมูล ในกรณีที่คำค้นและข้อมูลมีความสัมพันธ์กันในองค์ความรู้อื่นๆ ที่นอกเหนือจากองค์ความรู้ทางภาษาได้ โดยมีองค์ประกอบที่ถูกแบบมาเพื่อให้ง่ายต่อการนำไปใช้ในการพัฒนาโปรแกรมเชิงวัตถุ และองค์ประกอบสำหรับจัดการกับข้อมูลของออนโทโลยีที่จะนำไปใช้ร่วมในการพัฒนาระบบค้นหาเชิงความหมายที่ง่ายต่อการปรับปรุงแก้ไขข้อมูลและสะดวกต่อการทดสอบการสืบค้น

## 1.3 ขอบเขตของการวิจัย

1.3.1 ระบบค้นหาเชิงความหมายที่งานวิจัยนี้ ใช้ออนโทโลยีในการอธิบายความสัมพันธ์ระหว่างข้อมูล

1.3.2 ซอฟต์แวร์ที่พัฒนาขึ้นจากงานวิจัยชิ้นถูกพัฒนาขึ้นโดยใช้ภาษาจาวาและใช้เจนาเฟรมเวิร์ก

1.3.3 ข้อมูลที่นำมาใช้ในการสร้างออนโทโลยีที่ใช้ในระบบค้นหา นำมาจากข้อมูลในการจัดจำแนกสิ่งมีชีวิตที่ถูกค้นพบในประเทศไทย ซึ่งจัดทำโดยสำนักความหลากหลายทางชีวภาพ สำนักงานนโยบายและแผนทรัพยากรธรรมชาติและสิ่งแวดล้อม กระทรวงทรัพยากรธรรมชาติและสิ่งแวดล้อม

1.3.4 ระบบค้นหาข้อมูลเชิงความหมายใช้องค์ความรู้ในการจัดจำแนกสิ่งมีชีวิตประเภทสัตว์สะเทินน้ำสะเทินบกตามองค์ความรู้ทางด้านชีววิทยาสำหรับสร้างออนโทโลยีเพื่ออธิบายการจัดจำแนกข้อมูลของสิ่งมีชีวิตในระบบ

1.3.5 ระบบค้นหาข้อมูลเชิงความหมายใช้ข้อมูลจากเอกสารการจัดจำแนกสิ่งมีชีวิตที่ถูกค้นพบในประเทศไทย ซึ่งจัดทำโดยสำนักความหลากหลายทางชีวภาพ สำนักงานนโยบายและแผนทรัพยากรธรรมชาติและสิ่งแวดล้อม กระทรวงทรัพยากรธรรมชาติและสิ่งแวดล้อมในการสร้างออนโทโลยีเพื่ออธิบายชื่อสามัญของสิ่งมีชีวิตแต่ละชนิด

## 1.4 ขั้นตอนการวิจัย

1.4.1 ศึกษาและค้นหาวิธีการแก้ไขปัญหาที่เกิดจากการพัฒนาระบบค้นหาเชิงความหมาย

1.4.2 ออกแบบและพัฒนาวิธีการแก้ไขปัญหาที่เกิดจากการพัฒนาระบบค้นหาเชิงความหมาย

1.4.3 ค้นคว้าและสร้างออนโทโลยีเพื่อใช้ในงานวิจัย

1.4.4 พัฒนาระบบค้นหาเชิงความหมายที่ใช้วิธีการที่พัฒนาขึ้นโดยใช้ภาษาจาวา

1.4.5 วัดและประเมินผลวิธีการจากระบบค้นหาเชิงความหมายที่สร้างขึ้น

1.4.6 เขียนบทความวิจัย

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

ได้วิธีการพัฒนาระบบค้นหาเชิงความหมาย ที่สามารถแก้ปัญหาระบบค้นหาในปัจจุบันที่ไม่สามารถค้นหาข้อมูลในกรณีที่คำค้นและข้อมูลมีความสัมพันธ์ตามองค์ความรู้อื่นนอกเหนือจากองค์ความรู้ทางภาษาได้ รวมทั้งมีวิธีการพัฒนาที่สอดคล้องกับแนวคิดกับการพัฒนาโปรแกรมเชิงวัตถุและมีองค์ประกอบสำหรับแก้ไขข้อมูลของออนโทโลยีที่นำไปใช้ร่วมในการพัฒนาระบบค้นหาเชิงความหมาย

## 1.6 ผลงานตีพิมพ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้นำเสนอในการประชุมวิชาการ Knowledge and Smart Technology ครั้งที่ 5 (KST-2556), Burapha University Thailand, January 21 – February 1, 2013 ในบทความเรื่อง การพัฒนาระบบซีแมนติกเสิร์ชด้วยวิธีการอ็อบเจกต์ออนโทโลยีแมปปิง กรณีศึกษาองค์ความรู้ทางด้านชีววิทยา เรื่องการจัดจำแนกสิ่งมีชีวิตประเภทสัตว์สะเทินน้ำสะเทินบก โดยผู้แต่งคือ สุทธิรักษ์ แสงจันทร์ และ รศ.ดร.พรศิริ หมื่นไชยศรี

# บทที่ 2
# ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

## 2.1 ทฤษฎีที่เกี่ยวข้อง

### 2.1.1 ออนโทโลยี (Ontology)

ออนโทโลยีเป็นสิ่งในปัจจุบันที่ถูกนำใช้เพื่อนำเสนอองค์ความรู้ในลักษณะของเซตของความสัมพันธ์ระหว่างสิ่งที่อยู่ภายในองค์ความรู้นั้นๆ ในทางวิทยาศาสตร์คอมพิวเตอร์มีการนำออนโทโลยีมาใช้ร่วมด้วยหลายสาขา เช่น ทางด้านสาขาปัญญาประดิษฐ์ (Artificial Inteligence) หรือ ทางด้านสาขาการพัฒนาระบบค้นหาเชิงความหมาย (Semantic Web) ซึ่งเป็นการนำออนโทโลยีมาร่วมใช้ในการพัฒนาร่วมกับระบบค้นหาให้มีความสามารถมากยิ่งขึ้นได้ เนื่องจากออนโทโลยีเป็นสิ่งที่สามารถใช้อธิบายองค์ความรู้ของมนุษย์ได้อย่างชัดเจนและทำให้มนุษย์สามารถสื่อสารกับระบบค้นหาได้ในลักษณะเดียวกับมนุษย์สื่อสารกับมนุษย์ [1]

### 2.1.2 อาร์ดีเอฟ (RDF)

อาร์ดีเอฟ หรือ รีซอร์สเดสคริปชันเฟรมเวิร์ก (Resource Description Freamework) เป็นรูปแบบของเอกสารที่ทางเวิร์ลไวด์เว็บคอนโสเตียม (World Wide Web Consortium) กำหนดให้เป็นรูปแบบของเอกสารมาตรฐานสำหรับอธิบายละเอียดของแหล่งข้อมูลที่ถูกใช้ในเว็บ ภายในเอกสารอาร์ดีเอฟจะประกอบไปด้วย เนื้อหาที่เรียกว่าทริเปิล (triples) ซึ่งจะประกอบไปด้วยองค์ประกอบที่ 3 ส่วน คือ ซับเจค-พรีดิเคท-ออบเจ็ค (subject-predicate-object) ทำหน้าที่อธิบายรายละเอียดของแหล่งข้อมูลนั้นๆ [2]

### 2.1.3 โอดับเบิลยูแอล (OWL)

โอดับเบิลยูแอล หรือ ออนโทโลยีเว็บแลงเกวจ (Ontology Web Language) เป็นรูปแบบมาตราฐานของเอกสารที่เวิร์ลไวด์เว็บคอนโสเตียมระบุให้เป็นรูปแบบเอกสารมาตรฐานสำหรับ

เขียนออนโทโลยีโดยมีรูปแบบของเอกสารที่ใช้รูปแบบ มาตรฐานของเอกสารแบบอาร์ดีเอฟในการอธิบายรายละเอียดของออนโทโลยี [3]

### 2.1.4 สปาร์เคิล (SPARQL)

สปาร์เคิลหรือที่ย่อมาจากชื่อเต็มว่า สปาร์เคิลโปรโตคอลแอนด์คิวรีแลงเกวจ (SPARQL Protocol and RDF Query Language) เป็นรูปแบบของภาษาสำหรับสืบค้นข้อมูลที่ถูกสร้างบนมาตรฐานอาร์ดีเอฟในลักษณะเดียวกับภาษาเอสคิวแอล (SQL) หรือสตรักเจอร์คิวรีแลงเกวจ (Structured Query Language) ที่ใช้ในการสืบค้นข้อมูลจากระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database) โดยเวิร์ลไวด์เว็บ คอนโสเตียมกำหนดให้สปาร์เคิลเป็นภาษามาตรฐานกลางสำหรับสืบค้นข้อมูลจากเอกสารที่ถูกสร้างโดยใช้โครงสร้างของเอกสารแบบอาร์ดีเอฟ [4]

### 2.1.5 อาปาเชเจนา (Apache Jena)

อาปาเชเจนาหรือเจนา เป็นเฟรมเวิร์กบนภาษาจาวาที่ถูกสร้างและพัฒนาขึ้นมาสำหรับพัฒนาระบบค้นหาเชิงความหมาย ภายในตัวเจนามีองค์ประกอบสำหรับช่วยให้นักพัฒนาสามารถจัดการกับเอกสารมาตรฐานที่เกี่ยวข้องกับออนโทโลยีที่ถูกกำหนดโดยเวิร์ลไวด์เว็บคอนโสเตียม ได้แก่ อาร์ดีเอฟ โอดับเบิลยูแอล หรือสปาร์เคิล[5]

เจนาประกอบไปองค์ประกอบต่างๆที่สำคัญ ได้แก่

1. เออาร์คิว องค์ประกอบที่ทำหน้าที่สืบค้นข้อมูลจากออนโทโลยีโดยใช้สปาร์เคิลซึ่งเป็นภาษามาตรฐานในการสืบค้นข้อมูลจากออนโทโลยี

2. ทีดีบี องค์ประกอบสำหรับบันทึกข้อมูลเอกสารอาร์ดีเอฟ สำหรับเก็บข้อมูลของออนโทโลยีเพื่อนำไปใช้ในการสืบค้น

3. เอสดีบี องค์ประกอบสำหรับบันทึกข้อมูลของออนโทโลยี ลงในฐานข้อมูลเชิงสัมพันธ์เพื่อใช้ในการสืบค้นข้อมูลของออนโทโลยี

4. เอลเออาร์คิว เป็นองค์ประกอบที่รวมความสามารถของเฟรมเวิร์กที่ใช้ สำหรับค้นหาข้อมูลเอกสารที่มีชื่อว่าอาปาเช ลูซีน (Apache Lucene) เพื่อเพิ่มประสิทธิภาพของสปาร์เคิลให้รองรับการค้นหาข้อมูลได้ดีมากยิ่งขึ้น

โครงการเจนาเริ่มต้นการพัฒนาจากบริษัทฮิวเลตต์-แพคการ์ด ซึ่งในปัจจุบัน อาปาเชเจนาเป็นโครงการโอเพนซอร์สภายใต้การดูแลของมูลนิธิ อาปาเช ซอฟต์แวร์ ฟาวน์เดชัน (Apache Software Foundation)

### 2.1.6 จาวา (Java)

จาวาหมายถึงภาษาในการพัฒนาโปรแกรมในรูปแบบเชิงวัตถุที่ถูกพัฒนาขึ้น โดยบริษัทซัน ไมโครซิสเท็ม ในปีพ.ศ. 2538 ภายใต้แนวความคิดเขียนครั้งเดียว รันได้ทุกที่ (write once , run anywhere) ภายหลังจากซันไมโครซิสเท็ม ถูกรวมกิจการเข้ากับบริษัทออราเคิล ทำให้ในปัจจุบันจาวาถูกดูแลและพัฒนาโดยบริษัทออราเคิล

จุดเด่นที่ทำให้ภาษาจาวามีคุณสมบัติเขียนครั้งเดียวใช้ได้ทุกที่นั้น เนื่องจากหลังจากที่ผู้พัฒนาทำการคอมไพล์ (Compile) ได้เรียบร้อยแล้วนั้นซอร์สโค้ดทั้งหมด จะถูกแปลงให้อยู่ในรูปไบต์โค้ดซึ่งเป็นไฟล์ของโปรแกรมที่จะสามารถรันได้โดยใช้ จาวาเวอร์ชวลแมชชีน (Java Virtual Machine) ของแต่ละระบบปฏิบัติการทำให้นักพัฒนาไม่ต้องคอมไพล์ซอร์สโค้ดใหม่ สามารถรันโปรแกรมบนระบบปฏิบัติการที่ต้องการได้ทันที [6]

โดยในภาษาจาวาได้แบ่งออกเป็นเอดิชันต่างๆสำหรับการใช้งาน โดยแบ่งเป็น เอดิชันหลัก 3 เอดิชันคือ

1. Java ME เอดิชันสำหรับพัฒนาบนอุปกรณ์เคลื่อนที่
2. Java SE เอดิชันสำหรับพัฒนาบนอุปกรณ์ทั่วไป
3. Java EE เอดิชันสำหรับพัฒนาเพื่อใช้งานระบบขนาดใหญ่

ภาษาจาวาถือได้ว่าเป็นภาษาที่ได้ความนิยมเป็นอย่างมากในการนำไปพัฒนาโปรแกรม ดังจะเห็นได้จากสถิติพบว่าภาษาจาวาได้รับความนิยมเป็นลำดับต้นๆของโลก โดยในปี 2013

ภาษาจาวาถูกจัดอันดับอยู่ในลำดับที่ 2 รองจากภาษาซีที่ได้รับความนิยมจากนักพัฒนาตามการจัดอันดับของดัชนี TIOBE [7]

### 2.1.7 แอนโนเตชัน (Annotation)

แอนโนเตชันเป็นความสามารถประการหนึ่งของภาษาจาวา ที่ให้นักพัฒนาโปรแกรมภาษาจาวา สามารถแทรกข้อมูลเข้าไปในซอร์สโค้ดได้โดยตรง[8] ทำให้มีข้อดี คือ ซอร์สโค้ดมีความซับซ้อนน้อยลง การพัฒนาโปรแกรมสามารถทำได้ง่ายขึ้นตัว ซอร์สโค้ดสามารถถูกนำกลับมาใช้ใหม่ได้ง่ายมากยิ่งขึ้น ซึ่งส่งผลทำให้การดูแลรักษาโปรแกรมสามารถทำได้ง่ายขึ้นตามไปด้วย [9] [10] [11] [12]

### 2.1.8 สปริงเฟรมเวิร์ก

สปริงเฟรมเวิร์ก (Spring Framework) หมายถึงชุดของแอพพลิเคชันเฟรมเวิร์กสำหรับใช้ในการพัฒนาซอฟต์แวร์บนภาษาจาวา โดยมีจุดเริ่มต้นในการพัฒนามาจากหนังสือชื่อ Expert One-on-One J2EE Design and Development ที่เขียนโดย Rod Johnson ในปีพ.ศ. 2545 ซึ่งถือว่าเฟรมเวิร์กดังกล่าวนี้ ได้รับความนิยมเป็นอย่างมากในการพัฒนาแอพพลิเคชันบนภาษาจาวา [13] ภายในสปริงเฟรมเวิร์กประกอบไปด้วยความสามารถต่างๆมากมายที่ถูกออกแบบมาเพื่อช่วยในการพัฒนา เช่น

1. มีองค์ประกอบสำหรับการพัฒนาด้วยเทคโนโลยี ในการพัฒนาเว็บสมัยใหม่ เช่น เฮททีเอ็มเอลไฟว์ (HTML5) , เรสเว็บเซอร์วิส (REST Webservice) หรือ เอเจ็ค (AJAX)

2. มีองค์ประกอบสำหรับการเข้าถึงข้อมูลในระบบฐานข้อมูลต่างๆ เช่น ระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database) , กลุ่มโนซีควล (NoSQL) หรือ การใช้บริการระบบฐานข้อมูลจากสถาปัตยกรรมแบบกลุ่มเมฆ

3. มีองค์ประกอบที่ถูกออกแบบมาเพื่อใช้พัฒนาเว็บสำหรับใช้งานบนอุปกรณ์เคลื่อนที่

4. มีองค์ประกอบสำหรับการพัฒนาแอพพลิเคชันร่วมกับเครือข่ายสังคมต่างๆ

5. มีองค์ประกอบสำหรับใช้ในการพัฒนาและป้องกันรวมถึงการเข้าสู่ระบบ เพื่อเข้าใช้งานแอพพลิเคชัน

6. รองรับการพัฒนาแอพพลิเคชันเพื่อทำไปในสถาปัตยกรรมแบบกลุ่มเมฆ หรือ คลาวด์คอมพิวติง (Cloud Computing)

2.1.9 กูเกิลแอฟเอนเจิน

กูเกิลแอฟเอนจิน (Google App Engine) เป็นรูปแบบการให้บริการสำหรับพัฒนาแอพพลิเคชันบนคลาวด์คอมพิวติงรูปแบบหนึ่งของบริษัทกูเกิลที่ได้รับความนิยมอย่างมากในปัจจุบัน ซึ่งมีสภาพแวดล้อมของการพัฒนาที่ถูกออกแบบมาให้นักพัฒนาสามารถพัฒนาระบบโดยใช้สภาวะแวดล้อมดังกล่าว โดยการใช้โปรแกรมภาษาในการพัฒนาได้ทั้งภาษาจาวาและภาษาไพธอน [14] นอกจากนั้นแล้วภายในกูเกิลแอพเอนจินยังมีบริการเสริมอื่นๆที่ให้ผู้พัฒนาสามารถเรียกใช้งานได้ผ่านเอพีไอ ได้แก่

1. บ็อบสโตร์ (Blobstore) ใช้สำหรับบันทึก หรือ เรียกใช้ข้อมูลที่ถูกจัดเก็บ อยู่ในฐานข้อมูล

2. อิมเมจ (Images) ใช้สำหรับการจัดการรูปภาพต่างๆ ไม่ว่าจะเป็นการขยาย หมุน หรือตัดรูปภาพ

3. เมลล์ (Mail) ใช้สำหรับการรับหรือส่งอีเมลล์

4. เมมเคช (Memchache) บริการสำหรับเรียกใช้แคชเพื่อเก็บข้อมูลบางส่วน ไว้ ทำให้การเรียกใช้งานในครั้งต่อไปสามารถทำได้รวดเร็วขึ้น

5. ทาสก์คิว (Task Queues) สำหรับใช้เพื่อประมวลผลแบบต่อเนื่องสำหรับ การประมวลผลที่ต้องอาศัยยาวนานกว่าปกติในการประมวลผล

2.1.10 ผลการค้นคืน (Recall)

ในการค้นหาข้อมูลนั้น เราสามารถคำนวณผลการค้นคืนในแต่ละครั้งของการค้นหาโดยใช้การคำนวณจากสมการ

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

จากสมการจะพบว่า ผลของการค้นคืนสามารถหาได้จากจำนวนเอกสารที่เกี่ยวข้องกับคำค้นเมื่อเทียบกับเอกสารที่ค้นคืนขึ้นมาได้ หารด้วยจำนวนของเอกสารที่เกี่ยวข้องกับคำค้นทั้งหมดนั้น ผลที่ได้จะได้ค่าผลการค้นคืนของคำค้นนั้นๆ

2.1.11 ความถูกต้องจากการค้นคืน (Percision)

ในการค้นหาข้อมูลนั้น เราสามารถค้นหาความถูกต้องจากการค้นคืนในแต่ ละคำค้นได้จากสมการ

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

จากสมการจะพบว่า เราสามารถหาความถูกต้องจากการค้นคืนได้ โดยคำนวณจากจำนวนเอกสารที่เกี่ยวข้องกับคำค้นจากจำนวนเอกสารที่ค้นคืนมาได้หารด้วยจำนวนเอกสารที่ค้นคืนมาได้ทั้งหมด ผลที่ได้จะเป็นความถูกต้องของผลการค้นคืนของคำค้นนั้นๆ

## 2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 วิธีการสร้างโดเมนของออนโทโลยีโดยใช้รีซอร์สโมเดลและเจนา (An Approach of Domain Ontology Construction based on Resource Model and Jena) โดย Shao-min Zhang, Jian-yi Guo, Zheng-tao Yu, Chun-ya Lei, Cun-li Mao, Hai-xiong Wang [15]

งานวิจัยเรื่องได้กล่าวถึงความสำคัญของออนโทโลยีว่า แนวคิดเรื่องออนโทโลยีเป็นแนวคิดที่ของสำคัญของงานวิจัยที่เกี่ยวข้องกับวิศวกรรมในการจัดการองค์ความรู้ เนื่องจากออน

โทโลยีมีความสามารถที่สำคัญ คือ สามารถบันทึกข้อมูลที่เกี่ยวข้องกับองค์ความรู้ต่างๆให้เป็นรูปธรรม และสามารถแบ่งปันองค์ความรู้เหล่านั้น จากโดเมนหนึ่งไปยังไปยังอีกโดเมนหนึ่งได้ แต่อย่างไรก็ตามการสร้างออนโทโลยีขึ้นมาใช้งานโดยการใช้เฟรมเวิร์กนั้น มีขั้นตอนหลายขั้นตอนในการสร้างที่ซ้ำซ้อนกันเป็นอย่างมากซึ่งหมายถึงกระบวนการในการสร้างและพัฒนาออนโทโลยีขึ้นมาใช้งานดังกล่าวยังคงไม่มีประสิทธิภาพนั่นเอง

ผู้วิจัยของงานวิจัยชิ้นนี้ยังเสนอวิธีการสร้างออนโทโลยีให้มีประสิทธิภาพมากยิ่งขึ้นด้วยวิธีการใช้รีซอร์สโมเดลร่วมกับเจนา ซึ่งวิธีการที่ผู้วิจัยได้เสนอดังกล่าวประกอบไปด้วย 2 ขั้นตอนคือ

1. สร้างคอนเซ็ปของโดเมนขึ้น จากนั้นทำการตรวจสอบหาความสัมพันธ์ระหว่างองค์ประกอบต่างๆภายในคอนเซ็ปนั้นๆ แล้วทำการบันทึกรายละเอียดของความสัมพันธ์ต่างๆลงในรีซอร์สโมเดล

2. เมื่อได้ความสัมพันธ์ของอ็อบเจกต์ภายในออนโทโลยีเรียบร้อยแล้ว ให้ทำการบันทึกข้อมูลทั้งหมดของออนโทโลยีที่สร้างขึ้นมานั้นโดยใช้เจนา

โดยในตอนท้ายของงานวิจัยชิ้นนี้ ผู้วิจัยได้นำวิธีการพัฒนาออนโทโลยี ดังกล่าวมาทำการทดสอบ โดยนำมาใช้สร้างออนโทโลยีเกี่ยวกับการท่องเที่ยวของเมือง ยูนนาน ซึ่งผลการทดลองพบว่าวิธีการที่ใช้สร้างออนโทโลยีดังกล่าวสามารถสร้างออนโทโลยีที่อธิบายองค์ความรู้เกี่ยวกับการท่องเที่ยวในเมืองยูนนานได้อย่างถูกต้อง

จากการศึกษางานวิจัยชิ้นนี้ ผู้วิจัยพบว่างานวิจัยชิ้นดังกล่าวนี้เป็นตัวอย่าง ของงานวิจัยที่เกี่ยวข้องกับออนโทโลยีงานวิจัยหนึ่ง ที่พยายามแก้ปัญหาความยุ่งยากในการสร้างออนโทโลยีและนำออนโทโลยีมาใช้งาน ผู้วิจัยของงานวิจัยชิ้นดังกล่าวได้เสนอวิธีการสร้างออนโทโลยีแบบใหม่โดยมีความพยายามในการลดความยุ่งยากซับซ้อนในการสร้างออนโทโลยีลง ซึ่งจากผลการทดลองในตอนท้ายก็แสดงให้เห็นว่า วิธีการดังกล่าวให้สามารถนำไปใช้ในการสร้างออนโทโลยีได้อย่างถูกต้องวิธีการหนึ่ง

แต่สิ่งหนึ่งที่งานวิจัยชิ้นนี้ยังคงขาด คือ การนำวิธีการดังกล่าวไปใช้ร่วมกับการพัฒนาระบบอื่นๆ เพื่อที่จะนำความสามารถของออนโทโลยีไปร่วมใช้เพื่อเพิ่มประสิทธิภาพการทำงาน ซึ่งประเด็นดังกล่าวนี้ถือว่าเป็นเรื่องที่มีความสำคัญ เพื่อให้ระบบอื่นๆ สามารถนำเอาความสามารถของออนโทโลยีไปใช้เพื่อเพิ่มประสิทธิภาพให้ดีมากยิ่งขึ้นได้

2.2.2 การรวมการประมวลผลอาร์ดีเอฟในโปรแกรมภาษา (Integration of RDF Processing into a Programming Language) โดย Ik-Hyun Jhin , Nak-Min และ Choi Eun-Sun Cho [16]

งานวิจัยชิ้นนี้ได้กล่าวถึงความสำคัญของเว็บเชิงความหมาย (Semantic Web) ที่ทำให้เราสามารถประมวลผลข้อมูลต่างๆบนเว็บในระดับภาษาเหมือนที่มนุษย์เข้าใจได้ ซึ่งในการที่พัฒนาให้เว็บสามารถประมวลผลข้อมูลในลักษณะดังกล่าวได้นั้น นักพัฒนาจะต้องเรียนรู้ เอพีไอที่จะต้องนำมาใช้ร่วมในการพัฒนามากมาย หนึ่งในเฟรมเวิร์กที่นักพัฒนาจะต้องเรียนรู้เพื่อนำมาช่วยในการพัฒนาก็คือเจนา ซึ่งเป็นเฟรมเวิร์กที่ได้รับความนิยมเป็นอย่างมากในการนำมาร่วมใช้พัฒนาระบบเว็บเชิงความหมาย

แต่จากการศึกษาของงานวิจัยชิ้นนี้กลับพบว่าการพัฒนาระบบเว็บเชิงความหมายด้วยเจนายัง คงมีข้อบกพร่องในประเด็นต่างๆ ได้แก่

1. การใช้งานเจนามีความซับซ้อนสูงมาก การใช้งานเฟรมเวิร์กยังต้องอาศัย ขั้นตอนหลายขั้นตอน ทำให้ซอร์สโค้ดของระบบมีความซับซ้อนสูงมาก ทำให้ซอร์สโค้ดของระบบโดยรวมยากแก่การดูแลรักษา

2. เอพีไอของเจนาซับซ้อนสูงมาก ยากต่อการใช้งาน

3. การเข้าถึงข้อมูลเพื่อนำข้อมูลมาใช้งานในการพัฒนาระบบเว็บเชิงความหมายยังต้องอาศัยขั้นตอนหลายขั้นตอน ทำให้ยากต่อการนำไปพัฒนาต่อ

จากปัญหาจากการใช้เจนาข้างต้น ทำให้ผู้วิจัยงานวิจัยชิ้นเสนอวิธี แก้ปัญหาดังกล่าว โดยเสนอภาษาสคริปรูปแบบใหม่ที่เรียกว่าเจล (Jey) ที่มีรูปแบบการใช้งานที่ง่ายขึ้น เพื่อลดขั้นตอนในการเรียกใช้งานแทนการใช้ผ่านเจนาแบบเดิม เพื่อให้สามารถจัดการข้อมูลของเอกสารอาร์ดีเอฟได้ง่ายมากยิ่งขึ้น ทำให้สามารถพัฒนาต่อเป็นระบบค้นหาเชิงความหมายต่างๆ ได้ง่ายดายมากยิ่งขึ้นนั่นเอง

แต่จากการศึกษางานวิจัยชิ้นดังกล่าวนี้พบว่าสิ่งที่งานวิจัยนี้ยังคงขาด คือ วิธีการนำภาษาสคริปดังกล่าวไปพัฒนาร่วมกับโปรแกรมภาษาที่ได้รับความนิยมในปัจจุบัน ซึ่งจากการศึกษางานวิจัยชิ้นดังกล่าวนี้ ยังไม่พบว่ามีองค์ประกอบส่วนใดที่ออกแบบมาเพื่อให้สามารถทำงานร่วมกับภาษาที่ใช้ในการพัฒนาโปรแกรมที่ได้รับความนิยมในปัจจุบันได้ ซึ่งหากงานวิจัยชิ้น

ดังกล่าวนี้มีองค์ประกอบดังกล่าว จะสามารถทำให้นักพัฒนาสามารถพัฒนาระบบค้นหาเชิงความหมายได้ง่ายดายมากยิ่งขึ้น เนื่องจากโปรแกรมภาษาอื่นๆ ล้วนแล้วแต่มีเฟรมเวิร์กที่ช่วยอำนวยความสะดวกในการพัฒนาโปรแกรมเป็นจำนวนมาก หากสามารถนำภาษาสคริปดังกล่าวมาร่วมใช้ด้วยได้ จะทำให้การพัฒนาเว็บเชิงความหมายสามารถทำได้ง่ายดายมากยิ่งขึ้นผ่านการใช้เฟรมเวิร์กอื่นๆ ของแต่ละโปรแกรมภาษาได้

2.2.3 การวิจัยเรื่องการใช้เจนาในการพัฒนาระบบวิเคราะห์พฤติกรรมความฉลาดในการเรียนรู้เชิงความหมายแบบออนไลน์ (The Research of Using Jena in the Semantic-based Online Learning Intelligent Behavior Analysis System) โดย Lei Shi, Lei Fan และ Zhenzhen Meng [17]

ในงานวิจัยชิ้นนี้ได้กล่าวถึงการนำเอาเจนาเฟรมเวิร์ก ซึ่งเป็นเครื่องมือในการพัฒนาระบบเว็บเชิงความหมาย มาใช้ในการพัฒนาระบบวิเคราะห์พฤติกรรมการเรียนรู้ของผู้เรียนในรูปแบบออนไลน์ โดยใช้เป็นเครื่องมือเพื่อช่วยในการวิเคราะห์เชิงความหมายของพฤติกรรมการเรียนรู้ ผู้วิจัยของการงานวิจัยชิ้นนี้ได้ใช้เจนาในการสร้างและปรับปรุงออนโทโลยี จากนั้นทำการสืบค้นข้อมูลพฤติกรรมการเรียนรู้ของผู้เรียนจากออนโทโลยีที่สร้างขึ้น

ในตอนท้ายของงานวิจัย ผู้วิจัยของงานวิจัยดังกล่าวได้พบว่าการวิเคราะห์ข้อมูลเชิงความหมายของพฤติกรรมการเรียนรู้ของผู้เรียนที่อาศัยความสามารถของเจนาเฟรมเวิร์กนั้นสามารถนำมาช่วยในการวิเคราะห์พฤติกรรมการเรียนรู้ของผู้เรียนได้ดีมากยิ่งขึ้น

จากการศึกษาพบว่า งานวิจัยชิ้นนี้เป็นอีกตัวอย่างหนึ่งของการนำเจนามาร่วมใช้ในการพัฒนาระบบเพื่อใช้ในการวิเคราะห์ข้อมูลเชิงความหมาย โดยอาศัยออนโทโลยีเพื่อเพิ่มประสิทธิภาพของการวิเคราะห์ข้อมูลให้ดีมากยิ่งขึ้น แต่อย่างไรก็ตาม ในงานวิจัยชิ้นดังกล่าวกลับไม่ได้อธิบายถึง วิธีการเพิ่ม หรือ แก้ไขข้อมูลของออนโทโลยีที่ถูกสร้างขึ้นเพื่อช่วยในการวิเคราะห์ต่างๆเหล่านั้นว่ามีกระบวนการอย่างไรหรือไม่ ซึ่งในความเป็นจริงแล้ว ออนโทโลยีซึ่งเป็นองค์ประกอบที่อธิบายองค์ความรู้ของมนุษย์ มีการเปลี่ยนแปลงตลอดเวลาตามองค์ความรู้ที่เปลี่ยนแปลงไป ซึ่งหากระบบดังกล่าวมี องค์ประกอบสำหรับปรับปรุงแก้ไขออนโทโลยีเพิ่มเติม จะทำให้ระบบสามารถวิเคราะห์ข้อมูลโดยอาศัยข้อมูลจากออนโทโลยีที่เป็นปัจจุบันได้ง่ายมากยิ่งขึ้น และส่งผลให้การวิเคราะห์ถูกต้องแม่นยำมากยิ่งขึ้น

2.2.4 ประโยชน์ในการบำรุงรักษาซอฟต์แวร์ด้วยการเขียนโค้ดแบบ แอนโนเตชันดริเวน (Software Maintainability Benefits From Annotation-Driven Code) โดย Leho Nigul และ Ernest Mah [18]

งานวิจัยชิ้นนี้ได้พูดถึงการใช้จาวาแอนโนเตชัน เพื่อช่วยในการดูแลรักษาโค้ดของซอฟต์แวร์ โดยในงานวิจัยชิ้นนี้ได้กล่าวว่า จาวาแอนโนเตชัน และเอ็กซ์ด็อกเล็ทเป็น องค์ประกอบที่สำคัญและมีประโยชน์เป็นอย่างมากในการพัฒนาโปรแกรมภาษาจาวา ทำให้ซอร์สโค้ดของระบบง่ายแต่การดูแลรักษามากยิ่งขึ้น โดยองค์ประกอบทั้งคู่ของภาษาจาวามีส่วนช่วยให้สามารถรวมเมตะดาตาต่างๆ ที่เกี่ยวข้องสามารถรวมเข้าไปในตัวซอร์สโค้ดได้โดยตรง

โดยในงานวิจัยชิ้นดังกล่าวนี้ได้พัฒนาเครื่องมือที่มีชื่อว่า เจทูอีอี คอนเน็คเตอร์ ทูล (J2EE Connector Tools) เพื่อใช้สำหรับสร้างซอร์สโค้ดที่มีการใช้งานแอนโนเตชันอยู่ภายใน โดยแอพพลิเคชันดังกล่าวจะทำการเชื่อมต่อกับระบบข้อมูลขององค์กรผ่าน ทาง เจทูอีอี คอนเน็ตเตอร์ เอพีไอ (J2EE Connector API) ด้วยการใช้แอนโนเตชัน

นอกจากนั้นแล้วในงานวิจัยชิ้นนี้ยังได้แสดงให้เห็นว่า การนำแอนโนเตชันมาร่วมใช้ในการพัฒนาซอฟต์แวร์ ทำให้ซอร์สโค้ดดูแลรักษาง่ายขึ้น ลดข้อผิดพลาดที่อาจจะเกิดจากการพัฒนาโปรแกรมได้ดีมากยิ่งขึ้นนั่นเอง

จากการศึกษางานวิจัยชิ้นนี้ ทำให้ทราบถึงข้อดีของการนำแอนโนเตชันเข้ามา ร่วมใช้งานในการพัฒนาซอฟต์แวร์ การนำแอนโนเตชันมาร่วมใช้ในการพัฒนาซอฟต์แวร์ทำให้ซอร์สโค้ดมีขนาดเล็กลง ดูแลรักษาง่ายขึ้นทำให้ข้อผิดพลาดที่อาจจะเกิดขึ้นมีน้อยตามลงไปด้วยนั่นเอง

# บทที่ 3
# การออกแบบวิธีการโอโอเอ็ม

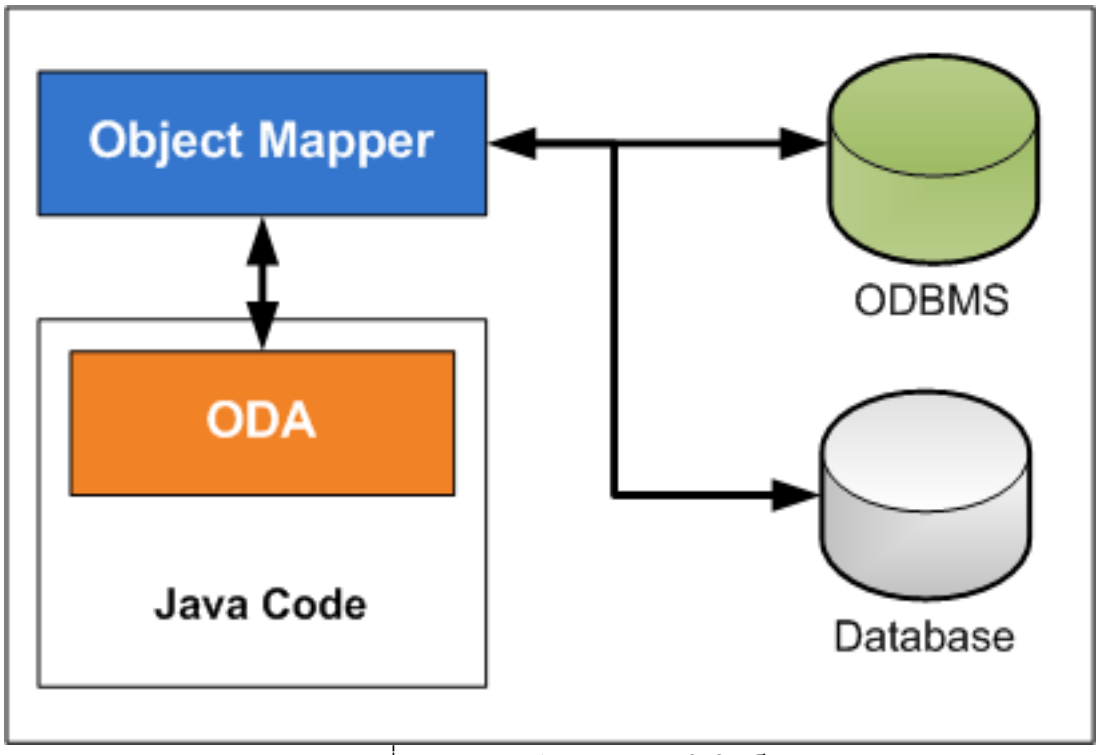
ในการนำออนโทโลยีมาใช้ร่วมในการพัฒนาระบบค้นหาเชิงความหมายนั้น ผู้วิจัยได้พัฒนาวิธีการในการนำออนโทโลยีมาร่วมใช้ในการพัฒนาระบบค้นหาเชิงความหมายด้วยวิธีการที่มีชื่อว่า อ็อบเจกต์ออนโทโลยีแมปปิง (Object-Ontology Mapping) หรือ โอโอเอ็ม (OOM) ซึ่งวิธีการดังกล่าวถูกสร้างขึ้นเพื่อแก้ปัญหาการพัฒนาระบบค้นหาเชิงความหมายที่มีการนำออนโทโลยีมาร่วมใช้การพัฒนา ที่ในปัจจุบันยังคงพบปัญหาที่สำคัญอยู่ 2 ประการ คือ

1. ความยุ่งยากซับซ้อนของเอพีไอสำหรับการนำเอาข้อมูลของออนโทโลยีไปใช้ร่วมในการพัฒนาระบบค้นหาเชิงความหมาย
2. วิธีการพัฒนาระบบค้นหาเชิงความหมายด้วยออนโทโลยีในปัจจุบัน ยังไม่มีองค์ประกอบสำหรับปรับปรุงแก้ไขข้อมูลของออนโทโลยีที่จะนำมาใช้ร่วมในการพัฒนาได้โดยตรง

จากปัญหาที่พบจากการนำออนโทโลยีมาร่วมในการพัฒนาระบบค้นหาเชิงความหมาย ทำให้ผู้วิจัยสามารถสรุปสิ่งที่นำมาใช้ในการแก้ปัญหาของการพัฒนาระบบค้นหาเชิงความหมายด้วยออนโทโลยีของวิธีการโอโอเอ็ม ได้ดังนี้

- ออกแบบให้มีการใช้จาวาแอนโนเตชันในการลดความซับซ้อนของซอร์สโค้ด
- ออกแบบให้มีองค์ประกอบสำหรับรองรับการเพิ่ม ปรับปรุง แก้ไขออนโทโลยีที่จะนำมาใช้ร่วมกับระบบค้นหา
- ออกแบบให้มีองค์ประกอบสำหรับทดสอบสปาร์เคิลที่จะนำไปใช้งานร่วมกับการพัฒนาระบบค้นหา
- ออกแบบให้วิธีการดังกล่าว สอดคล้องกับการพัฒนาโปรแกรมเชิงวัตถุตามแนวทางการพัฒนาซอฟต์แวร์กระแสหลักในปัจจุบัน

จากแนวทางการออกแบบของวิธีการโอโอเอ็มข้างต้น ทำให้ผู้วิจัยสามารถสรุปองค์ประกอบของวิธีการโอโอเอ็ม ให้มีองค์ประกอบที่สามารถตามที่ได้ออกแบบไว้ข้างต้นทั้งหมด 3 องค์ประกอบ ซึ่งสามารถแสดงองค์ประกอบต่างๆได้ดังภาพที่ 1




ภาพที่ 1 แสดงองค์ประกอบของโอโอเอ็ม

จากภาพที่ 1 สามารถสรุปเป็นภาพที่แสดงการทำงานร่วมกันขององค์ประกอบต่างๆของวิธีการโอโอเอ็ม โดยวิธีการโอโอเอ็มประกอบไปด้วยองค์ประกอบต่อไปนี้

### 3.1 **ออนโทโลยีดาต้าแอนโนเตชัน** (Ontology Data Annotation) **หรือโอดีเอ** (ODA)

จากข้อดีของแอนโนเตชันที่ถูกแบบมาเพื่อลดความซับซ้อนของซอร์สโค้ดลง ผู้วัจัยจึงได้นำข้อดีดังกล่าวนี้ มาร่วมใช้เพื่อร่วมแก้ไขปัญหาการนำออนโทโลยีมาร่วมใช้ในการพัฒนาระบบค้นหาเชิงความหมาย โดยมีจุดประสงค์เพื่อให้วิธีการโอโอเอ็มมีองค์ประกอบสำหรับเรียกใช้ข้อมูลของออนโทโลยีและจากฐานข้อมูลที่สามารถเรียกใช้งานได้ง่ายขึ้น โดยผู้วิจัยได้ออกแบบแอนโนเตชันขึ้นมาทั้งหมด 5 รูปแบบ คือ

3.1.1. เดฟฟินิชัน (Definition) เป็นแอนโนเตชันหลักสำหรับประกาศการใช้เรียกใช้ข้อมูลทั้งจากออนโทโลยีและจากระบบฐานข้อมูล โดยภายในเดฟฟินิชันแอนโนเตชันมีค่าที่ผู้ใช้งานต้องระบุค่าทั้งหมด 2 ค่า คือ

3.1.1.1. คิวรี (Query) เป็นค่าสำหรับระบุสปาร์เคิล (SPARQL) เพื่อทำหน้าที่สืบค้นข้อมูลจากออนโทโลยีที่จะนำมาใช้ร่วมในการพัมนาระบบค้นหา

3.1.1.2. <u>เอนติตี (Entity)</u> ใช้สำหรับระบุชื่อฐานข้อมูลที่ต้องการเรียกใช้งานจากระบบฐานข้อมูล

3.1.2. <u>แมปเปอร์ (Mapper)</u> เป็นแอนโนเตชันสำหรับระบุ เพื่อกรองเลือกเฉพาะแถวของข้อมูลที่ต้องการจากกลุ่มข้อมูลที่ระบุไว้ในเอนติตีแอนโนเตชัน

3.1.3. <u>วาริเอเบิล (Variable)</u> เป็นแอนโนเตชันที่ทำหน้าที่ดึงค่าจากตัวแปรของสปาร์เคิลตามที่ระบุไว้ในคิวรีแอนโนเตชันไปใส่ไว้ในตัวแปรของคลาสอ็อบเจกต์ ซึ่งภายในแอนโนเตชันนี้ผู้ใช้สามารถ ระบุค่าต่างๆเพิ่มเติมเพื่อกรองผลการค้นคืนที่ได้จากการสืบค้นออนโนโลยีเพื่อนำไปใช้งานต่อได้ค่าที่สามารถระบุเพิ่มเติมได้ประกอบไปด้วย

3.1.3.1. <u>ทาร์เก็ต (Target)</u> เป็นค่าหลักใช้สำหรับระบุตัวแปรของสปาร์เคิลที่ต้องการนำไปใช้งาน

3.1.3.2. <u>ฟิลเตอร์ (Filter)</u> เป็นค่าเพิ่มเติมใช้สำหรับประกาศเพื่อเลือกที่จะกรองตัวแปรจากผลการค้นคืนที่ได้จากออนโทโลยี

3.1.3.3. <u>อีควล (Equal)</u> เป็นค่าเพิ่มเติมใช้ร่วมกับค่าฟิลเตอร์ โดยระบุค่าให้สำหรับกรองเฉพาะตัวแปรที่ถูกระบุไว้ในค่าฟิลเตอร์และมีค่าเท่ากับที่ระบุไว้ในค่านี้เท่านั้น

3.1.4. <u>พร็อพเพอร์ตี (Properties)</u> ทำหน้านที่ดึงข้อมูลจากฟิลด์ หรือ พร็อพเพอร์ตีจากฐานข้อมูลตามที่ระบุไว้ในเอนติตีแอนโนเตชัน

3.1.5. <u>พารามิเตอร์ (Parameter)</u> ใช้สำหรับระบุค่าที่ต้องแทรกเข้าไปในสปาร์เคิลเพื่อนำค่าดังกล่าวไปช่วยในการสืบค้นข้อมูลจากออนโทโลยี

โดยในการเรียกใช้งานโอดีเอทั้งหมดนั้น ผู้ใช้สามารถเรียกใช้งานได้โดยตรงจากซอร์สโค้ดของโปรแกรมที่จะใช้พัฒนาระบบโดยตรงดังตัวอย่างการใช้งานแสดงได้ดังในภาพที่ 2

```
@Definition(query="" +
        " PREFIX oboInOwl: <http://www.geneontology.org/formats/oboInOwl#>"
        +" PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>"
        +" PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>"
        +" PREFIX owl:<http://www.w3.org/2002/07/owl#>"
        +" PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>"
        +" SELECT ?resource ?name ?resourcetype ?typelabel ?sourceref"
        +" WHERE {"
        +" ?source rdfs:label ?sourceref."
        +" ?source rdfs:subClassOf+ ?resource."
        +" ?resource rdfs:label ?name."
        +" ?resource rdfs:subClassOf ?blanknode."
        +" ?blanknode owl:someValuesFrom ?resourcetype."
        +" ?resourcetype rdfs:label ?typelabel."
        +" FILTER (contains(lcase(?sourceref), lcase(?parameter)))."
        +" FILTER (!(isBlank(?resource)))."
        +"}"
        ,entity="amphibian-th")
public class Species {

    @Mapper
    @Parameter
    private String parameter;

    @Variable(target="name",
            filter="typelabel",
            equal="class")
    private String bClass;

    @Variable(target="name",
            filter="typelabel",
            equal="order")
    private String order;

    @Variable(target="name",
            filter="typelabel",
            equal="family")
    private String family;

    @Variable(target="sourceref")
    private String species;

    @Property(name="localname")
    private String localname;

    @Property(name="local")
    private String locality;
```

ภาพที่ 2 แสดงวิธีการใช้งานโอดีเอ

จากภาพที่ 2 จะพบว่าเป็นตัวอย่างของการเรียกใช้งานโอดีเอจากคลาสที่มีชื่อว่าสปีชีส์ จากคลาสดังกล่าวจะพบว่าส่วนบนของคลาสมีการประกาศเรียกใช้เดฟฟินิชันแอนโนเตชัน โดยระบุสปาร์เคิลและชื่อฐานข้อมูลที่จะใช้งาน ภายในคลาสจะประกอบไปด้วยตัวแปรต่างๆ ในตัว

แปรแต่ละตัวได้มีการระบุแอนโนเตชันเพื่อเรียกใช้งาน เช่น ตัวแปรที่มีชื่อว่า parameter มีการประกาศใช้งานแมปเปอร์แอนโนเตชัน และพารามิเตอร์แอนโนเตชัน ซึ่งหมายความว่าค่าของตัวแปรนี้ทำหน้าที่ทั้งกรองแถวของข้อมูลที่ได้จากฐานข้อมูลและใช้เป็นส่วนหนึ่งของสปาร์เคิลเพื่อทำหน้าที่สืบค้นออนโนโลยี

ภายในคลาสยังมีการเรียกใช้งานวาริเอเบิลแอนโนเตชัน ดังที่ปรากฏในตัวแปร ที่ชื่อ bClass ซึ่งมีการระบุค่าภายในวาริเอเบิลแอนโนเตชัน โดยกำหนดให้ทาร์เก็ต มีค่าเท่ากับ name ฟิลเตอร์มีค่าเท่ากับ typelabel และ อีควลมีค่าเท่ากับ class ซึ่งจากการระบุค่าดังกล่าวเป็นการประกาศเพื่อทำการดึงข้อมูลที่ได้จากการสืบค้นจากออนโทโลยี โดยเลือกเฉพาะชุดของข้อมูลที่มีตัวแปรที่ชื่อว่า typelabel และมีค่าของตัวแปรนั้นๆเท่ากับ class แต่ดึงค่าจากตัวแปรที่มีชื่อว่า name ออกมาใส่เป็นค่าของตัวแปรตัวนี้

นอกจากนั้นแล้วภายในคลาสยังมีการเรียกใช้งานพร็อพเพอร์ตีแอนโนเตชันดังที่ปรากฏในตัวแปรที่มีชื่อว่า localname มีการเรียกใช้งานโดยระบุค่าเนมของ พร็อพเพอร์ตีแอนโนเตชันว่า localname ซึ่งมีหมายความว่าให้ดึงข้อมูลจากฐานข้อมูลที่ระบุไว้ในเดฟฟินิชันแอนโนเตชัน โดยเลือกเฉพาะคอลัมน์หรือพร็อพเพอร์ตีมีชื่อว่า localname นั่นเอง

## 3.2 **ออนโทโลยีดาต้าเบสเมเนจเมนท์ซิสเซ็ม** (Ontology Database Management System) **หรือโอดีบีเอ็มเอส** (ODBMS)

เพื่อพัฒนาให้การพัฒนาระบบค้นหาเชิงความหมาย ให้สามารถปรับปรุง แก้ไข หรือเพิ่มออนโทโลยีสำหรับนำมาใช้ร่วมกับการพัฒนาระบบค้นหาได้ง่ายดายมากยิ่งขึ้น ผู้วิจัยได้พัฒนาองค์ประกอบอีกส่วนเพื่อจัดการงานทางด้านนี้โดยเฉพาะ โดยใช้ชื่อว่าออนโทโลยีดาต้าเบสเมเนจเมนท์ซิสเซ็ม (Ontology Database Management System) หรือโอดีบีเอ็มเอส (ODBMS) ซึ่งเป็นระบบหลังบ้านสำหรับจัดการกับข้อมูลของออนโทโลยีทั้งหมดที่ใช้ในระบบค้นหา ซึ่งโอดีบีเอ็มเอสมีความสามารถที่ครอบคลุมต่อการแก้ไขจัดการและนำข้อมูลของออนโทโลยีไปใช้ใน 2 ด้าน คือ

3.1.1. <u>เพิ่ม , แก้ไขและลบข้อมูลของออนโทโลยี</u>

เพื่อให้โอดีบีเอ็มเอสสามารถจัดการกับข้อมูลที่เกี่ยวข้องกับออนโทโลยีที่จะนำมาใช้ร่วมกับการพัฒนาค้นหาเชิงความหมาย ผู้วิจัยจึงพัฒนาโอดีบีเอ็มเอสโดยใช้เจนาเฟรมเวิร์กในการพัฒนา ส่งผลให้โอดีเอ็มเอสสามารถประมวลผลข้อมูลของออนโทโลยีในรูปแบบไฟล์เอกสาร แบบ

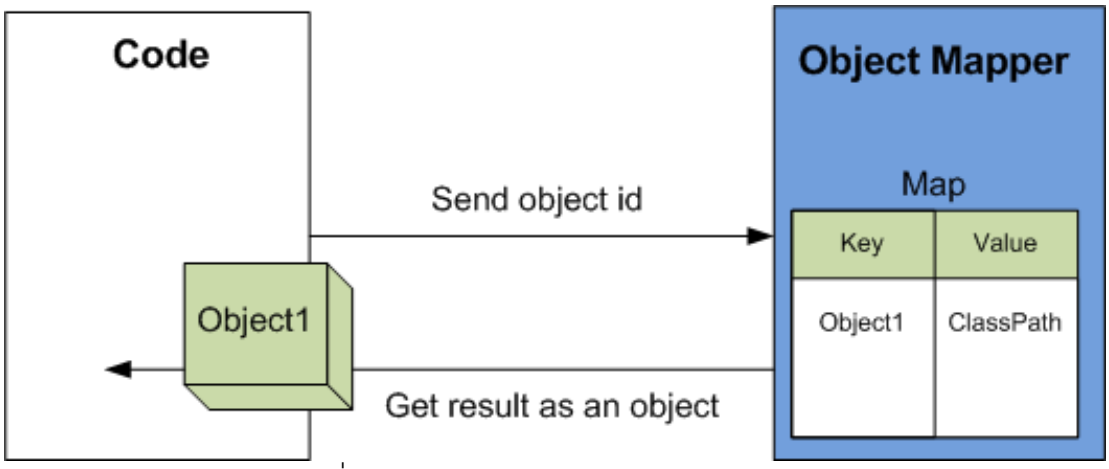
อาร์ดีเอฟหรือโอดับเบิลยูแอลได้โดยตรง ซึ่งรูปแบบของไฟล์ออนโทโลยีรูปแบบดังกล่าวถือเป็นไฟล์ในรูปแบบมาตราฐานสำหรับการเขียนออนโทโลยีในรูปแบบไฟล์ในปัจจุบัน ซึ่งทำให้ผู้ใช้สามารถเพิ่มข้อมูลของออนโทโลยีได้โดยการเขียนไฟล์ออนโทโลยีในรูปแบบมาตราฐานแล้วนำเข้าสู่โอดีบีเอ็มเอส ซึ่งโอดีบีเอ็มเอสที่ใช้ความสามารถของเจนาก็จะสามารถอ่านและเพิ่มข้อมูลของออนโทโลยีดังกล่าวเข้าไปในระบบได้ นอกจากนั้นยังสามารถจัดการแก้ไขหรือลบข้อมูลของออนโทโลยีโดยใช้ความสามารถเจนาได้อีกทางหนึ่งด้วย

3.1.2. <u>สืบค้นข้อมูลของออนโทโลยีด้วยสปาร์เคิล</u>

ในการพัฒนาระบบค้นหาโดยการนำออนโทโลยีมาร่วมใช้ สิ่งสำคัญอีกประการหนึ่งคือการทดสอบสปาร์เคิลที่จะนำไปใช้ร่วมกับการพัฒนาระบบค้นหา การทดสอบสปาร์เคิลที่นำมาใช้ร่วมกับการพัฒนาระบบจึงเป็นสิ่งสำคัญเพื่อให้ทราบได้ว่าผลของการค้นคืนของสปาร์เคิลมีลักษณะเป็นอย่างไร เพื่อให้การทดสอบการเรียกใช้ข้อมูลของออนโทโลยีสามารถทำได้สะดวกมากยิ่งขึ้น โอดีบีเอ็มเอสจึงมีฟังก์ชันสำหรับให้ผู้ใช้สามารถทดสอบสปาร์เคิลเพื่อทดสอบการสืบค้นข้อมูลจากออนโทโลยีที่ถูกสร้างขึ้นเพื่อใช้งานในในระบบได้ โดยวิจัยได้ใช้เจนาเฟรมเวิร์กช่วยในการพัฒนา เพื่อช่วยให้โอดีบีเอ็มเอสสามารถทดสอบการสืบค้นข้อมูลจากออนโทโลยีโดยใช้สปาร์เคิลด้วยเช่นกัน

## 3.3 **อ็อบเจกต์แมปเปอร์** (Object Mapper)

เพื่อให้ผู้พัฒนาระบบค้นหาสามารถเรียกใช้ข้อมูลจากออนโทโลยีและสามารถนำข้อมูลออกมาใช้ในรูปแบบอ็อบเจกต์ได้โดยตรงและสอดคล้องกับวิธีการพัฒนาโปรแกรมเชิงวัตถุ ผู้วิจัยจึงพัฒนาองค์ประกอบเพื่อทำหน้าที่สำหรับเรียกใช้งานข้อมูลที่ผู้ใช้ทำการสืบค้นจากออนโทโลยีและข้อมูลจากฐานข้อมูลได้ในรูปแบบของอ็อบเจกต์ โดยผู้วิจัยได้ออกแบบองค์ประกอบที่มีความสามารถนี้ชื่อว่าอ็อบเจกต์แมปเปอร์ โดยมีจุดประสงค์เพื่อให้ผู้พัฒนาสามารถได้อ็อบเจกต์ที่มีข้อมูลของออนโทโลยีสำหรับการพัฒนาโปรแกรมเชิงวัตถุได้ทันที โดยหลักการทำงานของอ็อบเจกต์แสดง ดังภาพที่ 3



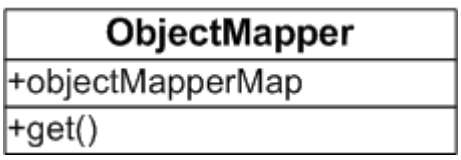

ภาพที่ 3 แสดงการทำงานของอ็อบเจกต์แมปเปอร์

จากภาพที่ 3 จะพบว่าหลักการทำงานของอ็อบเจกต์แมปเปอร์ คือ ภายในอ็อบเจกต์แมปเปอร์จะมีองค์ประกอบหลักที่สำคัญ คือ โครงสร้างข้อมูลแบบแมป (Map) ซึ่งโครงสร้างข้อมูลนี้จะทำหน้าที่เก็บคลาสพาธของจาวาอ็อบเจกต์ที่ผู้พัฒนาระบบค้นหาต้องการเรียกใช้และคีย์ของแต่ละคลาสพาธนั้น เมื่อผู้พัฒนาต้องการเรียกใช้งาน ก็ทำการส่งคีย์ของคลาสพาธที่ต้องการนั้นเข้าไปยังอ็อบเจกต์แมปเปอร์ อ็อบเจกต์แมปเปอร์ก็จะประมวลผลโดยทำการอ่านข้อมูลของคลาสที่ถูกกำหนดโดยค่าคีย์นั้นๆ จากนั้นอ็อบเจกต์แมปเปอร์จะทำการอ่านข้อมูลของคลาสนั้นๆซึ่งมีข้อมูลของโอดีเอที่ถูกระบุอยู่ภายใน จากนั้นอ็อบเจกต์แมปเปอร์จะทำการประมวลผลโอดีเอดังกล่าวแล้วดึงข้อมูลจากออนโทโลยีและฐานข้อมูลเข้าสู่อ็อบเจกต์และส่งอ็อบเจกต์คืนที่มีข้อมูลที่นักพัฒนาต้องการกลับไปในรูปแบบอ็อบเจกต์ ดังตัวอย่างที่แสดงตามภาพที่ 4

```
Animal animal = (Animal)objectMapper.get("animal");
```

ภาพที่ 4 แสดงการเรียกใช้งานคลาสอ็อบเจกต์แมปเปอร์

ซึ่งภายในคลาสอ็อบเจกต์แมปเปอร์ ผู้วิจัยได้ออกแบบให้คลาสมีรายละเอียดของคลาสดังที่รายละเอียดตามคลาสไดอะแกรมที่แสดงตามภาพที่ 5




ภาพที่ 5 แสดงคลาสไดอะแกรมของคลาสอ็อบเจกต์แมปเปอร์

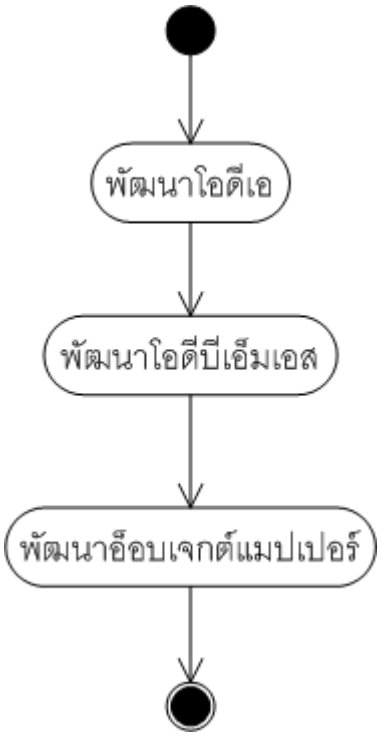
จากภาพที่ 5 พบว่าในคลาสออบเจ็กแมปเปอร์ ประกอบไปด้วยองค์ประกอบหลัก 2 ส่วน คือ

1. ตัวแปรที่เก็บข้อมูลเป็นโครงสร้างข้อมูลแบบแมพที่มีชื่อว่า objectMapperMap ซึ่งเป็นโครงสร้างข้อมูลที่ทำหน้าที่เก็บข้อมูลของคลาสพาธของคลาสแต่ละคลาส สำหรับเรียกใช้งานด้วยวิธีการโอโอเอ็ม เพื่อให้อ็อบเจกต์แมปเปอร์สามารถประมวลผลคลาสดังกล่าว และสามารถดึงข้อมูลจากออนโทโลยีเข้าสู่ออปเจกต์ตามข้อมูลที่ประกาศไว้ในโอดีเอ

2. เมธอดชื่อเก็ต (get) สำหรับให้นักพัฒนาสามารถเรียกใช้งานคลาสอ็อบเจกต์แมปเปอร์ และได้อ็อบเจกต์ที่มีข้อมูลของออนโทโลยีเพื่อนำไปใช้ในการพัฒนาระบบค้นหาต่อไป

# บทที่ 4
# การพัฒนาวิธีการโอโอเอ็ม

ในบทนี้ได้กล่าวถึงการการพัฒนาวิธีการโอโอเอ็ม ตามที่ได้มีการออกแบบให้วิธีการโอโอเอ็มมีความสามารถตามที่ได้ออกแบบไว้ในบทที่ 3 โดยมีขั้นตอนการในการพัฒนา 3 ขั้นตอน ตามองค์ประกอบของวิธีการโอโอเอ็ม ดังที่แสดงในภาพที่ 6 แสดงแผนภาพกิจกรรมการพัฒนาวิธีการโอโอเอ็ม




ภาพที่ 6 แผนภาพกิจกรรมการพัฒนาวิธีการโอโอเอ็ม

## 4.1. พัฒนาโอดีเอ

สำหรับในการพัฒนาโอดีเอนั้น ผู้วิจัยได้สร้างคลาสสำหรับสร้างจาวาแอนโนเตชันขึ้นมาทั้งหมด 5 คลาส โดยมีรายละเอียดของแอนโนเตชันตามที่ระบุไว้ในการออกแบบ ได้แก่ เดฟฟินิชัน , แมปเปอร์ , วาริเอเบิล , พร็อพเพอร์ตี และพารามิเตอร์ โดยมีวิธีการสร้างคลาสแต่ละคลาสตามวิธีการสร้างแอนโนเตชันคลาสในภาษาจาวา

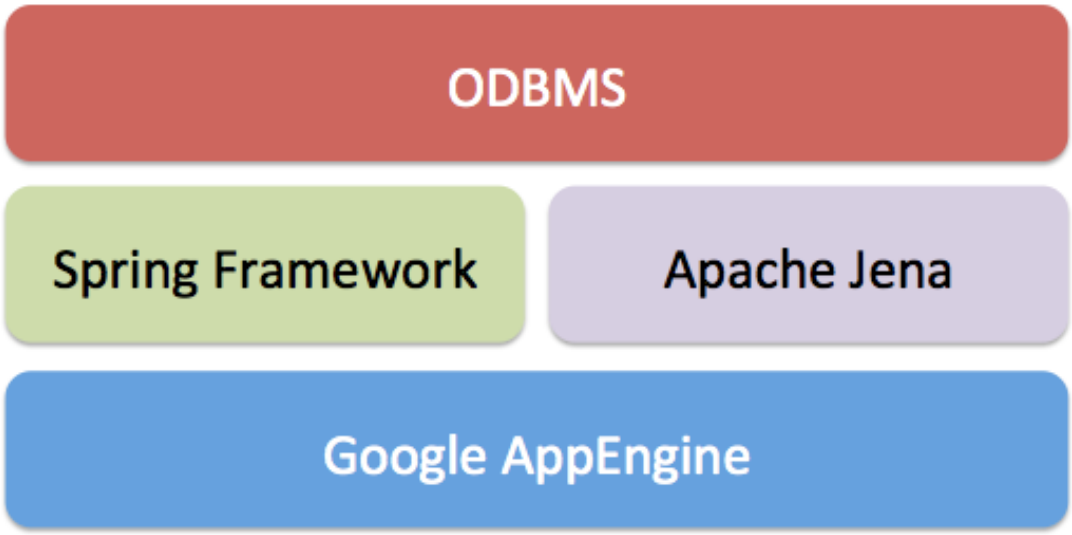
```
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.TYPE)

public @interface Definition {
    String query();
    String entity() default "";
}
```

ภาพ 7 แสดงตัวอย่างการสร้างคลาสแอนโนเตชัน

จากในภาพที่ 7 เป็นภาพตัวอย่างสำหรับการสร้างคลาสแอนโนเตชันของเดฟฟินิชันแอนโนเตชัน โดยในการออกแบบได้มีการกำหนดให้มีค่าที่สามารถระบุได้ 2 ค่าคือคิวรี และ แอนติตี ดังนั้นภายในเดฟฟินิชันจึงมีกำหนดค่าที่สามารถระบุค่าได้ 2 ค่า คือ คิวรี และ แอนติตี ตามวิธีการสร้างของรูปแบบแอนโนเตชันคลาสในภาษาจาวา

## 4.2. พัฒนาโอดีบีเอ็มเอส

สำหรับองค์ประกอบโอดีบีเอ็มเอสนั้น เป็นองค์ประกอบหลักสำหรับการจัดการกับข้อมูลของออนโทโลยีที่จะนำมาใช้ร่วมในการพัฒนาระบบค้นหาเชิงความหมาย เพื่อเพิ่มประสิทธิภาพการนำข้อมูลของออนโทโลยีมาใช้งานในระบบค้นหาให้มีประสิทธิภาพเพิ่มสูงขึ้น ผู้วิจัยจึงเลือกที่จะเก็บข้อมูลของออนโทโลยีที่ถูกประมวลผลแล้วทั้งหมดเก็บไว้ในหน่วยความจำของระบบเพื่อเพิ่มความรวดเร็วในการเข้าถึงข้อมูลของออนโทโลยีในระบบค้นหา โดยผู้วิจัยได้พัฒนาองค์ประกอบโอดีบีเอ็มเอสในรูปแบบเว็บแอพพลิเคชันด้วยภาษาจาวา เพื่อให้สะดวกต่อการจัดการเพิ่ม , แก้ไข หรือปรับปรุงของออนโทโลยีที่จะนำมาใช้พัฒนาระบบค้นหาได้ง่ายดายมากขึ้น ซึ่งรายละเอียดของเทคโนโลยีที่ใช้ในการพัฒนาแสดงรายละเอียดตามภาพที่ 8

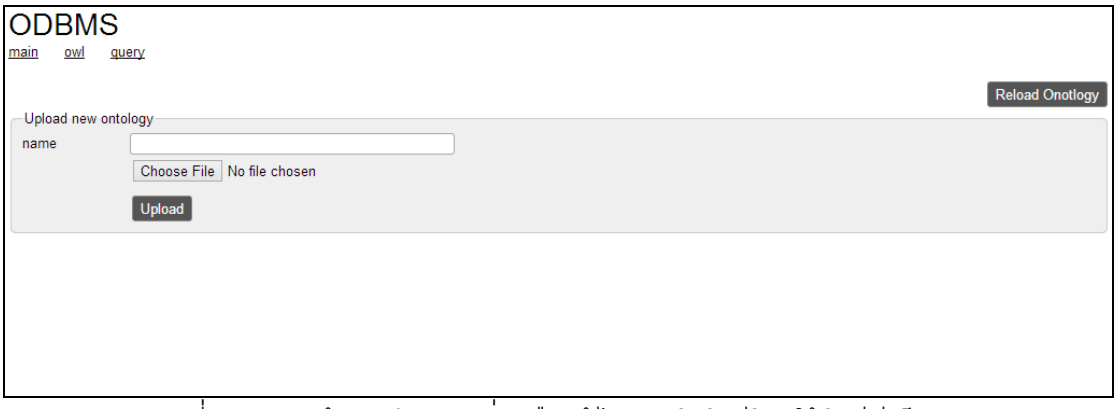



ภาพที่ 8 แสดงเทคโนโลยีที่ใช้ในการพัฒนาโอดีบีเอ็มเอส

โดยในการวิจัยชิ้นนี้ ผู้วิจัยได้พัฒนาองค์ประกอบโอดีบีเอ็มเอสภายใต้สภาพแวดล้อมของกูเกิลแอพเอนจิน โดยพัฒนาในรูปแบบเว็บแอพพลิเคชันภายในระบบมีการใช้สปริงเฟรมเวิร์กในการสร้างเว็บแอพพลิเคชัน และใช้เจนาเฟรมเวิร์กสำหรับการจัดการกับข้อมูลของออนโทโลยี เป้าหมายหลักในการสร้างองค์ประกอบโอดีบีเอ็มเอส จะเป็นการสร้างเพื่อพัฒนาองค์ประกอบให้มีความสามารถ 2 ประการตามที่ได้มีการออกแบบไว้คือ

4.2.1 <u>เพิ่ม , แก้ไขและลบข้อมูลของออนโทโลยีที่จะนำมาใช้ร่วมกับการค้นหา</u>

สำหรับฟังก์ชันดังกล่าวนี้ ผู้วิจัยได้พัฒนาองค์ประกอบโอดีบีเอ็มเอสให้มีลักษณะดังที่ปรากฏตามภาพที่ 9



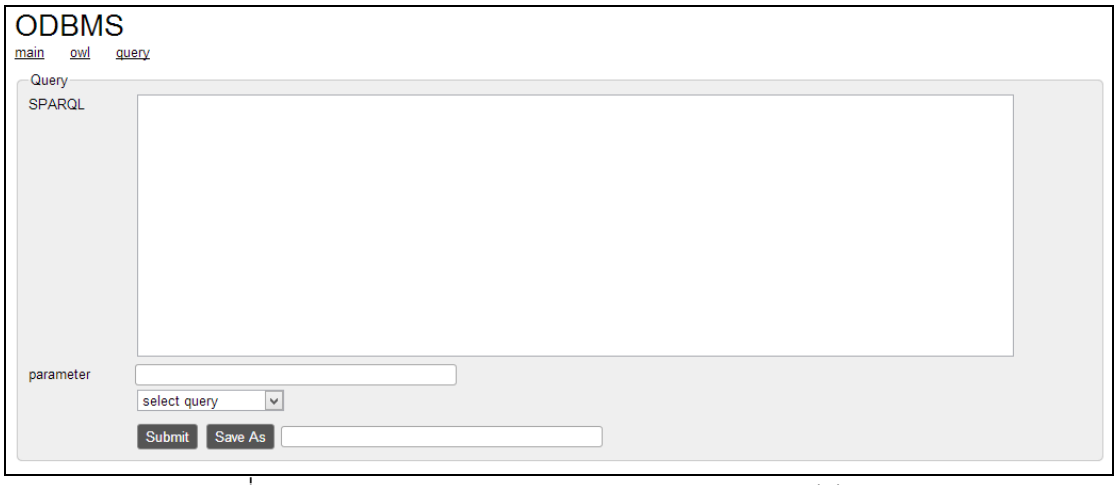

ภาพที่ 9 แสดงหน้าจอสำหรับเพิ่มหรือแก้ไขออนโทโลยีโดยใช้โอดีบีเอ็มเอส

จากภาพที่ 9 เป็นภาพที่แสดงหน้าจอ สำหรับการเพิ่มหรือแก้ไข ข้อมูลของออนโทโลยีโดยใช้โอดีบีเอ็มเอส เมื่อผู้ใช้ต้องการที่จะเพิ่มออนโทโลยีใหม่ เข้าไปในระบบ ผู้ใช้งานสามารถทำได้โดยการคลิกที่ปุ่มเลือกไฟล์ และระบุชื่อของออนโทโลยีที่ต้องการเพิ่มเข้าไปในระบบนั้น เมื่อผู้ใช้คลิกปุ่มอัพโหลด (Upload) โอดีบีเอ็มเอสจะบันทึกข้อมูลไฟล์ของออนโทโลยีนั้นๆ เข้าไปในฐานข้อมูลและข้อมูลที่อยู่ในไฟล์ดังกล่าวจะถูกนำไปประมวลผลและสร้างเป็นออนโทโลยีสำหรับนำไปใช้ร่วมกับระบบค้นหาต่อไป โดยผู้วิจัยในความสามารถดังกล่าวนี้ผู้วิจัยได้ใช้ความสามารถของเจนาเฟรมเวิร์กในการจัดการ

แต่ในกรณีที่ผู้ใช้งานต้องการลบหรือแก้ไขข้อมูลของออนโทโลยีที่บันทึกอยู่ใน โอดีบีเอ็มเอสแล้วผู้ใช้สามารถทำได้โดยคลิกที่ปุ่มแก้ไข (Edit) เพื่อทำการแก้ไขรายละเอียดของไฟล์ออนโทโลยีนั้นๆ หรือกดปุ่มลบ (Delete) เพื่อลบข้อมูลของไฟล์ออนโทโลยีนั้นๆ ออกไปจากระบบ

4.2.2 สืบค้นข้อมูลของออนโทโลยีด้วยสปาร์เคิล

ผู้วิจัยได้พัฒนาโอดีบีเอ็มเอสให้สามารถทดสอบสปาร์เคิลเพื่อนำไปใช้ร่วมกับการพัฒนาระบบค้นหาเชิงความหมายได้ โดยหน้าจอสำหรับการทดสอบสปาร์เคิลที่ต้องการนำไปใช้งาน แสดงดังที่ปรากฏในภาพที่ 10




ภาพที่ 10 แสดงหน้าจอสำหรับทดสอบสปาร์เคิลจากโอดีบีเอ็มเอส

จากภาพที่ 10 เมื่อผู้พัฒนาระบบค้นหา ระบุสปาร์เคิลที่ต้องการในช่องสปาร์เคิล (SPARQL) จากนั้นกดปุ่มซับมิท (Submit) โอดีบีเอ็มเอสจะแสดงผลการค้นคืนที่สืบค้นจากออนโทโลยีให้กับนักพัฒนา เพื่อให้ผู้ใช้ตรวจสอบผลการค้นคืนที่ได้และพิจารณานำสปาร์เคิลและผล

การค้นคืนที่ได้ไปใช้ในการพัฒนาระบบค้นหาเชิงความหมายต่อไป โดยในการประมวลผลสปาร์เคิลในการสืบค้นข้อมูลของออนโทโลยีนั้น โอดีบีเอ็มเอสใช้ความสามารถของเจนาเฟรมเวิร์กในการสืบค้น รวมถึงหลังจากทำการประมวลผลเสร็จเรียบร้อย โอดีบีเอ็มเอสจะก็จะนำผลการค้นคืนที่ได้จากการใช้เจนาเฟรมเวิร์กมาแสดงผลให้กับผู้ใช้ต่อไป

### 4.3. พัฒนอ็อปเจกต์แมปเปอร์

ในการพัฒนาองค์ประกอบอ็อปเจกต์แมปเปอร์นั้น ผู้วิจัยได้สร้างจาวาคลาสที่มีชื่อว่าอ็อบเจกต์แมปเปอร์ ที่มีรายละเอียดตามที่ระบุไว้ในคลาสไดอะแกรมดังที่ปรากฏตามภาพที่ 5 ซึ่งภายในคลาสดังกล่าวนี้ ได้มีการใช้เอพีไอของเจนาเพื่อช่วยในการประมวลผลข้อมูลของออนโทโลยี ซึ่งในการพัฒนาอ็อบเจกต์แมปเปอร์นั้นจำเป็นต้ององค์ประกอบที่สำคัญ 2 ส่วนคือ

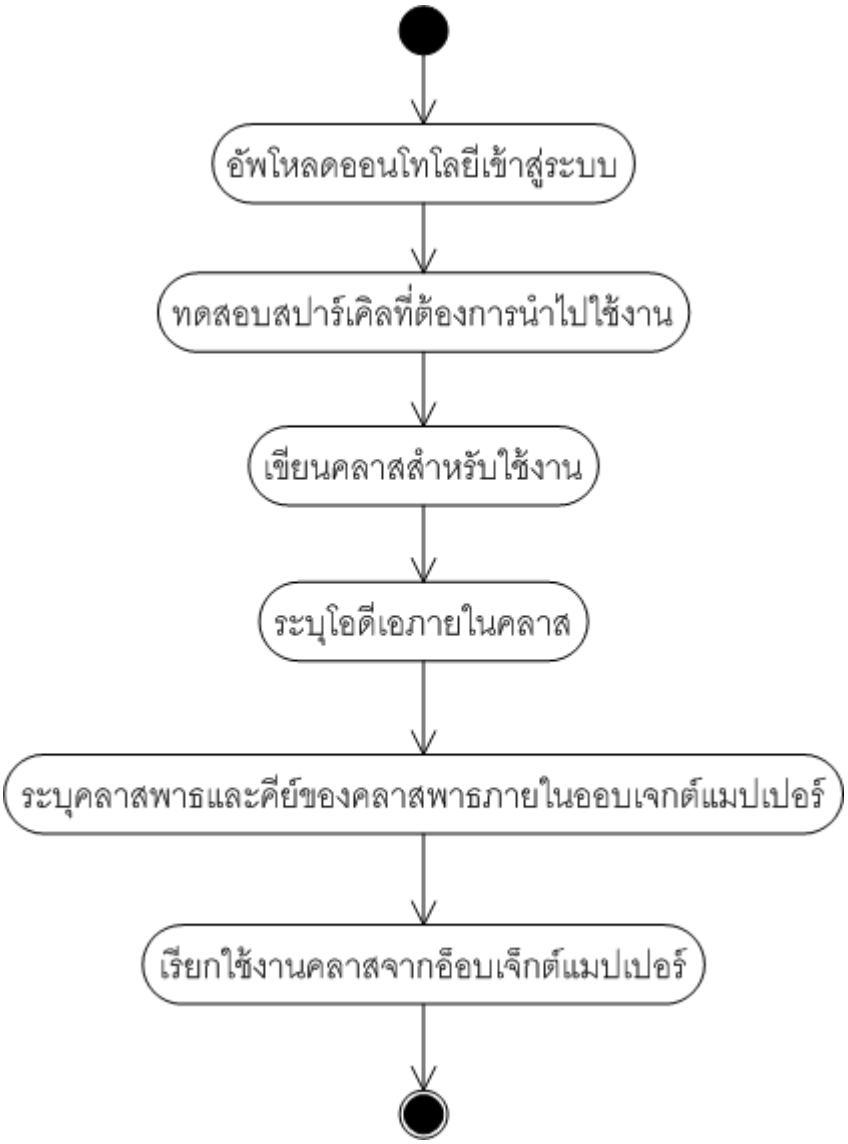
1. ความสามารถรีเฟลกชันของภาษาจาวา สำหรับอ่านข้อมูลของโอดีเอที่ถูกเขียนไว้ในคลาสที่ต้องการเรียกใช้งาน
2. เจนาเฟรมเวิร์กสำหรับสืบค้นข้อมูลของออนโทโลยีในระบบและดึงข้อมูลจากผลการค้นคืนเข้าสู่อ็อบเจกต์

ภายในคลาสอ็อปเจกต์แมปเปอร์จะประกอบไปด้วยขั้นตอนการทำงาน เพื่อให้สามารถส่งผลคืนเป็นอ็อบเจกต์ที่มีข้อมูลของออนโทโลยีที่ผู้ใช้ต้องการดังต่อไปนี้

1. ใช้ความสามารถของรีแฟลกชันของภาษาจาวา อ่านข้อมูลของแอนโนเตชันที่เขียนประกาศไว้ในคลาส
2. อ่านสปาร์เคิลที่ถูกระบุไว้ในเดฟฟินิชันแอนโนเตชัน ไปสืบค้นข้อมูลจากออนโทโลยีที่สร้างขึ้นโดยองค์ประกอบโอดีบีเอ็มเอสโดยใช้เจนาเฟรมเวิร์ก
3. อ็อบเจกต์แมปเปอร์ รับผลการค้นคืนจากการสืบค้นออนโทโลยี และใส่ผลค้นคืนเข้าสู่ตัวแปรของอ็อบเจกต์ ตามค่าที่ระบุไว้ในโอดีเอของตัวแปรแต่ละตัวแปรของอ็อบเจกต์นั้นๆ

# บทที่ 5
# การใช้งานวิธีการโอโอเอ็ม

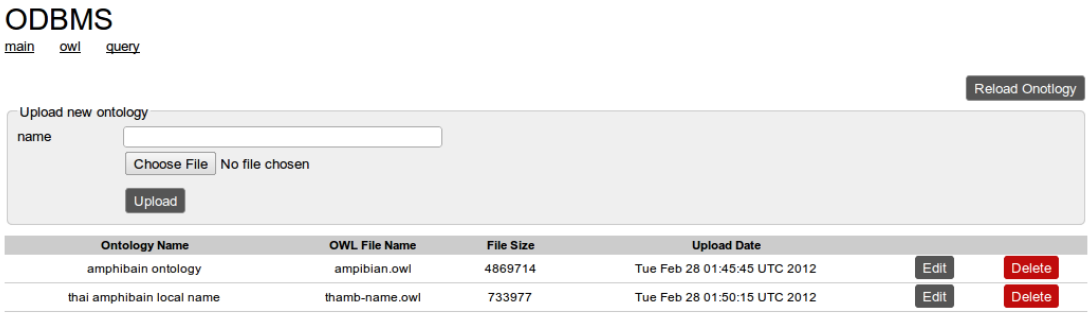
หลังจากที่ผู้วิจัยได้พัฒนาองค์ประกอบต่างๆของวิธีการเรียบร้อยแล้ว ในบทนี้จะเป็นการอธิบายขั้นตอนการนำวิธีการโอโอเอ็มไปใช้งาน โดยในการใช้งานวิธีการโอโอเอ็มประกอบไปด้วยขั้นตอนต่างๆ ดังที่แสดงตามภาพที่ 11 แสดงแผนภาพกิจกรรมการใช้งานวิธีการโอโอเอ็ม




ภาพที่ 11 แสดงแผนภาพกิจกรรมการใช้งานวิธีการโอโอเอ็ม

## 5.1. อัพโหลดออนโทโลยีเข้าสู่ระบบ

ขั้นตอนแรกของการใช้งานวิธีการโอโอเอ็มนั้น ผู้ใช้งานจะต้องทำการสร้างออนโทโลยีที่ต้องการนำไปใช้ในการพัฒนาระบบค้นหาเชิงความหมายขึ้นมาเป็นอันดับแรก โดยผู้ใช้สามารถสร้างออนโทโลยีของระบบขึ้นมาได้ โดยการอัพโหลดไฟล์ออนโทโลยีที่อยู่ในรูปแบบอาร์ดีเอฟหรือโอดับเบิลยูแอลเข้าสู่ระบบผ่านองค์ประกอบโอดีบีเอ็มเอส หลังจากที่ผู้ใช้อัพโหลดไฟล์ออนโทโลยีเข้าสู่ระบบผ่านโอดีบีเอ็มเอสแล้ว โอดีบีเอ็มเอสจะบันทึกไฟล์ของออนโทโลยีเหล่านั้นไว้ และนำข้อมูลไปสร้างเป็นออนโทโลยีของระบบเพื่อใช้ในระบบค้นหาต่อไป

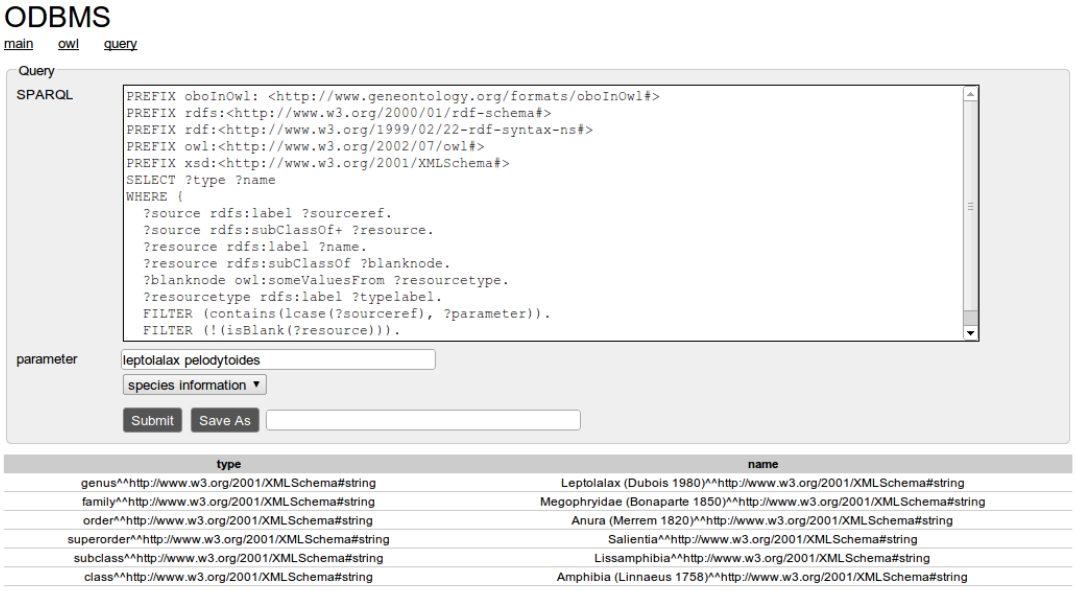



ภาพที่ 12 แสดงการอัพโหลดไฟล์ออนโทโลยีเข้าสู่ระบบ

จากภาพที่ 12 แสดงการอัพโหลดไฟล์ออนโทโลยีเข้าสู่ระบบผ่านองค์ประกอบโอดีบีเอ็มเอส โดยมีการอัพโหลดไฟล์ออนโทโลยีเข้าในระบบทั้งสิ้น 2 ไฟล์ โดยไฟล์ออนโทโลยีทั้งสองไฟล์เมื่ออัพโหลดเข้าไปแล้ว โอดีบีเอ็มเอสจะนำไปสร้างออนโทโลยีสำหรับใช้ระบบค้นหาภายในหน่วยความจำและแสดงรายละเอียดของไฟล์ของออนโทโลยีที่ถูกบันทึกไว้

## 5.2. สร้างและทดสอบสปาร์เคิลที่ต้องการนำไปใช้งาน

สำหรับขั้นตอนต่อมานั้น หลังจากที่ผู้ใช้สร้างออนโทโลยีขึ้นมาในระบบแล้ว ผู้ใช้จะต้องทำการสร้างสปาร์เคิลสำหรับสืบค้นข้อมูลที่ต้องการจากออนโทโลยีที่ถูกสร้างขึ้นในระบบนั้น และทดสอบผลการค้นคืนจากสปาร์เคิลที่ได้สร้างขึ้น เพื่อให้ได้ผลการค้นคืนจากออนโทโลยีตามที่ต้องการและนำผลการค้นคืนที่ได้ไปใช้ในการดึงข้อมูลเข้าสู่อ็อบเจกต์เพื่อนำไปใช้ในการพัฒนาระบบค้นหาต่อไป




ภาพที่ 13 ทดสอบการสปาร์เคิลเพื่อนำไปใช้ในการพัฒนาระบบ

จากภาพ 13 เป็นการทดสอบสปาร์เคิลเพื่อสืบค้นข้อมูลจากออนโทโลยีที่สร้างขึ้นในระบบ เพื่อให้ได้สปาร์เคิลสำหรับนำไปร่วมใช้ในการพัฒนาระบบค้นตามที่ต้องการ หลังจากที่ผู้กดปุ่มซับมิท โอดีบีเอ็มเอสจะแสดงผลการค้นคืนจากสปาร์เคิลที่ผู้ใช้ระบุไป ทำให้นักพัฒนาสามารถตรวจสอบความถูกต้องของสปาร์เคิลที่ระบุได้และสามารถนำข้อมูลของผลการค้นคืนที่ได้ไปใช้ร่วมในการพัฒนาระบบค้นหาต่อไป

## 5.3. **สร้างคลาสสำหรับใช้งาน**

เมื่อนักพัฒนาทราบแล้วว่าต้องการใช้ค่าอะไรบ้างจากผลการค้นคืนที่ได้จากออนโทโลยีมาร่วมใช้ในการพัฒนาระบบค้นหา นักพัฒนาจะต้องสร้างคลาสของภาษาจาวาเพื่อทำการดึงข้อมูลจากผลการค้นคืนดังกล่าวเข้าสู่คลาสอ็อบเจกต์นั้นๆ เพื่อนำไปใช้ต่อในการพัฒนาระบบค้นหา

```
public class Animal {

    private String parameter;

    private String species;

    private String localname;

}
```

ภาพที่ 14 แสดงการสร้างคลาสเพื่อใช้ในระบบค้นหา

จากภาพที่ 14 เป็นตัวอย่างการสร้างจาวาคลาสที่ชื่อว่า Animal เพื่อนำไปร่วมใช้ในการพัฒนาระบบค้นหา ภายในมีตัวแปรทั้งสิ้น 3 ตัวคือ parameter , species และ localname

### 5.4. ระบุโอดีเอภายในคลาส

หลังจากที่นักพัฒนาสร้างคลาสบนจาวาเพื่อเป็นเก็บข้อมูลของภาษาจาวาเรียบร้อยแล้วนั้น นักพัฒนาที่จะต้องระบุโอดีเอสำหรับดึงข้อมูลจากออนโทโลยีมาใส่ไว้ในอ็อบเจกต์ในลักษณะที่แสดงในภาพที่ 15

```
@Definition(query = "PREFIX biodb:<http://www.biologydb/element#>"
      + " PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>"
      + " PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>"
      + " PREFIX owl:<http://www.w3.org/2002/07/owl#>" + " SELECT ?species "
      + " WHERE {" + " ?element biodb:name ?parameter."
      + " ?element biodb:name ?species." + "}", entity = "amphibian")
public class Animal {

   @Mapper
   @Parameter("parameter")
   private String parameter;

   @Variable(target = "species")
   private String species;

   @Property(name = "localname")
   private String localname;
```

ภาพที่ 15 แสดงการระบุโอดีเอภายในคลาสอ็อบเจกต์ของภาษาจาวา

จากภาพที่ 15 เป็นการประกาศเรียกใช้โอดีเอภายในคลาสที่มี Animal ที่สร้างขึ้นจากขั้นตอนที่ 5.3 ภายในคลาส Animal มีการระบุสปาร์เคิลที่ต้องการใช้งานไว้ในเดฟฟินิชันแอนโนเตชัน มีการเรียกใช้ฐานข้อมุลที่มีค่า amphibian จากการเรียกใช้งานแอนติตีแอนโนเตชัน และมีการ

ดึงค่าต่างๆที่ได้จากผลการค้นคืนมาใส้ไว้ในตัวแปรของอ็อปเจกต์ โดยการเรียกใช้วาริเอเบิลแอนโนเตชัน และมีการดึงข้อมูลจากฟิลด์ในฐานข้อมูลที่มีชื่อว่า localname ผ่านการประกาศใช้งานพร็อพเพอร์ตีแอนโนเตชัน

### 5.5.ระบุคลาสพาธและคีย์ของคลาสพาธภายในอ็อบเจกต์แมปเปอร์

หลังจากที่ผู้ใช้ทำการระบุโอดีเอภายในคลาสที่ต้องการเรียกใช้งานแล้ว ผู้ใช้งานจะต้องทำการระบุคลาสพาธของคลาสนั้นๆ และคีย์ของคลาสพาธนั้นๆ ลงในตัวแปรที่มีชื่อ objectMapperMap ขององค์ประกอบอ็อบเจกต์แมปเปอร์ เพื่อให้สามารถเรียกใช้งานคลาสดังกล่าวที่มีการดึงข้อมูลจากออนโทโลยีเรียบร้อบแล้ว ผ่านทางการใช้งานองค์ประกอบอ็อบเจกต์แมเปอร์ได้

จากขั้นตอนที่ 5.3 และ 5.4 ซึ่งเป็นตัวอย่างของการสร้างอ็อบเจกต์คลาสสำหรับใช้งานจากออนโทโลยี ดังนั้นในขั้นตอนที่ 5.5 ซึ่งเป็นการระบุคีย์และคลาสพาธของคลาสอ็อบเจกต์ที่ต้องการเรียกใช้งาน ให้สามารถใช้งานผ่านองค์ประกอบอ็อบเจกต์แมปเปอร์ได้ นักพัฒนาก็จะต้องเพิ่มข้อมูลของคลาส Animal ลงในตัวแปรที่มีชื่อว่า objectMapperMap ของอ็อบเจกต์แมปเปอร์ โดยสามารถใช้ค่าคีย์มีค่าว่า animal และระบุค่าของคลาสพาธเป็นคลาสพาธของคลาส Animal นั้น

### 5.6.เรียกใช้งานอ็อบเจกต์ผ่านองค์ประกอบอ็อบเจกต์แมปเปอร์

หลังจากที่ผู้ใช้ระบุคลาสที่ต้องการเรียกใช้งานภายในตัวแปรที่มีชื่อว่า objectMapperMap ขององค์ประกอบอ็อบเจกต์แมปเปอร์เรียบร้อยแล้ว ผู้ใช้งานสามารถเรียกใช้งานอ็อบเจกต์ที่มีข้อมูลของออนโทโลยีตามสปาร์เคิลที่ระบุดังด้วยการใช้งานดังภาพที่ 16

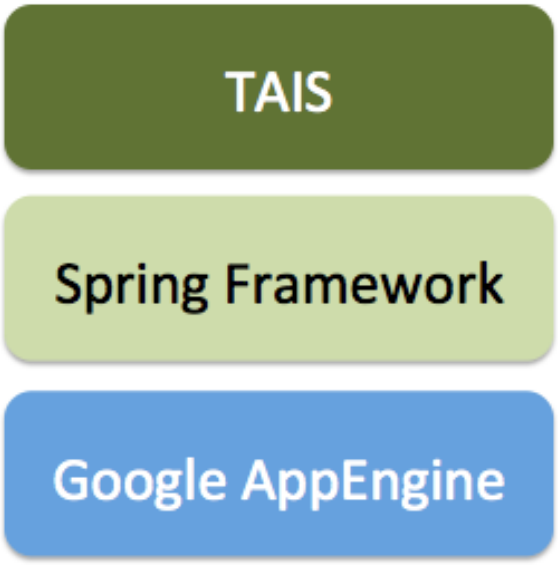
```
Animal animal = (Animal) objectMapper.get("animal");
```

ภาพที่ 16 แสดงการเรียกใช้งานคลาส Animalผ่านองค์ประกอบอ็อบเจกต์แมปเปอร์

จากภาพที่ 16 เป็นการเรียกใช้งานคลาสอ็อบเจกต์ที่มีชื่อว่า Animal ผ่านองค์ประกอบอ็อบเจกต์แมปเปอร์ โดยทำการส่งคีย์ที่มีค่าว่า “animal” ที่กำหนดไว้ในขั้นตอนที่ 5.5 เข้าไปในองค์ประกอบอ็อบเจกต์แมปเปอร์ จากนั้นอ็อบเจกต์แมปเปอร์จะคืนอ็อบเจกต์คลาสกลับคืนมา ผู้ใช้ก็จะต้องทำการแปลงคลาสจากคลาสอ็อบเจกต์มาเป็นคลาส Animal และนำอ็อบเจกต์ที่ถูกสร้างขึ้นมานั้นไปใช้ในการพัฒนาระบบค้นหาต่อไป

# บทที่ 6
# การนำวิธีการโอโอเอ็มไปพัฒนาระบบค้นหา

เพื่อทดสอบการนำวิธีการโอโอเอ็มไปใช้งาน ผู้วิจัยได้นำวิธีการโอโอเอ็มมาสร้างระบบค้นหาเชิงความหมาย โดยเป็นระบบค้นหาข้อมูลของสิ่งมีชีวิตประเภทสัตว์สะเทินน้ำสะเทินบกของประเทศไทย (Thai Amphibian Information System) หรือไทส์ (TAIS) โดยตัวระบบมีรูปแบบของสถาปัตยกรรมดังที่แสดงตามภาพที่ 17



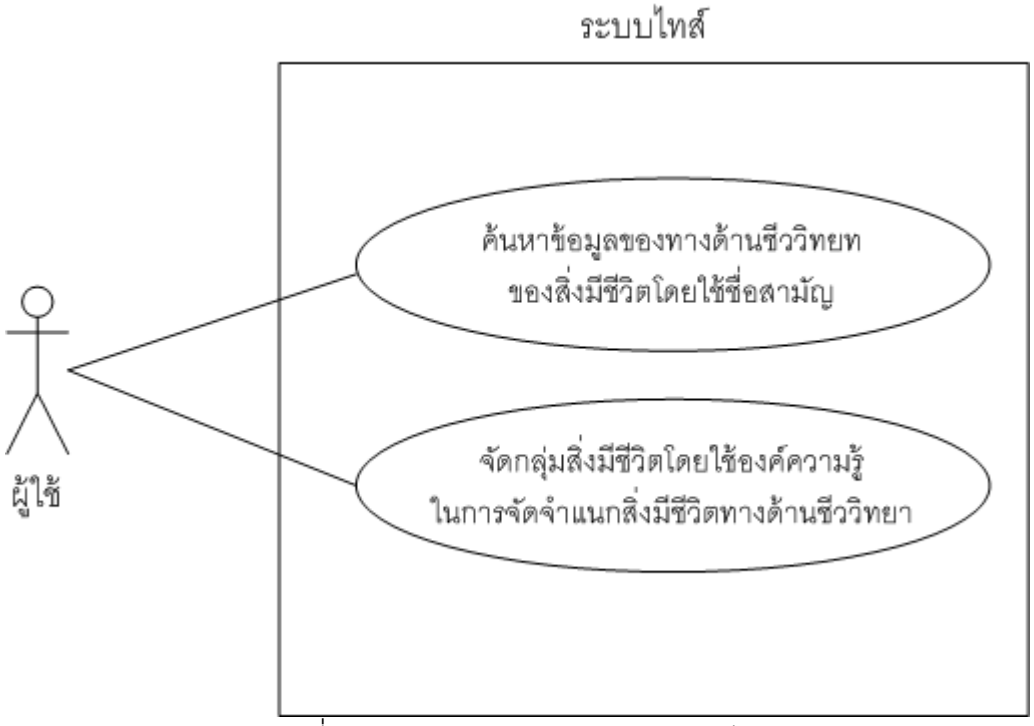

ภาพที่ 17 แสดงเทคโนโลยีที่ใช้ในการพัฒนาระบบไทส์

ในการพัฒนาระบบไทส์นั้น ผู้วิจัยเลือกพัฒนาในรูปแบบเว็บแอพพลิเคชันโดยใช้ภาษาจาวาในการพัฒนา ผู้วิจัยเลือกพัฒนาแอพพลิเคชันโดยใช้เฟรมเวิร์กช่วยในการพัฒนา โดยผู้วิจัยได้เลือกเฟรมเวิร์กที่มีชื่อว่าสปริงเฟรมเวิร์กซึ่งเป็นเฟรมเวิร์ก สำหรับพัฒนาเว็บแอพพลิเคชันด้วยภาษาจาวาที่ได้รับความนิยมอย่างสูงในปัจจุบัน โดยการพัฒนาระบบค้นหาทั้งหมดพัฒนาอยู่บนสภาวะแวดล้อมของกูเกิลแอพเอนจินซึ่งเป็นสภาวะแวดล้อมแบบคลาวด์คอมพิวติงที่ได้รับความนิยมอย่างมากในปัจจุบัน

ในการพัฒนาระบบค้นหาเชิงความหมายเพื่อทดสอบการใช้งานวิธีการโอโอเอ็มนี้ ผู้วิจัยออกแบบระบบไทส์ให้มีความสามารถดังต่อไปนี้

- ค้นหาข้อมูลของสิ่งมีชีวิตทางชีววิทยาโดยใช้ชื่อสามัญของสิ่งมีชีวิตได้
- จัดกลุ่มความสัมพันธ์ของสิ่งมีชีวิตตามการจัดจำแนกสิ่งมีชีวิตทางด้านชีววิทยาได้

ซึ่งสามารถดังกล่าว สามารถสรุปเป็นแผนภาพยูสเคสดังที่แสดงในภาพที่ 18




ภาพที่ 18 แสดงแผนภาพยูสเคสของระบบไทส์

เมื่อผู้วิจัยสรุปความต้องการทั้งหมดของระบบไทส์ได้เรียบร้อยแล้ว ผู้วิจัยจึงนำความต้องการทั้งหมดไปพัฒนาระบบไทส์ด้วยวิธีการโอโอเอ็มในลำดับต่อไป โดยในการพัฒนาระบบค้นหาข้อมูลสิ่งมีชีวิตประเภทสัตว์สะเทินน้ำสะเทินบนบกนี้ ผู้วิจัยได้ทำการพัฒนาระบบ โดยทำตามวิธีการโอโอเอ็มตามขั้นตอนต่อไปนี้

### 6.1. **อัพโหลดออนโทโลยีเข้าสู่ระบบ**

สิ่งที่สำคัญในการพัฒนาระบบค้นหาเชิงความหมายด้วยวิธีการโอโอเอ็ม คือ การนำออนโทโลยีมาร่วมใช้ในการพัฒนาระบบค้นหา ในกรณีนี้ผู้วิจัยได้สร้างไฟล์ออนโทโลยีในรูปแบบ

เอกสารโอดับเบิลยูแอลซึ่งเป็นรูปแบบมาตราฐานของการเอกสารที่อธิบายรายละเอียดของออนโทโลยีขึ้นมาทั้งสิ้น 2 ไฟล์ โดยมีรายละเอียดดังต่อไปนี้

1. ไฟล์ออนโทโลยีที่อธิบายรายละเอียดการจัดจำแนกสิ่งมีชีวิตประเภทสัตว์สะเทินน้ำสะเทินบกทางด้านชีววิทยา
2. ไฟล์ออนโทโลยีที่อธิบายรายละเอียดชื่อสามัญของสิ่งมีชีวิตประเภทสัตว์สะเทินน้ำสะเทินบกกับชื่อสปีชีส์ของสิ่งมีชีวิตนั้น

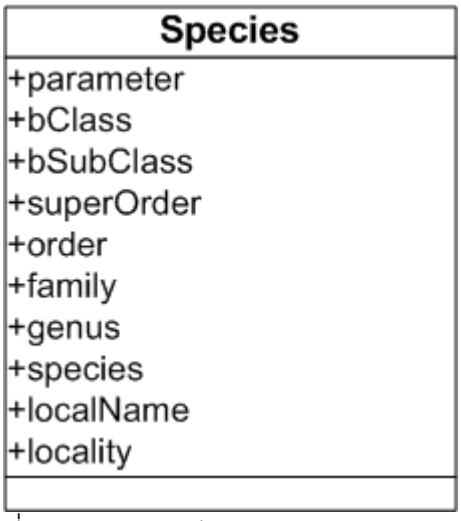
หลังจากนั้นผู้วิจัยทำการอัพโหลดไฟล์ออนโทโลยีทั้งสองเข้าไปในระบบผ่านองค์ประกอบโอดีบีเอ็มเอสเรียบร้อยแล้ว โอดีบีเอ็มเอสจะนำไฟล์ออนโทโลยีทั้งสองไปประมวลผลและสร้างเป็นออนโทโลยีของระบบไทส์ เพื่อให้นักพัฒนานำไปใช้ร่วมในการพัฒนาระบบค้นหาได้

### 6.2. **สร้างและทดสอบสปาร์เคิลที่จะนำไปไปใช้งาน**

จากความต้องการของระบบที่ออกแบบให้สามารถค้นหาข้อมูลของสิ่งมีชีวิตได้จากชื่อสามัญและสามารถจัดกลุ่มสิ่งมีชีวิตตามองค์ความรู้ทางด้านชีววิทยาได้ ผู้วิจัยจึงพัฒนาสปาร์เคิลขึ้นมาใช้ 2 รูปแบบ คือ สปาร์เคิลสำหรับสืบค้นหาข้อมูลทางชีววิทยาของสิ่งมีชีวิต และสปาร์เคิลสำหรับสืบค้นข้อมูลของสิ่งมีชีวิตทางชื่อสามัญ โดยผู้วิจัยได้นำสปาร์เคิลทั้งสองไปทดสอบสืบค้นโดยใช้องค์ประกอบโอดีบีเอ็มเอส เพื่อทดสอบผลการค้นคืนจากสปาร์เคิลดังกล่าวและตรวจสอบค่าของผลการค้นคืนที่ได้จากสปาร์เคิลทั้งสองนั้น

### 6.3. **สร้างคลาสสำหรับใช้งาน**

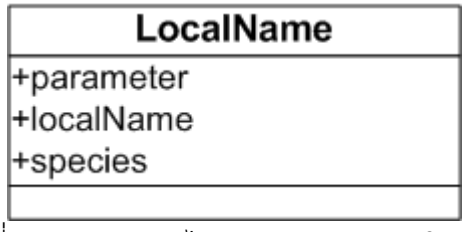
ในการพัฒนาระบบไทส์นั้นผู้วิจัยได้ทำการสร้างคลาสขึ้นมา 2 คลาสเพื่อทำหน้าที่เก็บข้อมูลสำหรับใช้ในการพัฒนาระบบค้นหา โดยคลาสแรกเป็นคลาสที่มีชื่อว่า Species เป็นคลาสที่ทำหน้าที่เก็บข้อมูลในการจัดการจำแนกสิ่งมีชีวิตทางชีววิทยาของสิ่งมีชีวิตนั้นๆ ไว้ในคลาส โดยในคลาส Species มีรายละเอียดตามที่แสดงในภาพที่ 19




ภาพที่ 19 แสดงคลาสไดอะแกรมของคลาสสปีชีส์

จากภาพที่ 19 เป็นคลาสไดอะแกรมของคลาสสปีชีส์ ซึ่งภายในคลาสจะประกอบไปด้วยตัวแปรที่ทำหน้าที่เก็บข้อมูลในการจัดจำแนกสิ่งมีชีวิตทางด้านชีววิทยาของสิ่งมีชีวิตแต่ละชนิดไว้ในคลาสทั้งหมด ซึ่งเราสามารถดึงข้อมูลจากออนโทโลยีในการจัดจำแนกสิ่งมีชีวิตทางด้านชีวิตของสิ่งมีชีวิตที่ต้องการเข้าสู่อ็อบเจกต์นี้ได้ เราก็จะสามารถนำอ็อบเจกต์ที่มีข้อมูลในการจัดจำแนกสิ่งมีชีวิตทางด้านชีววิทยาไปใช้ในการพัฒนาระบบค้นหาข้อมูลได้

จากนั้นผู้วิจัยสร้างคลาสอีกคลาสที่มีชื่อว่า "LocalName" เพื่อทำหน้าที่เก็บข้อมูลชื่อสามัญของสิ่งมีชีวิตและสปีชีส์ของสิ่งมีชีวิตนั้น โดยมีรายละเอียดของคลาสตามที่ปรากฏตามคลาไดอะแกรมที่แสดงตามภาพที่ 20




ภาพที่ 20 แสดงคลาสไดอะแกรมของคลาสโลคัลเนม

จากภาพที่ 20 จะพบว่าภายในคลาส LocalName มีตัวแปร 2 ตัวที่ทำหน้าที่เก็บข้อมูลชื่อสามัญของสิ่งมีชีวิตแต่ละชนิดและเก็บชื่อสปีชีส์ของสิ่งมีชีวิตนั้นๆไว้ ซึ่งจากการสร้างคลาสนี้หากใช้งานร่วมกับออนโทโลยีที่ทำหน้าที่อธิบายชื่อสามัญเข้ากับสปีชีส์ของสิ่งมีชีวิตนั้นๆ จะทำให้เราสามารถทราบได้ว่าชื่อสามัญของสิ่งมีชีวิตนั้นๆ มีชื่อสปีชีส์ว่าอย่างไรหรือในทางกลับกันเราก็สามารถทราบได้ว่าสิ่งมีชีวิตที่มีชื่อสปีชีส์นั้นๆมีชื่อสามัญว่าอย่างไรได้เช่นเดียวกัน

### 6.4. ระบุโอดีเอที่จะใช้ภายในคลาส

หลังจากที่สร้างคลาสสปีชีส์เก็บข้อมูลทางด้านชีววิทยาของสิ่งมีชีวิตเสร็จเรียบร้อยแล้วในขั้นตอนที่ 6.3 ผู้วิจัยก็นำสปาร์เคิลและค่าของผลการค้นคืนที่ได้จากการสร้างในขั้นตอนที่ 6.2 มาใช้งานด้วยองค์ประกอบโอดีเอภายในคลาส Species เพื่อดึงข้อมูลจากออนโทโลยีในการจัดจำแนกสิ่งมีชีวิตทางด้านชีววิทยาที่ต้องการมาใช้งาน ดังที่แสดงในภาพที่ 21

```
@Definition(query = "PREFIX oboInOwl: <http://www.geneontology.org/formats/oboInOwl#>"
      + " PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>"
      + " PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>"
      + " PREFIX owl:<http://www.w3.org/2002/07/owl#>"
      + " PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>"
      + " SELECT ?resource ?name ?resourcetype ?typelabel ?sourceref"
      + " WHERE {"
      + " ?source rdfs:label ?sourceref."
      + " ?source rdfs:subClassOf+ ?resource."
      + " ?resource rdfs:label ?name."
      + " ?resource rdfs:subClassOf ?blanknode."
      + " ?blanknode owl:someValuesFrom ?resourcetype."
      + " ?resourcetype rdfs:label ?typelabel."
      + " FILTER (contains(lcase(?sourceref), lcase(?parameter)))."
      + " FILTER (!(isBlank(?resource)))." + "}", entity = "amphibian-th")
public class Species implements Serializable {

   @Mapper
   @Parameter("parameter")
   private String parameter;

   @Variable(target = "name", filter = "typelabel", compare = Comparator.EQUAL, equal = "class")
   private String bClass;
   @Variable(target = "name", filter = "typelabel", compare = Comparator.EQUAL, equal = "subclass")
   private String bSubClass;

   @Variable(target = "name", filter = "typelabel", compare = Comparator.EQUAL, equal = "superorder"
   private String superOrder;

   @Variable(target = "name", filter = "typelabel", compare = Comparator.EQUAL, equal = "order")
   private String order;

   @Variable(target = "name", filter = "typelabel", compare = Comparator.EQUAL, equal = "family")
   private String family;

   @Variable(target = "name", filter = "typelabel", compare = Comparator.EQUAL, equal = "genus")
   private String genus;

   @Variable(target = "sourceref")
   private String species;

   @Property(name = "localname")
   private String localname;

   @Property(name = "local")
   private String locality;
```

ภาพที่ 21 แสดงการใช้งานโอดีเอภายในคลาสสปีชีส์

จากนั้นแล้ว ผู้วิจัยได้ใช้องค์ประกอบโอดีเอภายในคลาส LocalName เพื่อดึงข้อมูลจากออนโทโลยีที่อธิบายชื่อสามัญของสิ่งมีชีวิตและสปีชีส์ของสิ่งมีชีวิตนั้นๆ เข้าด้วยกันดังที่แสดงในภาพที่ 22

```
@Definition(query=" PREFIX oboInOwl: <http://www.geneontology.org/formats/oboInOwl#>"
   +" PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>"
   +" PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>"
   +" PREFIX biodb:<http://www.biologydb/element#>"
   +" SELECT ?localname ?species"
   +" WHERE {"
   +" ?element biodb:name ?parameter."
   +" ?element biodb:name ?localname."
   +" ?element biodb:localnameof ?species."
   +" }")
public class LocalName {

   @Parameter("parameter")
   private String parameter;

   @Variable(target="localname")
   private String localName;

   @Variable(target="species")
   private String Species;
```

ภาพที่ 22 แสดงการใช้งานโอดีเอภายในคลาส LocalName

### 6.5. ระบุคลาสพาธและคีย์ของคลาสพาธภายในอ็อบเจกต์แมปเปอร์

เนื่องจากระบบไทส์ที่สร้างขึ้น ถูกพัฒนาขึ้นในรูปแบบเว็บแอพพลิเคชันด้วยสปริงเฟรมเวิร์ก สิ่งหนึ่งที่สามารถทำได้เพื่อทำให้เว็บแอพพลิเคชันที่ถูกพัฒนาขึ้นมีประสิทธิภาพเพิ่มสูงขึ้น คือการประกาศเรียกใช้งานคลาสที่ต้องการไว้ในคอนเท็คของสปริงเฟรมเวิร์ก ซึ่งไฟล์คอนเท็คของสปริงเฟรมเวิร์กดังกล่าวนี้จะถูกเรียกขึ้นอ่านก่อนทุกครั้งที่มีการสั่งเริ่มการทำงานเว็บแอพพลิเคชันและสร้างอินสแตนซ์ของคลาสที่ถูกระบุไว้นั้นในรูปแบบซิงเกิลตันแพทเทิร์น

เพื่อให้ระบบไทส์มีการเรียกใช้องค์ประกอบอ็อบเจกต์แมปเปอร์ได้อย่างมีประสิทธิภาพ ผู้วิจัยระบุคลาสอ็อบเจกต์ให้ถูกเรียกใช้ภายในคอนเท็คของสปริงดังรายละเอียดในภาพที่ 23

```
<bean id="objectMapper" class="com.tais.mapping.ObjectMapper">
  <property name="objectMappingMap">
    <map>
      <entry>
        <key>
          <value>species</value>
        </key>
        <value>com.tais.object.Species</value>
      </entry>
      <entry>
        <key>
          <value>localname</value>
        </key>
        <value>com.tais.object.LocalName</value>
      </entry>
    </map>
  </property>
</bean>
```

ภาพที่ 23 แสดงการประกาศสร้างอินสแตนซ์ของคลาสอ็อบเจกต์แมปเปอร์จากสปริงคอนเท็ค

จากภาพที่ 23 เป็นการประกาศสร้างอินสแตนซ์ของคลาสอ็อบเจกต์แมปเปอร์จากภายในสปริงคอนเท็ค จะพบว่าผู้ใช้งานจะต้องทำการระบุคลาสพาธและคีย์ของแต่ละคลาสพาธนั้นภายในตัวแปรที่มีชื่อว่าออปเจ็กตเมปเปอร์เมป (objectMappingMap) ซึ่งตัวแปรดังกล่าวเป็นตัวแปรที่เก็บข้อมูลแบบเมปที่ระบุคลาสพาธและคีย์ของแต่ละคลาสพาธนั้น

## 6.6. เรียกใช้งานอ็อบเจกต์ผ่านองค์ประกอบอ็อบเจกต์แมปเปอร์

หลังจากที่ผู้วิจัยได้ทำการพัฒนาองค์ประกอบต่างๆเสร็จเรียบร้อย ผู้วิจัยก็จะได้คลาสของอ็อบเจกต์ที่มีข้อมูลที่พร้อมนำไปพัฒนาระบบค้นหาเชิงความหมายได้ ผ่านการเรียกใช้จากองค์ประกอบอ็อบเจกต์แมปเปอร์ โดยผู้วิจัยได้นำอ็อบเจกต์ดังกล่าวไปพัฒนาให้ระบบไทส์มีความสามารถดังต่อไปนี้

### 6.6.1. การค้นหาข้อมูลทางชีววิทยาสิ่งมีชีวิตจากชื่อทั่วไปสิ่งมีชีวิต

โดยทั่วไปแล้วสิ่งมีชีวิตแต่ละชนิดจะมีชื่อเฉพาะที่ใช้ในทางวิทยาศาสตร์แต่ในขณะเดียวกันสิ่งมีชีวิตแต่ละชนิดก็จะมีชื่อสามัญที่ใช้เรียกทั่วไปตามท้องถิ่นที่พบสิ่งมีชีวิตประเภทนั้นด้วย ผู้วิจัยจึงพัฒนาระบบเพื่อรองรับความสามารถดังกล่าวนี้ผู้วิจัย ได้สร้างออนโทโลยีที่บอกถึงชื่อสามัญของสิ่งมีชีวิตแต่ละชนิดและนำเข้าสู่ระบบ เพื่อให้ผู้ใช้งานสามารถค้นหาข้อมูลของสิ่งมีชีวิตโดยใช้เพียงชื่อสามัญได้ จากนั้นผู้วิจัยได้สร้างคลาสที่ทำหน้าที่เก็บข้อมูลขึ้นมา 2 คลาส

คลาสแรกคือคลาสที่มีชื่อว่าโลคัลเนม (LocalName) และคลาสที่มีชื่อว่าสปีชีส์ (Species) โดยคลาสที่มีชื่อว่าโลคัลเนมจะเป็นคลาสสำหรับดึงข้อมูลทางออนโทโลยีเพื่อทำให้ทราบว่าสิ่งมีชีวิตชนิดนั้นๆ มีชื่อ สามัญชนิดและมีชื่อทางวิทยาศาสตร์ว่าอย่างไร

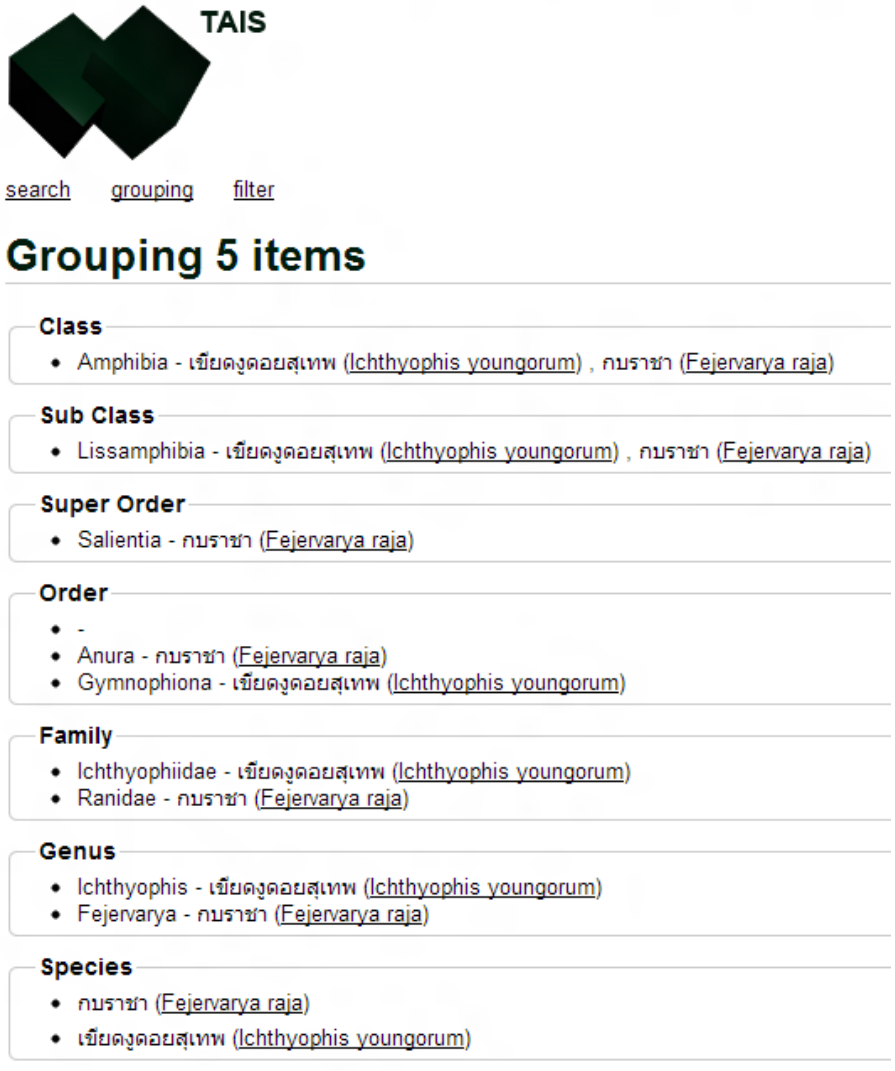
ซึ่งเมื่อเราทราบแล้วว่าสิ่งมีชีวิตนั้นมีชื่อสามัญและมีชื่อทางวิทยาศาสตร์หรือสปีชีส์ว่าอย่างไร ก็จะทำให้เราสามารถดึงข้อมูลทางชีววิทยาทั้งหมดของสิ่งมีชีวิตนั้นๆได้จากออนโทโลยีในการจัดจำแนกสิ่งมีชีวิตที่ระบุชื่อทางด้านวิทยาศาสตร์ของสิ่งมีชีวิตนั้นๆได้ โดยในการดึงข้อมูลทางออนโทโลยีในการจัดจำแนกสิ่งมีชีวิตทางชีววิทยาดังกล่าวออกมาใช้นั้น ผู้วิจัยได้สร้างคลาสที่สองที่มีชื่อว่าสปีชีส์ เพื่อทำหน้าที่เก็บข้อมูลดังกล่าวหลังจากที่ทำการเรียกใช้งานคลาสสปีชีส์ผ่านทางองค์ประกอบอ็อบเจกต์แมปเปอร์ก็จะทำให้เราได้อ็อบเจกต์สปีชีส์ที่มีข้อมูลทั้งหมด ของสิ่งมีชีวิตนั้นๆ ตามออนโทโลยีการจัดจำแนกสิ่งมีชีวิตประเภทสัตว์สะเทินน้ำสะเทินบกทางด้านชีววิทยา และสามารถนำอ็อบเจกต์ดังกล่าวไปแสดงผลการค้นคืนให้กับผู้ใช้ได้ดังที่แสดงในภาพที่ 24

ภาพที่ 24 แสดงการแสดงข้อมูลของสิ่งมีชีวิตจากการสืบค้นด้วยชื่อสามัญด้วยระบบไทส์

6.6.2. <u>การค้นหาความสัมพันธ์ระหว่างสิ่งมีชีวิตจากองค์ความรู้ในการจัดจำแนกสิ่งมีชีวิตทางด้านชีววิทยา</u>

ด้วยความสามารถของออนโทโลยีที่สามารถอธิบายความสัมพันธ์ระหว่างสิ่งต่างๆได้ ทำให้เราสามารถนำความสามารถดังกล่าวของออนโทโลยีมาใช้เพื่อจัดกลุ่มของสิ่งมีชีวิต ตามการจัดจำแนกสิ่งชีวิตทางด้านชีววิทยาได้อีกด้วย ตัวอย่างเช่น หากผู้ใช้ต้องการที่ทราบว่า สิ่งมีชีวิต 4 ชนิด ได้แก่ กบเหลือง (Yellow Frog) , รานา ฟิชีริ (Rana fisheri) , กาสโทรดีคา อันโทเมีย (Gastrotheca antomia) และ เขียดงูหัวแหลม (Pointed-snouted Caecilian) มีความสัมพันธ์กันอย่างไรทางชีววิทยา เราก็จะสามารถหาคำตอบนี้โดยใช้ข้อมูลที่ถูกสร้างขึ้นจาก ออนโทโลยีในการ

จัดจำแนก สิ่งมีชีวิตประเภทสัตว์สะเทินสะเทินบกทางด้านชีววิทยาร่วมกับออนโทโลยีที่อธิบายชื่อสามัญและชื่อทางวิทยาศาสตร์หรือสปีชีส์ของสิ่งมีชีวิตแต่ละชนิดในการช่วยหา คำตอบได้ จากข้อที่ 4.3.1 เราสามารถหาข้อมูลทางด้านชีววิทยาของสิ่งมีชีวิตได้ทั้งจาก ชื่อสามัญและชื่อสปีชีส์ของสิ่งมีชีวิตและได้ข้อมูลที่อยู่ในรูปของอ็อบเจกต์เรียบร้อยแล้วนั้นก็จะทำให้เราสามารถทราบได้ว่าสิ่งมีชีวิตทั้ง 4 ชนิดดังกล่าวมีการจัดจำแนกที่ใกล้เคียงกันอย่างไรบ้างตามองค์ความรู้ในการจัดจำแนกสิ่งมีชีวิตประเภทสัตว์สะเทินน้ำสะเทินบกและทำให้สามารถแสดงข้อมูลสรุปการจัดกลุ่มของสิ่งมีชีวิตตาม องค์ความรู้ดังกล่าวได้ดังที่แสดงในภาพที่ 25




ภาพที่ 25 แสดงการจัดกลุ่มของสิ่งมีชีวิตด้วยระบบไทส์

จากการความสามารถในการดึงข้อมูลสปีชีส์จากออนโทโลยี ทางด้าน ชีววิทยาของสิ่งมีชีวิตทั้ง 4 ชนิด ทำให้เราสามารถจัดกลุ่มของสิ่งมีชีวิตที่คล้ายคลึงได้ ดังตัวอย่างที่แสดงในภาพที่ 13 ทำให้เราทราบว่า กบเหลือง , รานา ฟิชชิริ , กาสโทรดีคา อันโทเมีย และ เขียดงูหัวแหลม

ถูกจัดอยู่ในชั้นแอมฟิเบีย (Class Amphibia) เหมือนกัน แต่กบเหลือง , รานา ฟิชีริ และ กาสโทรดีคา อันโทเมีย มีลักษณะที่ใกล้เคียงกันมากกว่า เขียดงูหัวแหลม เนื่องจากสิ่งมีชีวิตทั้งสามชนิดข้างต้นถูกจัดอยู่ในอันดับอะนูรา (Order Anura) เช่นเดียวกันทั้งสามชนิด ในขณะที่เขียดงูหัวแหลมถูกจัดอยู่ในอันดับ จิมโนฟิโอนา (Order Gymnophiona) นั่นหมายความว่ากบเหลือง , รานา ฟิชีริ และ กาสโทรดีคา อันโทเมีย มีความใกล้เคียงกันทางด้านชีววิทยามากกว่าเขียดงูหัวแหลมตามองค์ความรู้ในการจัดจำแนกสิ่งมีชีวิตทางด้านชีววิทยานั่นเอง

# บทที่ 7
# การทดสอบวิธีการโอโอเอ็ม

ในการวัดและประเมินผลผู้วิจัยได้ประเมินผลวิธีการโอโอเอ็มงาน 2 ด้าน คือ

## 7.1.ด้านประสิทธิผล (Effectiveness)

ในการประเมินความถูกต้องของการค้นหาข้อมูลนั้น ผู้วิจัยได้ประเมินจากการวัดผลการค้นคืน (Recall) และความถูกต้องของผลของการค้นคืน (Precision) โดยทดสอบจากความสามารถของระบบไทส์ ในการค้นหาข้อมูลของสิ่งมีชีวิตทางด้านชีววิทยาโดยระบุชื่อสามัญของสิ่งมีชิวิตตามสปีชีส์ของสิ่งมีชีวิตแต่ละชนิดโดยชื่อสามัญที่นำมาใช้ในการค้นหานั้นเป็นชื่อสามัญของสิ่งมีชีวิตที่ถูกระบุอยู่ในเอกสารของสำนักงานนโยบายและแผนทรัพยากรธรรมชาติและสิ่งแวดล้อม (Office of Environmental Policy and Planning) กระทรวงทรัพยากรธรรมชาติและสิ่งแวดล้อม (Ministry of Natural Resources and Environment) ซึ่งเอกสารชุดดังกล่าวเป็นเอกสารชุดเดียวกับที่ผู้วิจัยใช้สร้างออนโทโลยีที่อธิบายความสัมพันธ์ระหว่างชื่อสามัญและสปีชีส์ของสิ่งมีชีวิตแต่ละชนิดที่ถูกใช้ในระบบไทส์

จากนั้นผู้วิจัยได้ทำการสุ่มทดสอบค้นหาสิ่งมีชิวิตจำนวนทั้งสิ้น 30 ชนิดจากจำนวนสิ่งมีชีวิตทั้งหมด 5,428 สปีชีส์ จากการทดสอบได้ผลการทดลองดังตารางที่ 1

| ลำดับ | คำค้น | ชื่อสปีชีส์ | ผลการค้นคืน | ความถูกต้องการค้นคืน |
|---|---|---|---|---|
| 1 | กบเขาสูง | Rana alticola | 1 | 1 |
| 2 | เขียดงูหูแหลม | Ichthyophis acuminatus | 1 | 1 |
| 3 | ป๊าดงูหูแหลม | - | 0 | 0 |
| 4 | เขียดทราย | Occidozyga martensii | 1 | 1 |
| 5 | ป๊าดทราย | - | 0 | 0 |

ตารางที่ 1 แสดงตัวอย่างผลการทดลองวัดประสิทธิผลของวิธีการโอโอเอ็ม

จากตารางที่ 1 แสดงตัวอย่างจากผลการทดลองวัดประสิทธิผลของวิธีการ โอโอเอ็มที่พัฒนาขึ้นในระบบไทส์พบว่า ระบบไทส์สามารถให้ผลการค้นคืนข้อมูลของสิ่งมีชีวิตจากชื่อสามัญได้เฉพาะในกรณีที่คำค้นที่เป็นชื่อสามัญของสิ่งมีชีวิตถูกระบุไว้ในออนโทโลยี เช่น คำค้นที่ชื่อว่าเขียดงูหูแหลมระบบไทส์จะสามารถใช้ออนโทโลยีในการค้นหาและตอบได้ว่าเขียดงูหูแหลมคือสิ่งมีชีวิตที่มีสปีชีส์คือ Ichthyophis acuminatus

จากการศึกษาของผู้วิจัยพบว่าในทางภาคเหนือของประเทศมีคำท้องถิ่นที่ใช้เรียกสิ่งมีชีวิตประเภทเขียดว่า “ป๊าด” เมื่อผู้วิจัยนำข้อมูลดังกล่าวมาช่วยในค้นหา เช่น ค้นหาด้วยคำค้นว่า ป๊าดงูหูแหลม ระบบไทส์จะไม่สามารถค้นหาข้อมูลของสิ่งมีชีวิตจากคำค้นดังกล่าวได้

เหตุผลสำคัญที่ทำให้การค้นหาด้วยชื่อสามัญพบหรือไม่พบข้อมูลทางด้านชีววิทยาของสิ่งมีชีวิตแต่ละชนิดเนื่องจากออนโทโลยีที่ใช้ในระบบค้นหา เนื่องจากระบบไทส์ในปัจจุบันมีเพียงออนโทโลยีที่ทำหน้าที่อธิบายความสัมพันธ์ระหว่างชื่อสามัญกับชื่อสปีชีส์ของสิ่งมีชีวิตแต่ละชนิด ซึ่งยังไม่ครอบคลุมถึงองค์ความรู้ที่เกี่ยวข้องกับภาษาท้องถิ่น ทำให้ระบบค้นหาเชิงความหมายยังไม่เข้าใจว่า คำว่า เขียด มีความหมายเหมือนกับคำว่า ป๊าด ทำให้ระบบไทส์ยังไม่สามารถค้นหาข้อมูลได้พบ แต่จะหาพบในกรณีที่คำค้นมีข้อมูลตรงกับชื่อสามัญที่ถูกสร้างขึ้นในออนโทโลยีนั่นเอง

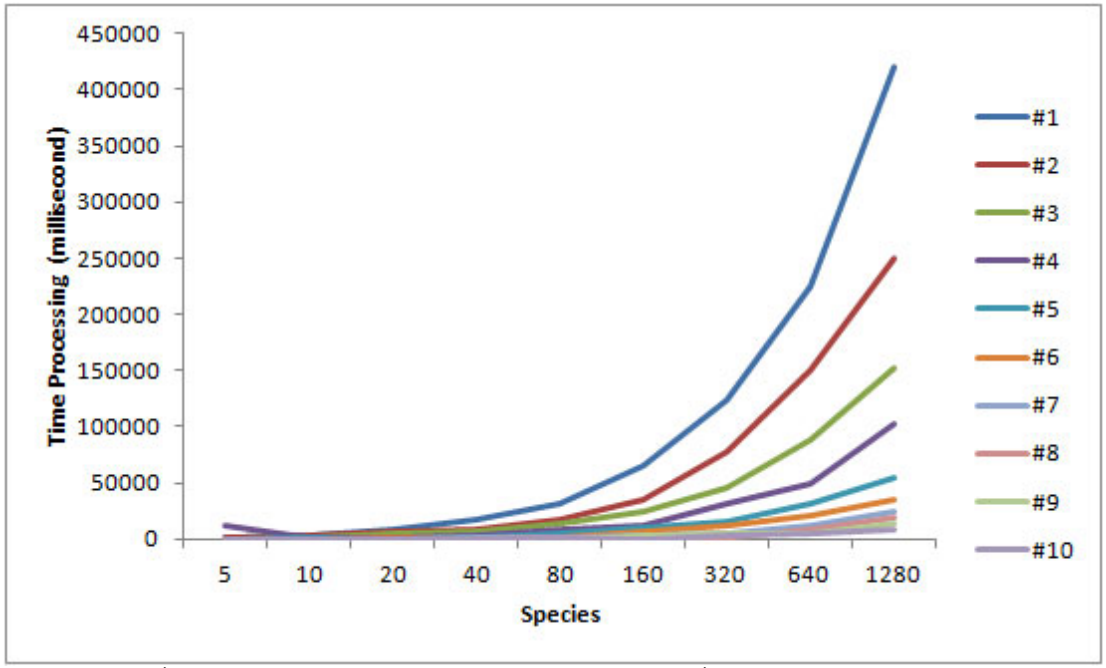
### 7.2. **ด้านประสิทธิภาพ** (Efficiency)

ในการวัดประสิทธิภาพของวิธีการโอโอเอ็มนั้น ผู้วิจัยได้ทดสอบโดยวัดความเร็วของการจัดกลุ่มของสิ่งมีชีวิตบนระบบไทส์โดยทดสอบบนสภาพแวดล้อมของกูเกิลแอพเอนจิน ซึ่งเป็นสภาพแวดล้อมสำหรับการพัฒนาระบบในลักษณะคลาวด์คอมพิวติงที่ได้รับความนิยมในปัจจุบัน โดยผู้วิจัยเลือกอินสแตนคลาสเอฟสี่ (F4 Class) ในการทดสอบ ซึ่งมีความเร็วของซีพียูที่ 2.4 กิกกะเฮิร์ทซ และความจุของแรมที่ 512 เมกะไบต์ นอกจากนั้นผู้วิจัยยังได้สร้างทาสก์คิว (Task-queue) ซึ่งเป็นบริการของกูเกิลแอพเอนจินสำหรับประมวลผลงานแบบต่อเนื่องโดยตั้งค่าความเร็วในการประมวลผลอยู่ที่ 500 ครั้งต่อวินาที (500/s) รวมถึงมีการใช้งานบริการเมมแคช (Memcache) ของกูเกิลแอพเอนจินเพื่อเก็บข้อมูลสิ่งมีชีวิตที่ถูกค้นหาออกมาแล้วเก็บไว้เพื่อให้การเรียกใช้ข้อมูลในครั้งต่อไปสามารถทำได้รวดเร็วขึ้น โดยในการทดลองนั้นผู้วิจัยได้ทำการทดลองในการจัดกลุ่ม ของสิ่งมีชีวิตโดยการสุ่มเลือกสิ่งมีชีวิตจำนวน 5 ชนิด และเพิ่มจำนวนสิ่งมีชีวิตเป็นจำนวน 2 เท่า ในแต่ละครั้งจนกระทั่งจำนวน สิ่งมีชีวิตที่ใช้ในการจัดกลุ่มมีจำนวน 1280 ชนิด ทำการทดลองทั้งสิ้น 10 รอบ ซึ่งสามารถแสดงตัวอย่างของผลการทดลองได้ดังตารางที่ 2

| จำนวนสิ่งมีชีวิต | เวลาในการประมวลผล ครั้งที่ 1 (วินาที) | เวลาในการประมวลผล ครั้งที่ 10 (วินาที) |
|---|---|---|
| 5 | 2.106 | 0.021 |
| 10 | 3.688 | 0.038 |
| 20 | 7.69 | 0.491 |
| 40 | 18.041 | 0.123 |
| 80 | 32.142 | 0.987 |
| 160 | 65.077 | 0.535 |
| 320 | 123.549 | 2.991 |
| 640 | 224.744 | 5.884 |
| 1280 | 420.457 | 9.369 |

ตารางที่ 2 แสดงการเปรียบเทียบเวลาที่ใช้ในการประมวลครั้งที่ 1 และครั้งที่ 10

ซึ่งเมื่อนำผลการทดลองที่ทำการวัดเวลาในการประมวลผลในการหาความสัมพันธ์ของสิ่งชีวิตตามจำนวนที่เพิ่มขึ้นในแต่ละรอบ จะสามารถสรุปผลการทดสอบได้ดังที่แสดงตามภาพที่ 22




ภาพที่ 26 แสดงกราฟผลการทดสอบการจัดกลุ่มของสิ่งมีชีวิตด้วยวิธีการโอโอเอ็ม

จากภาพที่ 26 จะพบว่า เวลาในการประมวลเพื่อหาข้อมูลทางด้านชีววิทยาและจัดกลุ่มสิ่งมีชีวิตเข้าด้วยกันจะเพิ่มขึ้นตามจำนวนสิ่งมีชีวิตแต่จะลดลงเรื่อยๆ ตามจำนวนรอบของการจัดกลุ่มของสิ่งมีชีวิต จากผลการทดสอบจะพบว่ารอบแรกของจัดกลุ่มของสิ่งมีชีวิตจำนวน 1,280 ชนิดใช้เวลาโดยรวมประมาณ 420 วินาที เนื่องจากระบบต้องทำการประมวลผลข้อมูลของสิ่งมีชีวิตเกือบทั้ง 1,280 ชนิด แต่ในขณะที่ รอบที่ 10 ที่ทำการจัดกลุ่มสิ่งมีชีวิต จำนวน 1280 ชนิดเช่นกันแต่กลับใช้เวลาเพียง 0.9 วินาทีเท่านั้น เนื่องจากข้อมูลของสิ่งมีชีวิตโดยส่วนใหญ่ถูกจัดเก็บอยู่ในแคชเรียบร้อยแล้ว ทำให้การเรียกใช้ข้อมูลสามารถทำได้อย่างรวดเร็วมากขึ้นอย่างมากนั่นเอง

# บทที่ 8
# สรุปผลการวิจัยและข้อเสนอแนะ

## 8.1.ข้อสรุป

ผู้วิจัยได้พัฒนาวิธีการสร้างระบบค้นหาเชิงความหมายด้วยวิธีการโอโอเอ็มซึ่งเป็นวิธีการที่มีองค์ประกอบที่ถูกออกแบบมาเพื่อสำหรับการพัฒนาระบบค้นหาเชิงความหมายทั้งสิ้น 3 องค์ประกอบ ได้แก่ องค์ประกอบที่หนึ่งโอดีเอเป็นองค์ประกอบที่ออกแบบเพื่อให้สามารถเรียกใช้งานข้อมูลจากออนโทโลยีได้ง่ายขึ้นลดความซับซ้อนของซอร์สโค้ดของตัวโปรแกรมลง โดยใช้ความสามารถของแอนโนเตชันในภาษาจาวา องค์ประกอบที่สองโอดีบีเอ็มเอสที่ถูกออกแบบมาเพื่อสำหรับปรับปรุง แก้ไข ข้อมูลของออนโทโลยีที่จะถูกนำไปใช้ในระบบค้นหาสามารถทำได้สะดวกมากยิ่งขึ้น และองค์ประกอบสุดท้ายอ็อบเจกต์แมปเปอร์ เป็นองค์ประกอบสำหรับเรียกใช้งานข้อมูลในรูปแบบอ็อบเจกต์สำหรับการพัฒนาโปรแกรมเชิงวัตถุ ผู้วิจัยได้ทำการทดสอบวิธีการโอโอเอ็มด้วยการสร้างระบบค้นหาที่มีชื่อว่าไทส์พบว่า ระบบดังกล่าวสามารถให้ผลการค้นคืนได้อย่างถูกต้องและสามารถเพิ่มสมรรถนะของระบบได้ด้วยการใช้องค์ประกอบอื่นๆช่วยในการพัฒนา เช่น การใช้แคชเพื่อเพิ่มความรวดเร็วในการเข้าถึงข้อมูลของระบบ

## 8.2.ข้อเสนอแนะ

วิธีการโอโอเอ็มสามารถนำไปประยุกต์ใช้ในการพัฒนาระบบค้นหาเชิงความหมายในองค์ความรู้อื่นๆได้อีกมากมาย เช่น การนำไปพัฒนาระบบค้นหาข้อมูลทางด้านชีววิทยาของพืช ซึ่งในการนำวิธีโอโอเอ็มไปใช้ในนั้นทำเพียงเพิ่มออนโทโลยีที่อธิบายรายละเอียดเกี่ยวกับข้อมูลทางด้านชีววิทยาของพืชเข้าไปในระบบผ่านองค์ประกอบโอดีบีเอ็มเอสและสร้างคลาสใหม่ที่ใช้โอดีเอสำหรับระบุสปาร์เคิลสำหรับสืบค้นข้อมูลของออนโทโลยีใหม่และค่าที่ต้องการใช้ก็สามารถพัฒนาระบบค้นหาข้อมูลทางด้านชีววิทยาของพืชได้ทันที

แต่อย่างไรก็ดีสิ่งยังคงต้องได้รับการพัฒนาสำหรับวิธีการโอโอเอ็ม คือ ความเร็วในการดึงข้อมูลเข้าสู่อ็อบเจกต์ขององค์ประกอบอ็อบเจกต์แมปเปอร์ ซึ่งยังคงสามารถพัฒนาให้มีความเร็วเพิ่มสูงขึ้นและใช้ทรัพยากรของระบบลดลงเพิ่มเติมได้อีกซึ่งแนวทางในการพัฒนาต่อไป ในประเด็นดังกล่าวนี้จะช่วยให้วิธีการโอโอเอ็มมีความสมบูรณ์ทั้งประสิทธิภาพและประสิทธิผลในการนำไปใช้ในการพัฒนาระบบค้นหาเชิงความหมายได้ต่อไป

# รายการอ้างอิง


[1] Cun-Gen GU Fang and CAO , Ontology Research and Existing Problems in Knowledge Engineering , Computer Science Vol. 31. , 2004.

[2] W3C , Resource Description Framework (RDF) [Online]. Available from: http://www.w3.org/RDF/ [2012, April 21]

[3] W3C , OWL Web Ontology Language [Online]. Available from: http://www.w3.org/TR/owl-features/ [2012 , April 21]

[4] W3C , SPARQL Query Language for RDF [Online]. Available from: http://www.w3.org/TR/rdf-sparql-query/ [2012 , April 21]

[5] Apache Software Foundation , Apache Jena [Online]. Available from: http://jena.apache.org [2012 , April 21]

[6] Oracle , The History of Java Technology [Online]. Available from: http://oracle.com.edgesuite.net/timeline/java/ [2012 ,April 27]

[7] TIOBE , TIOBE Programming Community Index for April 2013 [Online]. Available from: http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html [2012, April 27]

[8] Oracle , Annotations [Online]. Available from: http://docs.oracle.com/javase/1.5.0/docs/guide/language/annotations.html [2012 , April 21]

[9] A Cyment et al , Improving AOP systems evolvability by decoupling advices from base code, RAM-SE'06– ECOOP'06 Workshop on Reflection, AOP, and Meta-data for Software Evolution, 2006.

[10] Biberstein et al, Instrumenting annotated programs , 1st ACM/USENIX international conference on Virtual execution environments., 2005.

[11] Cordy, J.R. et al, HSML: design directed source code hot spots , IWPC 2001, 2001.

[12] Leho Nigul, Ernest Mah , Software maintainability benefits from annotation-driven code., IEEE International Conference on Software Maintenance , 2009

[13] SpringSource , Spring Framework [Online]. Available from: http://www.springsource.org/spring-framework [2012 , April 21]

[14] Google , Google AppEngine [Online]. Available from: https://developers.google.com/appengine/ [2012, April 21]

[15] Shao-min Zhang, Jian-yi Guo, Zheng-tao Yu, Chun-ya Lei, Cun-li Mao, Hai-xiong Wang , An Approach of Domain Ontology Construction Based on Resource Model and Jena. , Third International Symposium on Information Processing, 2010

[16] Jhin, I.-H., Choi, N.-M., & Cho, E.-S. , Integration of RDF Processing into a Programming Language. , Fifth International Joint Conference on INC, IMS and IDC, 2009.

[17] Shi, L., Fan, L., & Meng, Z. , The Research of Using Jena in the Semantic-Based Online Learning Intelligent Behavior Analysis System. Fifth International Joint Conference on INC, 2009

[18] Nigul, L., & Mah, E., Software maintainability benefits from annotation-driven code. , IEEE International Conference on Software Maintenance, 2009

# ภาคผนวก

# ภาคผนวก ก

## ตารางผลการทดลองการวัดประสิทธิผลด้วยวิธีโอโอเอ็ม

| ลำดับ | คำค้น | ชื่อสปีชีส์ | ผลการค้นคืน | ความถูกต้องการค้นคืน |
|---|---|---|---|---|
| 1 | กบเขาสูง | Rana alticola | 1 | 1 |
| 2 | กบดอยอินทนนท์ | Rana schmackeri | 1 | 1 |
| 3 | กบราชา | Fejervarya raja | 1 | 1 |
| 4 | กบหมื่น | Limnonectes kuhlii | 1 | 1 |
| 5 | กบเขาดอย | Rana montivaga | 1 | 1 |
| 6 | กบเขาหลังตอง | Rana chalconota | 1 | 1 |
| 7 | กบชะง้อนหินภาคใต้ | Rana hosii | 1 | 1 |
| 8 | กบเขาทองลาย | Rana luctuosa | 1 | 1 |
| 9 | กบหลังไพล | Rana lateralis | 1 | 1 |
| 10 | กบสังขละ | Rana leptoglossa | 1 | 1 |
| 11 | กบหลังจุด | Rana signata | 1 | 1 |
| 12 | กบหงอนมลายู | Limnonectes plicatellus | 1 | 1 |
| 13 | กบหลังลายเฉียง | Rana miopus | 1 | 1 |
| 14 | กบเขาสอยดาว | Rana montivaga | 1 | 1 |
| 15 | กบตะนาวศรี | Ingerana tenasserimensis | 1 | 1 |
| 16 | เขียดงูหูแหลม | Ichthyophis acuminatus | 1 | 1 |
| 17 | เขียดงูศุภชัย | Ichthyophis supachaii | 1 | 1 |
| 18 | เขียดทราย | Occidozyga martensii | 1 | 1 |
| 19 | เขียดจะนา | Occidozyga lima | 1 | 1 |
| 20 | เขียดงูดอยสุเทพ | Ichthyophis youngorum | 1 | 1 |
| 21 | เขียดเหลือง | Rana lateralis | 1 | 1 |
| 22 | เขียดลื่น | Occidozyga laevis | 1 | 1 |
| 23 | ป๊าดงูหูแหลม | - | 0 | 0 |

| | | | | |
|---|---|---|---|---|
| 24 | ป๊าดงูศุภชัย | - | 0 | 0 |
| 25 | ป๊าดทราย | - | 0 | 0 |
| 26 | ป๊าดจะนา | - | 0 | 0 |
| 27 | ป๊าดงูดอยสุเทพ | - | 0 | 0 |
| 28 | ป๊าดเหลือง | - | 0 | 0 |
| 29 | ป๊าดดลื่น | - | 0 | 0 |
| 30 | อึ่งกลายหัวแหลม | Megophrys nasuta | 1 | 1 |

## ภาคผนวก ข

## ตารางผลการทดลองการวัดประสิทธิภาพของวิธีโอโอเอ็ม

| จำนวนสิ่งมีชีวิต | เวลาในการประมวลผลของการทดลองครั้งที่ (วินาที) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 5 | 2.106 | 0.816 | 0.39 | 12.87 | 0.43 | 0.02 | 0.019 | 0.022 | 0.023 | 0.021 |
| 10 | 3.688 | 2.866 | 1.644 | 1.751 | 0.861 | 0.039 | 0.035 | 0.043 | 0.047 | 0.038 |
| 20 | 7.69 | 7.367 | 4.432 | 2.132 | 0.493 | 0.89 | 0.064 | 0.062 | 0.067 | 0.491 |
| 40 | 18.041 | 7.941 | 6.734 | 2.651 | 2.189 | 0.556 | 0.876 | 0.12 | 0.114 | 0.123 |
| 80 | 32.142 | 17.326 | 13.242 | 8.068 | 5.653 | 2.292 | 1.431 | 0.959 | 0.306 | 0.987 |
| 160 | 65.077 | 35.068 | 23.724 | 11.896 | 11.124 | 6.331 | 4.073 | 1.207 | 2.418 | 0.535 |
| 320 | 123.549 | 77.861 | 45.56 | 30.851 | 16.098 | 12.018 | 4.378 | 1.097 | 4.74 | 2.991 |
| 640 | 224.744 | 150.611 | 87.575 | 48.594 | 31.478 | 20.493 | 12.379 | 9.116 | 4.917 | 5.884 |
| 1280 | 420.457 | 249.241 | 152.105 | 102.05 | 58.844 | 34.327 | 24.534 | 20.069 | 13.952 | 9.369 |

# ภาคผนวก ค

# ซอร์สโค้ดของเดฟฟินิชันแอนโนเตชัน

```
import java.lang.annotation.ElementType;

import java.lang.annotation.Retention;

import java.lang.annotation.RetentionPolicy;

import java.lang.annotation.Target;


@Retention(RetentionPolicy.RUNTIME)

@Target(ElementType.TYPE)


public @interface Definition {

        String query();

        String entity() default "";

}
```

# ภาคผนวก ง

# ซอร์สโค้ดของแมปเปอร์แอนโนเตชัน

```
import java.lang.annotation.ElementType;

import java.lang.annotation.Retention;

import java.lang.annotation.RetentionPolicy;

import java.lang.annotation.Target;



@Retention(RetentionPolicy.RUNTIME)

@Target(ElementType.FIELD)

public @interface Mapper {

}
```

# ภาคผนวก จ

## ซอร์สโค้ดของพารามิเตอร์แอนโนเตชัน

```
import java.lang.annotation.ElementType;

import java.lang.annotation.Retention;

import java.lang.annotation.RetentionPolicy;

import java.lang.annotation.Target;



@Retention(RetentionPolicy.RUNTIME)

@Target(ElementType.FIELD)



public @interface Parameter {

        String value();

}
```

# ภาคผนวก ฉ

## ซอร์สโค้ดของพร็อพเพอร์ตีแอนโนเตชัน

```
import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.FIELD)

public @interface Property {
        String name();
}
```

# ภาคผนวก ซ

## ซอร์สโค้ดของวาริเอเบิลแอนโนเตชัน

```
import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;

import com.ma.enumulation.Comparator;

@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.FIELD)
public @interface Variable {
	String target();
	String filter() default "";
	Comparator compare() default Comparator.NONE;
	String equal() default "";
}
```

# ประวัติผู้เขียนวิทยานิพนธ์

นายสุทธิรักษ์ แสงจันทร์ เกิดวันที่ 28 มีนาคม พ.ศ. 2526 กรุงเทพมหานคร จบการศึกษาระดับปริญญาบัณฑิต จากคณะครุศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2545 มีประสบการณ์การในทำงานด้านพัฒนาซอฟต์แวร์ และการออกแบบ สื่อดิจิตอล ปัจจุบันทำหน้าที่เป็น นักพัฒนาซอฟต์แวร์ที่บริษัท เอบีซีเทค จำกัด กรุงเทพมหานคร