

ระบบแมปข้อมูลจากฐานข้อมูลเป็นออนโทโลยี

Database to Ontology Mapping System

ณัฐธิดา บุตรพรหม และปุษะพี ศิริแสงตระกูล

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น

nattida.b@kku.ac.th, pusadee@kku.ac.th

บทคัดย่อ

งานวิจัยนี้นำเสนอการพัฒนาเครื่องมือแมปข้อมูลจากฐานข้อมูลเป็นออนโทโลยี (ExtendNectecD2OMAP) โดยมีวัตถุประสงค์เพื่อเพิ่มความสามารถเครื่องมือแมปข้อมูลจากฐานข้อมูลเป็นออนโทโลยีซึ่งนำเสนอโดย NECTEC (NectecD2OMAP) ให้สามารถรองรับไฟล์ความรู้ออนโทโลยีที่สร้างจากหลายโปรแกรม และให้รองรับฐานข้อมูล เชิงสัมพันธ์ที่จัดการด้วยระบบจัดการฐานข้อมูลที่หลากหลายขึ้นด้วยวิธีเปรียบเทียบความสอดคล้องระหว่างฐานข้อมูลและออนโทโลยี เพื่อประเมินความสามารถของระบบที่พัฒนาขึ้น ผู้วิจัยได้ทดสอบระบบโดยใช้ชุดข้อมูลจากงานวิจัยของ Jerry ผลการทดสอบพบว่า ExtendNectecD2OMAP มีความสามารถเพิ่มขึ้นจาก NectecD2OMAP สามารถรองรับไฟล์ออนโทโลยีที่สร้างจาก OE5, Protégé และ Altova และรองรับฐานข้อมูลของระบบจัดการฐานข้อมูล PostgreSQL, MS-Access, SQL Server และ MySQL

□ คำสำคัญ :

ออนโทโลยี ฐานข้อมูลเชิงสัมพันธ์ แมปข้อมูล

Abstract

This paper presents the development of Database-to-Ontology Mapping tools. The purpose of this system is to extend the tools capabilities that developed by NECTEC in order to support variety ontology development tools and variety DBMS. The development approach is based on matching method. In order to evaluate the capability of the proposed system, we use data from the Jerry's research to evaluate the system comparing to the NectecD2OMAP. The results showed that the proposed system has more capability than the original tool. The proposed system can support the ontologies that developed by OE5, Protégé and Altova. Moreover, it can work together with database that created by the PostgreSQL, MS-Access, SQL Server and MySQL Database Management System.

Keywords :

ontology, relational database, schema matching



1. บทนำ

เว็บเชิงความหมาย (Semantic Web) [1] เป็นเทคโนโลยี ที่ใช้บูรณาการ และจัดระเบียบข้อมูลเชิงความหมาย ทำให้มีการเชื่อมโยงเครือข่ายของข้อมูลบนเว็บไซต์ต่างๆ และส่งผลให้การสืบค้นเป็นไปอย่างมีความสามารถ และสามารถสร้างความสัมพันธ์ให้กับข้อมูลจากหลายแหล่ง ออนโทโลยี [1] เป็นวิธีการหนึ่งที่มีบทบาทสำคัญ ที่ถูกนำมาใช้ในการเพิ่มความสามารถในการทำงานร่วมกันเชิงความหมายระหว่างแหล่งข้อมูลที่มีความหลากหลาย ปัจจุบันข้อมูลส่วนใหญ่ถูกจัดเก็บในรูปแบบฐานข้อมูลเชิงสัมพันธ์ [1] ซึ่งการทำงานร่วมกัน จำเป็นต้องมีการแมปโครงสร้างจากแหล่งข้อมูลต่างๆ เข้ากับออนโทโลยี จากการศึกษาพบว่า มีงานวิจัยที่นำเสนอเครื่องมือ และวิธีแมปฐานข้อมูลกับออนโทโลยีอย่างแพร่หลาย

เครื่องมือบางเครื่องมือมีข้อจำกัดบางประการ เช่น MapOnto [2,11] พัฒนาเพื่อใช้งานร่วมกับ Protégé-2.1 และนำเข้าได้เฉพาะไฟล์ SQL DDL, D2OMapper [3] สามารถแมปเฉพาะตารางความสัมพันธ์กับคลาสออนโทโลยี ที่มีความสัมพันธ์ RONTO [4] เป็นเครื่องมือแมป ซึ่งเป็นปลั๊กอินของ Protégé-2.0+ รวมทั้ง Database to Ontology Mapping [5, 6] ซึ่งพัฒนาโดยศูนย์เทคโนโลยี อิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ ใช้แมปฐานข้อมูล กับออนโทโลยี โดยสามารถรองรับฐานข้อมูลเชิงสัมพันธ์ และทำงานร่วมกับออนโทโลยีที่ถูกสร้างโดย OE5 ดังนั้น ในงานวิจัยนี้มุ่งเน้นที่การพัฒนาเครื่องมือแมปฐาน ข้อมูลเชิงสัมพันธ์กับออนโทโลยีให้มีความสามารถมากขึ้น และสามารถรองรับการทำงานร่วมกับออนโทโลยีที่สร้าง จากหลากหลายเครื่องมือได้

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ออนโทโลยี คือวิธีการอธิบายแนวความคิดเฉพาะด้าน ด้วยแนวคิดที่มีความสัมพันธ์กันแบบลำดับชั้น หรือกราฟ

[1] ออนโทโลยีถูกสร้างขึ้นแทนความรู้เฉพาะด้าน มีความสามารถในการใช้ข้อมูลร่วมกับข้อมูลหลากหลายประเภท สามารถนำข้อมูลกลับมาใช้ใหม่ และการถ่ายทอดคุณสมบัติ [7] จากคุณสมบัติของออนโทโลยีในการใช้ข้อมูลร่วมกัน จึงได้มีการพัฒนาเครื่องมือ เพื่อให้ออนโทโลยีใช้งานร่วมกับข้อมูลที่อยู่ในโครงสร้างอื่น เช่น Mapping ER Schemas to OWL ontologies [8], Ontology-Based Mediated XML Mapping [9], Database-to-Ontology Mapping Generation for Semantic Interoperability [10] เป็นต้น

การแมปฐานข้อมูลกับออนโทโลยี (Database-to-Ontology Mapping) เป็นกระบวนการเชื่อมระหว่างฐานข้อมูลและออนโทโลยี สามารถจำแนกออกเป็น 2 ประเภท คือการสร้างออนโทโลยีใหม่จากฐานข้อมูลเดิม และการแมปฐานข้อมูลเดิมกับออนโทโลยี [10] ซึ่งงานวิจัยนี้ได้เลือกวิธีการแมปฐานข้อมูลกับออนโทโลยี ด้วยวิธีเปรียบเทียบความสอดคล้องระหว่างข้อมูลและออนโทโลยี ซึ่งจะแมปฐานข้อมูลเชิงสัมพันธ์และออนโทโลยี

Haichi Liu et al. [2,11] ได้นำเสนอเครื่องมือชื่อ “MapOnto” พัฒนาขึ้นเพื่อใช้งานร่วมกับ Protégé2.1 นำเข้าได้เฉพาะไฟล์ SQL DDL ที่ใช้สร้างฐานข้อมูล

Igor Myroshnichenko. [3] ได้นำเสนอเครื่องมือชื่อ “D2OMapper” ใช้สำหรับอธิบายองค์ประกอบในเว็บเชิงความหมาย วิธีการคือใช้เทคนิคการจับคู่โครงสร้าง และการจับคู่ประโยคกับความหมายให้ตรงกับการกำหนดองค์ประกอบ แต่ไม่สามารถค้นหาโครงสร้างข้อมูลที่ขาดความสัมพันธ์กันได้

Nadine Cullot et al. [4] พัฒนาเครื่องมือชื่อ “RONTO” เป็นเครื่องมือที่เป็นปลั๊กอินของ Protégé-2.0+ สามารถแมปตารางและคลาสเข้าด้วยกัน โดยไม่คำนึงถึงโครงสร้างส่วนอื่นของฐานข้อมูล

หน่วยปฏิบัติการวิจัยเทคโนโลยีภาษาธรรมชาติและความหมาย ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และ

คอมพิวเตอร์แห่งชาติ (NECTEC) [5,6] ได้พัฒนาเครื่องมือ “Database to Ontology Mapping and Semantic Search Configuration System” (D2OMap) ที่ใช้แมปฐานข้อมูลกับออนโทโลยี สามารถทำงานได้กับ OE5 และระบบจัดการฐานข้อมูล MySQL ไม่สนับสนุนข้อมูลที่ไม่ได้อยู่ในระบบจัดการฐานข้อมูล เช่น CSV, Excel, Text, XML

ซึ่งในแต่ละวิธีการที่กล่าวมา ยังมีข้อจำกัดในการทำงาน งานวิจัยนี้ได้นำเสนอระบบแมปฐานข้อมูล และออนโทโลยีที่สามารถแก้ไขข้อจำกัดบางประการ ของเครื่องมืออื่น โดยเน้นที่จัดการแมปฐานข้อมูลกับ ออนโทโลยี โดยเน้นการพัฒนาต่อจากงานของ NECTEC [5,6] ที่ยังคงมีข้อจำกัดการทำงานดังนี้

- รองรับไฟล์ออนโทโลยี ที่สร้างจาก OE5 เท่านั้น
- ปัจจุบันรองรับเฉพาะระบบจัดการฐานข้อมูล MySQL

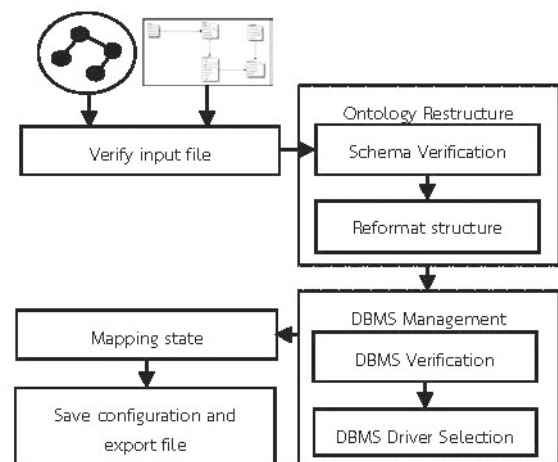
จากข้อจำกัดของโปรแกรมแมปฐานข้อมูลกับออนโทโลยีดังกล่าว ผู้วิจัยจึงได้นำโปรแกรมนี้มาเพื่อทำการพัฒนาต่อยอด เพื่อให้สามารถแก้ไขข้อจำกัดที่มีอยู่

3. ระบบการแมปฐานข้อมูลเป็นออนโทโลยี

การพัฒนาระบบเพื่อให้รองรับข้อมูลจากฐานข้อมูล และออนโทโลยีจากหลากหลายโปรแกรม ในภาพที่ 1 จะแสดงถึงกระบวนการทำงานทั้งหมดเพื่อให้สามารถรองรับฐานข้อมูล และนำเข้าข้อมูลออนโทโลยีของระบบแมปข้อมูลจากฐานข้อมูลเป็นออนโทโลยี ซึ่งประกอบไปด้วย 5 โมดูลคือ Verify input file, Ontology Restructure, DBMS Management, Mapping state และ Save configuration and export file ในงานวิจัยนี้ได้เพิ่มโมดูลเพื่อปรับโครงสร้างออนโทโลยี (Ontology restructure) และโมดูลเพื่อจัดการฐานข้อมูล (DBMS Management)

โมดูลเพื่อปรับโครงสร้างออนโทโลยี (Ontology restructure)

เนื่องจากระบบเดิม (NectecD2OMAP) รองรับการทำงานของความรู้ออนโทโลยีที่สร้างจาก OE5 เท่านั้น ผู้วิจัยได้พัฒนาเพิ่มศักยภาพของระบบ ให้ระบบสามารถรองรับไฟล์ออนโทโลยีที่สร้างจากโปรแกรม Protégé และ Altova ได้ ดังนั้น เพื่อปรับโครงสร้างออนโทโลยี จะตรวจสอบโครงสร้างของความรู้ออนโทโลยีที่นำเข้า กรณีเป็นโครงสร้างของ Protégé หรือ Altova โมดูลนี้จะทำการปรับโครงสร้างให้สอดคล้องกับ OE5 รายละเอียดการเปรียบเทียบโครงสร้างของ OE5, Protégé และ Altova รวมถึงวิธีการจัดการรูปแบบของไฟล์ออนโทโลยี Protégé และ Altova ให้อยู่ในรูปแบบของโปรแกรม OE5



ภาพที่ 1 ภาพรวมของระบบที่นำเสนอ

ในงานวิจัยนี้จะนำเสนอเฉพาะบางส่วนในโครงสร้างออนโทโลยีของ OE5, Protégé และ Altova ที่ต่างกัน ซึ่งประกอบไปด้วย 8 ส่วน คือ

- 1) Header
- 2) Comment ontology
- 3) ObjectProperty



- 4) Class
- 5) sub Class ที่กำหนดเงื่อนไขให้กับคุณสมบัติของคลาส
- 6) sub Class ที่ไม่กำหนดเงื่อนไขให้กับคุณสมบัติของคลาส
- 7) รูปแบบการจัดเรียงลำดับของไฟล์ออนโทโลยี
- 8) ObjectProperty ของโดเมนรายละเอียดแสดงในตารางที่ 1

โมดูลเพื่อจัดการระบบจัดการฐานข้อมูล (DBMS Management)

เนื่องจากระบบเดิมจะทำงานกับระบบจัดการฐานข้อมูลเชิงสัมพันธ์ MySQL เท่านั้น ผู้วิจัยได้พัฒนาเพิ่มให้ระบบสามารถรองรับฐานข้อมูลของระบบจัดการฐานข้อมูล PostgreSQL, MS-Access และ SQL Server

จากการศึกษาพบว่าส่วนที่จัดการเกี่ยวกับระบบจัดการฐานข้อมูลของโปรแกรม NectecD2OMAP คือ dbinteraction-Packages ดังนั้นผู้วิจัยได้ปรับแก้ไข dbinteraction-Packages โดยเพิ่มไดรฟ์เวอร์ของ

ระบบจัดการฐานข้อมูล PostgreSQL, MS-Access และ SQL Server ซึ่งส่วนรายละเอียดของไดรฟ์เวอร์ [12, 13] ตัวอย่างคลาสเนม และรูปแบบ URL ที่ได้นำมาใช้งานบางส่วนจะแสดงดัง ตารางที่ 2

Mapping state [5] เป็นขั้นตอนการกำหนดความสัมพันธ์ระหว่างฐานข้อมูล และฐานความรู้ออนโทโลยีเพื่อใช้ในการแมปข้อมูล (Data Mapping) ซึ่งสามารถแบ่งได้เป็น 3 ส่วนได้แก่

- กำหนดความสัมพันธ์ของคลาสกับตาราง (Class-Table Mapping)
- กำหนดความสัมพันธ์ของคุณสมบัติกับคอลัมน์ (Property Column Mapping)
- กำหนดค่าการแมปคำศัพท์ (Vocabulary Mapping)

Save configuration and export file [5] เป็นกระบวนการบันทึกข้อมูลหลังจากโมดูล Mapping state จากนั้นจะได้ผลลัพธ์ออกมาเป็นไฟล์ที่อยู่ในรูปแบบไฟล์ RDF ซึ่งจะสามารถนำไปใช้งานในด้านเว็บเชิง ความหมายได้

ตารางที่ 1 : รายละเอียดการเปรียบเทียบโครงสร้างของ OE5, Protégé และ Altova

Part	Protégé/Altova	OE5	ความแตกต่างระหว่าง Protégé กับ OE5
Header	<pre><?xml version="1.0"?> <rdf:RDF xmlns="http://www.hozo.jp/owl/WorldCup.owl#" xml:base="http://www.hozo.jp/owl/WorldCup.owl" xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#" xmlns:swrl="http://www.w3.org/2003/11/swrl#" xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#" xmlns:swrlb="http://www.w3.org/2003/11/swrlb#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:owl="http://www.w3.org/2002/07/owl#"></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns="http://www.hozo.jp/owl/WorldCup.owl#" xml:base="http://www.hozo.jp/owl/WorldCup.owl#"></pre>	ส่วนของ OE5 จะมีการตั้งค่าให้รองรับ encoding="UTF-8" และส่วนแสดงการสร้างไฟล์ออนโทโลยีปรับแก้โดยการจัดรูปแบบ header ของ Protégé/Altova ให้อยู่ในรูปแบบไฟล์จากโปรแกรม OE5

Comment ontology	<pre> <owl:Ontologyrdf:about=""> <rdfs:commentrdf:datatype = "&xsd:string"> HOZO:OWL Export </rdfs:comment> </owl:Ontology> </pre>	<pre> <owl:Ontologyrdf:about=""> <rdfs:comment> HOZO:OWL Export </rdfs:comment> </owl:Ontology> </pre>	ส่วน comment ontology ของ Protégé/Altova จะมี "rdf:datatype="&xsd:string"" ปรับแก้โดยการตัดสตริงโดยตัดเฉพาะส่วนของ "rdf:datatype="&xsd:string""
ObjectProperty	<pre> <owl:ObjectPropertyrdf:ID="hasPart"> <rdfs:labelrdf:datatype="&xsd:string">hasPart</rdfs:label> </owl:ObjectProperty> </pre>	<pre> <owl:ObjectPropertyrdf:ID="hasPart"> <rdfs:label>hasPart</rdfs:label> </owl:ObjectProperty> </pre>	Object Property ของ Protégé/Altova จะมี "rdf:datatype="&xsd:string"" ปรับแก้โดยการตัดสตริงโดยตัดเฉพาะส่วนของ "rdf:datatype="&xsd:string""
Class	<pre> <owl:Classrdf:ID="RelationalConcept"> <rdfs:labelrdf:datatype="&xsd:string"> RelationalConcept</rdfs:label> </owl:Class> </pre>	<pre> <owl:Classrdf:ID="RelationalConcept"> <rdfs:label>RelationalConcept</rdfs:label> </owl:Class> </pre>	Class ของ Protégé/Altova จะมี "rdf:datatype="&xsd:string"" ปรับแก้โดยการตัดสตริงโดยตัดเฉพาะส่วนของ "rdf:datatype="&xsd:string""
subClass ที่กำหนดเงื่อนไขให้กับ คุณสมบัติของคลาส	<pre> <owl:Classrdf:ID="Player"> <rdfs:subClassOfrdf:resource="#Any"/> <rdfs:subClassOf> <owl:Restriction> <owl:onPropertyrdf:resource="has_name"/> <owl:cardinalityrdf:datatype="&xsd:nonNegativeInteger">1</owl:cardinality> <owl:Restriction> </rdfs:subClassOf> </pre>	<pre> <owl:Classrdf:ID="Player"> <rdfs:label>Player</rdfs:label> <rdfs:subClassOfrdf:resource="#Any" /> <rdfs:subClassOf> <owl:Restriction> <owl:cardinalityrdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:cardinality> <owl:onPropertyrdf:resource="has_name" /> <owl:Restriction> </rdfs:subClassOf> </pre>	Subclass ของ Protégé/Altova ที่ owl:cardinalityrdf:datatype จะแตกต่างจากของ OE5 ปรับแก้โดยการแทรกสตริง "http://www.w3.org/2001/XMLSchema#nonNegativeInteger"" ลงใน <owl:cardinalityrdf:datatype=""String"" >เข้าไปแทน rdf:datatype เดิมและจัดเรียง owl:cardinality และ owl:cardinality ให้อยู่ในรูปแบบไฟล์จากโปรแกรม OE5
subClass ที่ไม่กำหนดเงื่อนไข ให้กับคุณสมบัติของ คลาส	<pre> <owl:Classrdf:ID="forward"> <rdfs:subClassOfrdf:resource="#Position"/> <rdfs:labelrdf:datatype="&xsd:string">forward</rdfs:label> </owl:Class> </pre>	<pre> <owl:Classrdf:ID="forward"> <rdfs:label>forward</rdfs:label> <rdfs:subClassOfrdf:resource="#Position" /> </owl:Class> </pre>	Subclass ของ Protégé/Altova จะมี "rdf:datatype="&xsd:string"" ปรับแก้โดยการตัดสตริงโดยตัดเฉพาะส่วนของ "rdf:datatype="&xsd:string"" และจัดเรียง rdfs:subClassOf และ rdfs:label ให้อยู่ในรูปแบบไฟล์จากโปรแกรม OE5
รูปแบบ การจัดเรียงลำดับของ ไฟล์ออนโทโลยี	<pre> <owl:Classrdf:ID="african"> <rdfs:subClassOfrdf:resource="#Country"/> <rdfs:labelrdf:datatype="&xsd:string">african</rdfs:label> </owl:Class> <owl:Classrdf:ID="american"> <rdfs:subClassOfrdf:resource="#Country"/> <rdfs:labelrdf:datatype="&xsd:string">american</rdfs:label> </owl:Class> <owl:Classrdf:ID="Any"> <rdfs:labelrdf:datatype="&xsd:string">Any</rdfs:label> </owl:Class> </pre>	<pre> <owl:Classrdf:ID="RelationalConcept"> <rdfs:label>RelationalConcept</rdfs:label> </owl:Class> <owl:ObjectPropertyrdf:ID="hasPart"> <rdfs:label>hasPart</rdfs:label> </owl:ObjectProperty> <owl:ObjectPropertyrdf:ID="hasAttribute"> <rdfs:label>hasAttribute</rdfs:label> </owl:ObjectProperty> <owl:Classrdf:ID="Any"> <rdfs:label>Any</rdfs:label> </owl:Class> </pre>	การจัดเรียงลำดับของ Class ในออนโทโลยี ที่สร้างจากโปรแกรม Protégé/Altova การเรียงลำดับก่อนหลัง จะพิจารณาจาก ชื่อของคลาสและชื่อของคุณสมบัติโดยเรียงจาก A-Z ดังตัวอย่าง ปรับแก้โดยการ จัดเรียงให้อยู่ในรูปแบบ ไฟล์จากโปรแกรม OE5 ซึ่งออนโทโลยีที่สร้างจากโปรแกรม OE5 จะพิจารณาจากลำดับความสำคัญของคลาสและ จะจัด subClassให้อยู่ติดกับ คลาสแม่ของตัวมันเองจากตัวอย่าง
ObjectProperty ของโดเมน	<pre> <owl:ObjectPropertyrdf:ID="has_label"> <rdfs:domain> <owl:Class> <owl:unionOfrdf:parseType="Collection"> <owl:Classrdf:about="#Country"/> <owl:Classrdf:about="#Football_club"/> <owl:Classrdf:about="#Position"/> </owl:unionOf> </owl:Class> </rdfs:domain> <rdfs:subPropertyOfrdf:resource="hasAttribute"/> </owl:ObjectProperty> </pre>	<pre> <owl:ObjectPropertyrdf:ID="has_label"> <rdfs:subPropertyOfrdf:resource="hasAttribute" /> <rdfs:domain> <owl:Class> <owl:unionOfrdf:parseType="Collection"> <owl:Classrdf:about="#Country" /> <owl:Classrdf:about="#Position" /> <owl:Classrdf:about="#Football_club" /> </owl:unionOf> </owl:Class> </rdfs:domain> </owl:ObjectProperty> </pre>	ObjectPropertyของโดเมน ใน Protégé/Altova ส่วนแสดงคุณสมบัติย่อยของคลาส จะแสดงอยู่ที่ปิดโดเมน ปรับแก้โดยการสลับ ตำแหน่งให้เปลี่ยนไปอยู่ก่อนการเปิดโดเมน ซึ่งอยู่ในรูปแบบไฟล์จากโปรแกรม OE5



**ตารางที่ 2 : ไดรเวอร์รองรับระบบจัดการฐานข้อมูล
สำหรับการใช้งานบนเว็บแอปพลิเคชัน**

DBMS	Driver JAR File	Driver ClassName	URL Format
PostgreSQL	postgresql-8.2-506.jdbc3.jar	org.postgresql.Driver	jdbc:postgresql://<host>:<port>/<database>
MySQL	mysql-connector-java-5.0.7-bin.jar	com.mysql.jdbc.Driver	-
MS-Access	-	sun.jdbc.odbc.JdbcOdbcDriver	jdbc:odbc:Driver={Microsoft Access Driver (*.mdb)};DBQ=TLDATABASES\BM.mdb
SQL Server	sqljdbc.jar	com.microsoft.sqlserver.jdbc.SQLServerDriver	jdbc:sqlserver://localhost:1433;data baseName=travel;selectMethod=cursor

ตารางที่ 3 : ชุดข้อมูลที่ใช้ทำการทดสอบ

Data Set	#Table	#Column	#Class	#Property
University	8	32	53	35
Conference	9	38	18	29
DBLP	5	27	67	81
OBSERVER	7	91	67	81

4. การประเมินผล

ลักษณะสำคัญของงานวิจัยนี้ มุ่งเน้นการปรับระบบให้สามารถรองรับ OWL และรองรับระบบจัดการฐานข้อมูลที่หลากหลาย ดังนั้นเพื่อประเมินความสามารถของระบบ ผู้วิจัยจึงได้ทำเปรียบเทียบการทำงานของระบบที่นำเสนอ (ExtendNectecD2OMAP) กับระบบ NectecD2OMAP และเปรียบเทียบกับระบบที่นำเสนอในงานวิจัยของ Jerry [11] ซึ่งในงานวิจัยของ Jerry ได้ทำการเปรียบเทียบ เครื่องมือดังต่อไปนี้คือ MapOnto และ RONTO เพื่อให้เป็นมาตรฐานเดียวกัน ผู้วิจัยจึงได้ใช้ข้อมูลของ Jerry ในการทดสอบ โดยดาวน์โหลดจาก “<http://www.cs.toronto.edu/~yuana/research/maponto/relational/testData.html>” [11] ชุดข้อมูล

ประกอบไปด้วยฐานข้อมูลเชิงสัมพันธ์ และไฟล์ฐานความรู้ออนโทโลยี โดยรายละเอียดแสดงในตารางที่ 3 ผลการเปรียบเทียบ ความสามารถแสดงในตารางที่ 4

5. บทสรุป และข้อเสนอแนะ

ผลการเปรียบเทียบความสามารถการทำงานของเครื่องมือ MapOnto, RONTO, NectecD2OMAP, ExtendNectecD2OMAP ในตารางที่ 4 พบว่า MapOnto และ RONTO เป็นปลั๊กอิน ซึ่งมีข้อจำกัดคือสามารถใช้งานได้เฉพาะกับโปรแกรม Protégé และรองรับการทำงานกับไฟล์ความรู้ออนโทโลยีของโปรแกรม Protégé ส่วนระบบ NectecD2OMAP เป็นเว็บแอปพลิเคชัน รองรับไฟล์ความรู้ออนโทโลยีเฉพาะ OE5 และระบบจัดการฐานข้อมูลเชิงสัมพันธ์ MySQL ExtendNectecD2OMAP สามารถรองรับไฟล์ความรู้ออนโทโลยีจากโปรแกรม Protégé 3.1+ และ Altova และรองรับระบบจัดการฐานข้อมูลเชิงสัมพันธ์ PostgreSQL, MySQL, MS-Access และ SQL Server ซึ่งพบว่า ExtendNectecD2OMAP มีความสามารถเพิ่มขึ้นจาก NectecD2OMAP

แนวทางในการพัฒนาต่อ จะดำเนินการให้ระบบแมปข้อมูลจากฐานข้อมูลเป็นออนโทโลยีรองรับระบบจัดการฐานข้อมูลอื่นๆ เช่น SQL Server และ Oracle และจะปรับการทำงานของ โมดูลของการแมปให้สามารถรองรับความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many to Many, M:N) ที่อาจเกิดขึ้นในฐานข้อมูล และปรับแก้ให้ระบบรองรับข้อมูลที่ถูกรวบรวมอยู่ในรูปแบบค่าที่เป็นไปได้มากกว่า 1 ค่า (Multi values)

6. กิตติกรรมประกาศ

คณะผู้วิจัยขอขอบคุณดร.มารุต บุรณรัช จากหน่วยปฏิบัติการวิจัยวิทยาการมนุษยาศาสตร์ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ ที่ให้การสนับสนุนข้อมูลการวิจัยในครั้งนี้

ตารางที่ 4 : ข้อแตกต่างของระบบแม่ข่ายข้อมูลจากฐานข้อมูลและออนโทโลยี

Tool name	Tool type			Formats file input						Relational support	Export File
	Plugin	Application	Web App.	MySQL	SQL Server	MS-Access	PostgreSQL	OWL	RDF		
MapOnto (MARSON, HyMRO)	✓				✓			Protégé 2.1		no support Ternary	
RONTO	✓			✓	✓	✓		Protégé-2.0+		no support n-ary database relationships and Ternary Relationships	
Nectec D2OMAP			✓	✓				OE5		Vocabulary Mapping no support Many-to-many	Standard RDF
Extend Nectec D2OMAP			✓	✓	✓	✓	✓	OE5, Protégé3.1+ Altova		Vocabulary Mapping no support Many-to-many	Standard RDF

เอกสารอ้างอิง

[1] Goutam K.S, “Web Ontology Language (OWL) and Semantic Web”, ACM Magazine Ubiquity, 8(1), pp. 1-24, 2007.

[2] Wei H. and Yuzhong O. “Discovering Simple Mappings Between Relational Database Schemas and Ontologies”, Lecture Notes in Computer Science, vol. 4825, pp 225-238, 2007.

[3] Yuzhong Q. et al., “Mapping Between Relational Database Schemas and Ontologies: The State of The Art [J]”, Journal of Computer Research and Development, vol. 2, 2008.

[4] Katsioulis P., et al., “Matching Relational Schemata to Semantic Web Ontologies”, Handbook of Research on Innovations in Database Technologies and Applications: Current and Future Trends, Information science reference volume II, Hershey, New York, 2009.

[5] หน่วยปฏิบัติการวิจัยเทคโนโลยีภาษารวมชาติและความหมาย ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ, “Tutorial on Database to Ontology Mapping and Semantic Search Configuration System”, Retrieved March 5, 2012, from <http://text.hlt.nectec.or.th/ontology>, 2552.

[6] D2RQ. Retrieved March 5, 2012, from <http://d2rq.org/>

[7] Geroimenko V. and Geroimenko L. “Visual Interaction with XML Metadata”, Proceedings of the Fifth International Conference on Information Visualisation, DC, USA, pp. 539-545, 2001.

[8] Myroshnichenko I. and Marguerite C. Murphy M. “Mapping ER Schemas to OWL Ontologies”, Third IEEE International Conference on Semantic Computing Berkeley, September 14-16, 2009, CA, USA, pp. 324-329, 2009.

[9] วิสิทธิ์ บุญชุม และนวลวรรณ สุนทรภักษ์,



“Ontology-Based Mediated XML Mapping,ออนโทโลยี: สื่อกกลางในการจับคู่ข้อมูลเอ็กซ์เอ็มแอลที่มีความสัมพันธ์กันเชิงความหมาย”, The 5th National Conference on Computing and Information Technology, May 22-23, 2009, Bangkok, 2009.

[10] Ahmed, W., AhtishamAslam, M., Shen, J. and Yong, J. “Database-to-Ontology Mapping Generation for Semantic Interoperability”, Proceedings of the 33rd International Conference on Very Large Databases, September 23 - 28, 2007, University of Vienna, Austria,2007.

[11] Yuan An J., “Sample Relational Schemas/Ontologies”, Retrieved Feb. 27, 2013, [online] from <http://www.cs.toronto.edu/~yuana/research/maponto/relational/testData.html>

[12] Netbeans, Retrieved March 5, 2012, [online] from <http://wiki.netbeans.org/DatabasesAndDrivers>

[13] Databases-Netbeans, Retrieved March 5, 2012, [online] from <http://www.coderanch.com/t/306729/JDBC/databases/Netbeans-MS-Access>