

หมายเหตุ: กฎในข้อ 3 ของนิยาม 5.1 และกฎในข้อ 2 ของนิยาม 5.2 เป็นกฎที่ตั้งอยู่บนพื้นฐานของโมเดล RDF แต่จากคุณสมบัติของ OWL จะยังสามารถเพิ่มเติมคุณลักษณะอื่นๆได้อีก เช่น การกำหนดกลุ่มอินสแตนซ์จากคลาสหนึ่งที่จะมีความสัมพันธ์กับกลุ่มอินสแตนซ์ของอีกคลาสหนึ่ง หรือการกำหนดจำนวนอินสแตนซ์ที่น้อยที่สุดหรือมากที่สุดของคลาสที่จะมีความสัมพันธ์กัน เป็นต้น ดังที่กล่าวไว้ในนิยามที่ 4.1

4.2 โครงสร้างของเมตาดาต้าสำหรับตัวบริการเว็บบนพื้นฐานของออนโทโลยี

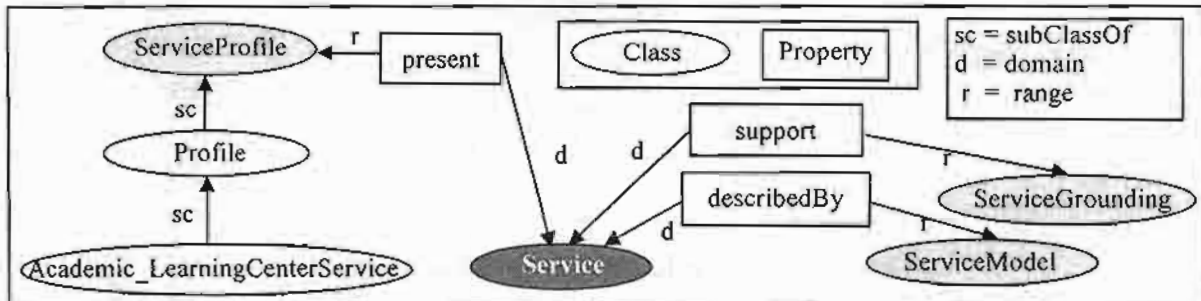
ในหัวข้อนี้จะนำเสนอโครงสร้างของเมตาดาต้าสำหรับตัวบริการเว็บบนพื้นฐานของออนโทโลยี ที่สอดคล้องกับองค์ประกอบพื้นฐานที่ได้กล่าวไว้ในหัวข้อข้างต้น โครงสร้างเมตาดาต้าสำหรับตัวบริการเว็บบนพื้นฐานของออนโทโลยีนี้จะถูกโมเดลบนพื้นฐานของการออกแบบจากล่างขึ้นบน (bottom-up design approach) [34] โดยมีขั้นตอนเริ่มจากผู้ให้บริการทำการประกาศบริการของตนกับระบบ LIS โดยการระบุตำแหน่งเอกสาร WSDL แล้ว ในกระบวนการสร้างเมตาดาต้าสำหรับตัวบริการเว็บจะประกอบด้วยกระบวนการแปลงข้อมูลบริการที่อธิบายอยู่ในเอกสาร WSDL ให้เป็นออนโทโลยีเริ่มต้น ที่อยู่ในรูปแบบของภาษา OWL-S โดยผ่าน OWL API [32] ออนโทโลยีเริ่มต้นนี้จะประกอบด้วยโครงสร้างที่สำคัญสามส่วนด้วยกันคือ ส่วนที่เรียกว่า service profile, service model, และ service grounding หลังจากนั้นก็จะมีการเพิ่มเติมข้อมูลอื่นๆเข้าไปอีกในออนโทโลยีเริ่มต้นนี้ เช่น ข้อมูลรายละเอียดของผู้ให้บริการประเภทและกลุ่มของข้อมูลบริการ เป็นต้น เพื่อสร้างเป็นเมตาดาต้าบนพื้นฐานของออนโทโลยีในระดับของข้อมูลบริการเพื่อสนับสนุนในด้านการสืบค้นและเรียกใช้งานตัวบริการเว็บให้สามารถทำได้โดยอัตโนมัติ ในขั้นตอนถัดไปคือการสร้างออนโทโลยีในระดับผู้ใช้ที่เรียกว่า ออนโทโลยีสามัญ (common ontology) ซึ่งเป็นออนโทโลยีที่มีโครงสร้างบนพื้นฐานของโมเดล LOM เพื่อให้ผู้ใช้ทั่วไปสามารถรู้จักและเรียกใช้งานเทอมต่างๆที่อยู่ในออนโทโลยีระดับนี้เพื่อการสืบค้นข้อมูลการเรียนรู้ที่ต้องการ และสำหรับในขั้นตอนสุดท้าย คือการเชื่อมโยงออนโทโลยีในระดับข้อมูลบริการให้เข้ากับออนโทโลยีในระดับผู้ใช้ เพื่อสร้างเป็นเมตาดาต้าสำหรับตัวบริการเว็บที่สมบูรณ์ ซึ่งในขั้นตอนสุดท้ายนี้ จำเป็นต้องสร้างเครื่องมือที่ช่วยในการเชื่อมโยง ซึ่งได้แก่ ตัวเชื่อมโยงข้อมูลบริการ ที่จะทำการเชื่อมโยงข้อมูลบริการในออนโทโลยีระดับข้อมูลบริการ เช่น เมชอด หรือพารามิเตอร์ ของตัวบริการเว็บที่มีความสอดคล้องกันจากผู้ให้บริการที่หลากหลาย ให้เข้ากับเทอมที่อยู่ในออนโทโลยีระดับผู้ใช้ เพื่อให้การสืบค้นข้อมูลจากผู้ให้บริการทำได้ในมุมมองของผู้ใช้ โดยไม่ต้องยุ่งเกี่ยวหรือทราบเกี่ยวกับข้อมูลบริการที่ถูกเก็บอยู่ในออนโทโลยีระดับบริการแต่อย่างใด

ดังนั้น โครงสร้างเมตาดาต้าสำหรับตัวบริการเว็บบนพื้นฐานของออนโทโลยีจะสามารถแบ่งออกได้เป็นสองระดับได้แก่ ระดับออนโทโลยีสำหรับข้อมูลบริการ (Service ontology level) และระดับออนโทโลยีสำหรับผู้ใช้ทั่วไป (Common ontology level) โดยแต่ละระดับจะมีโครงสร้างที่อธิบายได้ดังต่อไปนี้

4.2.1 ระดับออนโทโลยีสำหรับข้อมูลบริการ

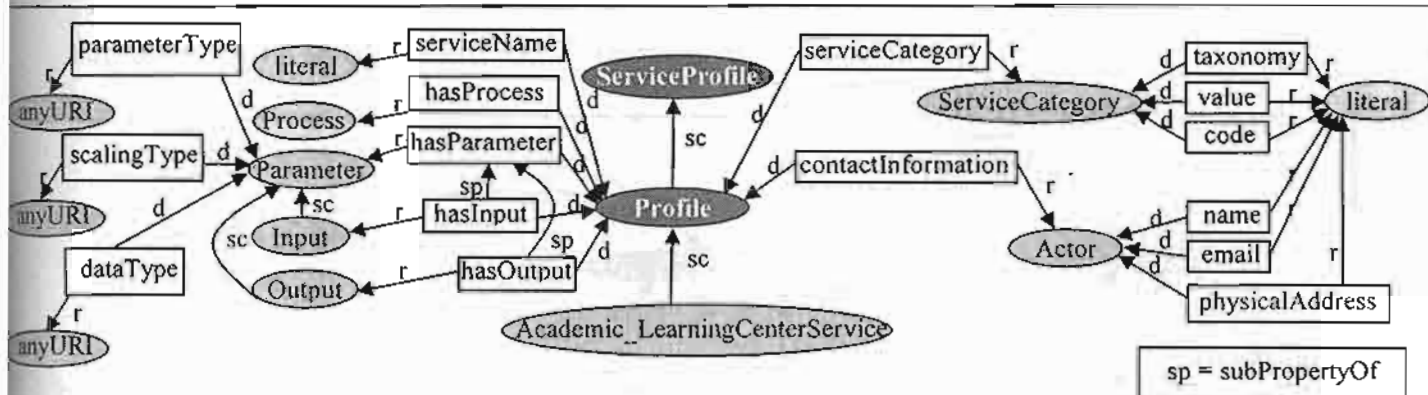
เป็นระดับออนโทโลยีที่ประกอบด้วยคำอธิบายข้อมูลบริการในสามส่วน ได้แก่ Service Profile, Service Model และ Service Grounding ดังแสดงในภาพที่ 4.1 จะทำหน้าที่โดยในภาพนี้จะสัญลักษณ์ วงรี จะหมายถึงคลาสหรือคอนเซป (class หรือ concept) ในขณะที่สัญลักษณ์รูปสี่เหลี่ยมผืนผ้าจะหมายถึง คุณสมบัติ (properties หรือ predicates) แต่ละคุณสมบัติ เช่น present, support และ describedBy จะทำหน้าที่เชื่อมความสัมพันธ์ระหว่างโดเมนที่ถูกเรียกว่าเป็น Subject ของคุณสมบัติ ไปยังเรนจ์ที่ถูกเรียกว่าเป็น ค่าข้อมูลของคุณสมบัติ ตัวอย่างเช่น (Service, present, ServiceProfile), (Service, describedBy, ServiceModel) และ (Service, support, ServiceGrounding) เป็นต้น แต่ละอินสแตนซ์ของคลาส Service ที่ถูกสร้างเมื่อต้องการประกาศบริการของตน จะมีการสืบทอดคุณสมบัติทุกตัวจากคลาส Service ด้วย เพื่อที่ว่าอินสแตนซ์ของคลาส Service จะมีคุณสมบัติ present เพื่อแสดงอินสแตนซ์จากคลาส

ServiceProfile คลาสแต่ละคลาสสามารถเป็น subclass ของคลาสอื่นได้ เช่น (Profile, subClassOf, ServiceProfile) สำหรับโดเมน e-Learning จะแสดงคลาส Academic_LearningCenterService เป็น subclass ของคลาส Profile ที่ประกอบด้วย Profile อินสแตนส์จากผู้ให้บริการแต่ละรายที่หลากหลาย สำหรับงานวิจัยนี้จะแสดงโครงสร้าง ServiceProfile ดังต่อไปนี้



ภาพที่ 4.1 โครงสร้างออนโทโลยีในระดับบริการ

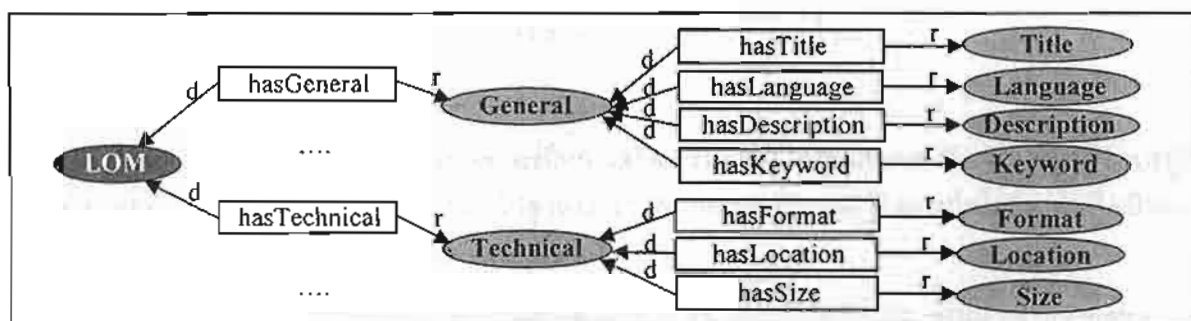
Service Profile จะมีหน้าที่อธิบายข้อมูลบริการของตัวบริการเว็บแต่ละตัวที่ผู้ให้บริการมาประกาศไว้ เพื่ออำนวยความสะดวกในการสืบค้นตัวบริการเว็บอย่างอัตโนมัติ องค์ประกอบพื้นฐานของ Service Profile จะได้แก่ คำอธิบายข้อมูลของผู้ให้บริการ (provider information) คำอธิบายหน้าที่ของบริการ (functional description of service) และคำอธิบายคุณสมบัติของบริการ (description of a host of properties) สำหรับคำอธิบายข้อมูลบริการจะประกอบด้วยคุณสมบัติชื่อผู้ให้บริการ (name), อีเมล (email), และ ที่อยู่ผู้ให้บริการ (physical address) ซึ่งได้ถูกอธิบายผ่านทางคุณสมบัติ contactInformation คำอธิบายหน้าที่ของบริการจะถูกอธิบายผ่านชื่อเมธอดหรือโพรเซส ชื่อพารามิเตอร์เข้าและออก เป็นต้น โดยคลาส Input และ Output จะถูกออกแบบให้เป็น subclass ของคลาส Parameter นอกจากนี้คุณสมบัติ hasInput และ hasOutput จะถูกออกแบบให้เป็น sub-properties ของคุณสมบัติ hasParameter สำหรับการออกแบบเมตาดาต้าสำหรับตัวบริการเว็บในงานวิจัยนี้จะมีการเพิ่มเติมคุณสมบัติพื้นฐานของ service profile โดยการเสนอคุณสมบัติเพิ่มเติมสามคุณสมบัติได้แก่ dataType, scalingType และ parameterType เพื่อใช้แก้ปัญหาความขัดแย้งกันของประเภทข้อมูล หน่วยวัดข้อมูล และชื่อพารามิเตอร์ ตามลำดับ โดยคุณสมบัติ dataType จะเป็นคุณสมบัติที่ใช้เชื่อมคลาส Parameter เข้ากับคลาสที่เป็นประเภทข้อมูล เช่น String, Integer, Decimal ซึ่งถูกนิยามโดยผ่านทาง anyURI สำหรับคุณสมบัติ scalingType จะใช้เชื่อมคลาส Parameter เข้ากับคลาสที่เป็นหน่วยวัดข้อมูลใดๆ เช่น USD, Euro, Byte, Kilobyte เป็นต้น และสำหรับคุณสมบัติ parameterType จะเชื่อมคลาส Parameter เข้ากับคลาสของ LOM ที่มีการประกาศอยู่ในออนโทโลยีสามัญ โดยคุณสมบัติ dataType และ scalingType นี้จะช่วยให้ข้อมูลที่คล้ายคลึงกันที่ถูกดึงจากตัวบริการเว็บที่หลากหลายที่อาจมีประเภทข้อมูล และ หน่วยวัดข้อมูล ที่แตกต่างกัน สามารถมีประเภทข้อมูล และ หน่วยวัดข้อมูล ที่เหมือนกันตามที่ระบุอยู่ในคุณสมบัติ dataType, scalingType ตามลำดับ แต่สำหรับคุณสมบัติ parameterType จะใช้ในการแก้ปัญหาความขัดแย้งกันของชื่อพารามิเตอร์ของผู้ให้บริการที่แตกต่างกัน ดังที่จะอธิบายอยู่ในหัวข้อถัดไป นอกจากนี้คำอธิบายคุณสมบัติของบริการจะใช้ในการอธิบายรูปแบบของบริการ เช่น ประเภทของบริการ และกลุ่มของบริการ ซึ่งจะถูกอธิบายโดยผ่านทางคุณสมบัติ serviceCategory ที่จะระบุว่าบริการนั้นถูกจัดกลุ่มตาม taxonomy หรือตามมาตรฐานใด (NAICS หรือ UNSPSC) และมีชื่อกลุ่มเป็นอะไร (categoryName) เป็นต้น ส่วนของโครงสร้าง Service Profile จะอธิบายอยู่ในภาพที่ 4.2



ภาพที่ 4.2 แสดงบางส่วนของโครงสร้าง Service Profile

4.2.2 ระดับออนโทโลยีสำหรับผู้ทั่วไป

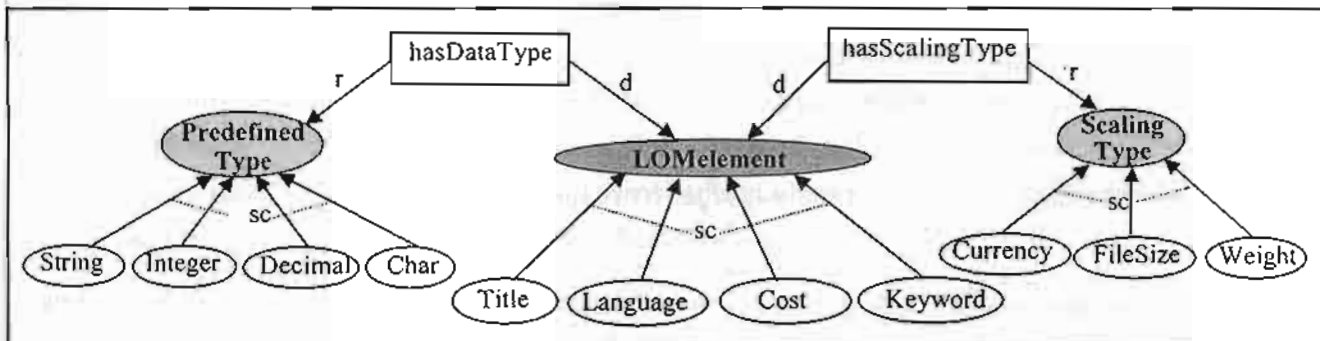
ออนโทโลยีสำหรับผู้ทั่วไปนี้จะช่วยให้ผู้ทั่วไปสามารถใช้เป็นข้อมูลในการสืบค้นข้อมูลการเรียนรู้จากระบบ LIS ได้ โดยออนโทโลยีในระดับนี้จะสร้างมาจากโครงสร้างของ LOM ที่ผู้ทั่วไปจะรู้จักและคุ้นเคย โดยอิลิเมนต์ต่างๆของ LOM นี้จะใช้อธิบายคุณสมบัติต่างๆของออบเจกต์การเรียนรู้ (learning object) ออนโทโลยีในระดับนี้จะจัดกลุ่มอิลิเมนต์ต่างๆของ LOM ที่มีอยู่เข้ากลุ่มให้เป็นคลาสต่างๆ เช่น General, LifeCycle, Technical ดังแสดงในภาพที่ 4.3 โดยมีการสร้างคุณสมบัติให้กับแต่ละคลาส เช่น *hasTitle*, *hasLanguage*, *hasDescription* เพื่อใช้เก็บค่าของคุณสมบัติต่างๆที่อยู่ในคลาส เช่น Title, Language, Description ซึ่งคลาสเหล่านี้จะถูกนิยามให้เป็น subclass ของคลาส LOMelement ซึ่งโครงสร้างออนโทโลยีในระดับผู้ใช้นี้จะช่วยให้ผู้ใช้ไม่ต้องทราบรายละเอียดของการเรียกใช้ตัวบริการเว็บแต่อย่างใด



ภาพที่ 4.3 แสดงบางส่วนของออนโทโลยีสามัญที่สอดคล้องกับโครงสร้าง LOM

การออกแบบออนโทโลยีสามัญในระดับผู้ใช้นั้นนอกจากจะช่วยให้ผู้ทั่วไปมีมุมมองของข้อมูลที่อยู่ในรูปแบบของ LOM ที่เหมือนกันแล้ว ยังจะช่วยในการแก้ปัญหาความขัดแย้งกันของประเภทข้อมูลและหน่วยวัดข้อมูลอีกด้วย ซึ่งจะมีการเพิ่มสองคุณสมบัติเข้าไปคือ *hasDataType* และ *hasScalingType* โดยคุณสมบัติ *hasDataType* จะถูกออกแบบมาเพื่อเชื่อมโยงคลาส LOMelement เข้ากับคลาสที่เป็น predefined data types เช่น Integer, String, Decimal หรือ Char เป็นต้น ในขณะที่ *hasScalingType* จะถูกออกแบบมาเพื่อใช้เชื่อมโยงคลาส LOMelement ให้เข้ากับคลาสที่เป็นหน่วยวัด เช่น currency, fileSize, Weight เป็นต้น ดังแสดงในภาพที่ 4.4 คุณสมบัติทั้งสองนี้จะถูกถ่ายทอดจากคลาส LOMelement ไปยังสับคลาสของคลาส LOMelement ด้วย ซึ่งจะทำให้ข้อมูลผลลัพธ์ที่ถูกดึงจากตัว

บริการเว็บที่หลากหลายที่อาจมีประเภทข้อมูลและหน่วยวัดข้อมูลที่แตกต่างกัน ก็จะถูกแปลงให้มีประเภทข้อมูลและหน่วยวัดข้อมูลที่เหมือนกันตามที่นิยามไว้ในออนโทโลยีระดับผู้ใช้



ภาพที่ 4.4 แสดงการแก้ปัญหาความขัดแย้งกันของประเภทข้อมูลและหน่วยวัดข้อมูล

4.3 การเชื่อมโยงออนโทโลยีในระดับบริการกับออนโทโลยีในระดับผู้ใช้ทั่วไป

ในหัวข้อนี้จะแสดงการเชื่อมโยงออนโทโลยีที่ถูกออกแบบในระดับบริการให้เข้ากับออนโทโลยีในระดับผู้ใช้ทั่วไป เพื่อให้ผู้ใช้ทั่วไปไม่ต้องยุ่งเกี่ยวหรือทราบเกี่ยวกับข้อมูลบริการ นอกจากนี้ยังช่วยในการแก้ปัญหาความขัดแย้งกันในเชิงความหมายของข้อมูลบริการได้อีกด้วย เนื่องจากความจริงที่ว่าผู้ให้บริการแต่ละรายอาจประกาศข้อมูลบริการในระบบ LIS ที่เหมือนกันแต่มีการใช้ชื่อเมธอด หรือพารามิเตอร์ที่แตกต่างกัน จึงทำให้เกิดปัญหาความขัดแย้งกันของข้อมูลบริการ ตัวอย่างเช่น ผู้ให้บริการแต่ละคนที่ให้บริการแบบเดียวกัน แต่อาจมีการประกาศชื่อเมธอด ชื่อพารามิเตอร์ทั้ง input และ output ที่แตกต่างกัน

กระบวนการของการเชื่อมโยงออนโทโลยีจะแบ่งออกได้เป็นสองขั้นตอน ได้แก่ การเชื่อมโยงออนโทโลยีในระดับของคลาส และการเชื่อมโยงออนโทโลยีในระดับของอินสแตนซ์ ดังรายละเอียดต่อไปนี้

4.3.1 การเชื่อมโยงออนโทโลยีในระดับของคลาส

การเชื่อมโยงออนโทโลยีในระดับของคลาส จะเป็นการสร้างความเชื่อมโยงของคลาส Parameter ที่นิยามอยู่ใน Service Profile ซึ่งเป็นออนโทโลยีระดับบริการ ไปสู่คลาส LOMelement ที่นิยามอยู่ในออนโทโลยีสามัญซึ่งเป็นออนโทโลยีในระดับผู้ใช้ทั่วไป ซึ่งจะเป็นไปตามนิยามดังต่อไปนี้

นิยามที่ 6 ถ้าให้ SO แทนออนโทโลยี Service Profile และให้ CO แทนออนโทโลยีสามัญ ดังนั้น คลาส Parameter ที่นิยามอยู่บน SO จะถูกเชื่อมโยงกับคลาส LOMelement บนออนโทโลยีสามัญโดยผ่านทางคุณสมบัติ parameterType โดยที่ว่าคุณสมบัติ parameterType จะถูกใช้ในการเชื่อมโยงคลาส Parameter ที่เป็นโดเมน เข้ากับคลาส LOMelement ที่เป็นเรนจ์ ที่เขียนได้ในรูปแบบของทริเพิล (Parameter, parameterType, LOMelement) ทั้งนี้ จากกฎข้อที่ 3 ของนิยามที่ 5.1 จะได้ว่า สำหรับคลาสใดๆ $c_k \in OC$ ถ้า $(c_k, subClassOf, Parameter)$ ดังนั้น $(c_k, parameterType, LOMelement)$ ตัวอย่างเช่น คลาส Input และคลาส Output เป็นสับคลาสของคลาส Parameter ดังนั้น คลาส Input และ Output จะสืบทอดคุณสมบัติ parameterType จากคลาส Parameter ด้วยดังนี้ (Input, parameterType, LOMelement) $\wedge$ (Output, parameterType, LOMelement)

4.3.2 การเชื่อมโยงออนโทโลยีในระดับอินสแตนส์

การเชื่อมโยงออนโทโลยีในระดับอินสแตนส์จะเป็นการเชื่อมโยงพารามิเตอร์อินสแตนส์ต่างๆของคลาส Input และคลาส Output ของแต่ละ Service Profile ให้เข้ากับสับคลาสของ LOMelement โดยอินสแตนส์ต่างๆของคลาส Input และ Output นี้จะสืบทอดคุณสมบัติ parameterType จากคลาส Parameter ในการนี้จำเป็นต้องอาศัยตัวเชื่อมโยงข้อมูลบริการจัดหาเครื่องมือที่อยู่ในรูปแบบของอินเตอร์เฟส เพื่อช่วยให้ผู้ให้บริการสามารถเลือกการเชื่อมโยงพารามิเตอร์ดังกล่าว ตัวเชื่อมโยงข้อมูลบริการนี้จะเปรียบเสมือนกับเป็นกลไกในการแก้ปัญหาความขัดแย้งกันของข้อมูลบริการจากผู้ให้บริการที่หลากหลายให้เป็นข้อมูลบริการที่อยู่ในมุมมองของ LOM ซึ่งเป็นออนโทโลยีที่ผู้ใช้มองเห็น สามารถนิยามการเชื่อมโยงออนโทโลยีในระดับอินสแตนส์ได้ดังนิยามที่ 7 ต่อไปนี้

นิยามที่ 7 กำหนดให้ $c_a \in OC$ เป็นสับคลาสของคลาส Parameter และ $o_c \in OI$ เป็นอินสแตนส์ของ c_a ซึ่งถูกประกาศเป็นพารามิเตอร์ที่ผู้ให้บริการทำการโฆษณาไว้กับระบบ LIS กำหนดให้ $c_b \in OC$ เป็นคลาส LOM element ที่ถือว่าเป็นพารามิเตอร์ที่ผู้ใช้ทั่วไปเรียกใช้ ถ้าพารามิเตอร์ใดใน Service Profile ออนโทโลยีมีความสอดคล้องกับคลาส LOM element พารามิเตอร์นั้นจะถูกเชื่อมโยงกับคลาส LOM element โดยผ่านทางคุณสมบัติ parameterType ดังต่อไปนี้

กฎที่ได้: ถ้า $o_c \sim c_b \longrightarrow (o_c, parameterType, c_b)$. โดยสัญลักษณ์ $\sim$ จะหมายถึงความคล้ายคลึงกัน

ตัวอย่างของการเชื่อมโยงกันของออนโทโลยีในแต่ละระดับจะถูกแสดงอยู่ในกรณีศึกษาของบทถัดไปนี้

บทที่ 5

กรณีศึกษา

ในบทนี้จะแสดงตัวอย่างความขัดแย้งกันในเชิงความหมายของข้อมูลบริการที่ผู้ให้บริการที่หลากหลายมาทำการประกาศข้อมูลบริการของตนไว้ในระบบ LIS เช่น การให้บริการที่เหมือนกันแต่มีการใช้ชื่อเมธอด และชื่อพารามิเตอร์ที่แตกต่างกัน เป็นต้น และจะแสดงตัวอย่างการสร้างเมตาดาต้าสำหรับตัวบริการเว็บบนพื้นฐานของออนโทโลยี ที่แบ่งออกเป็นสองระดับได้แก่ ระดับออนโทโลยีสำหรับข้อมูลบริการ และระดับออนโทโลยีสำหรับผู้ใช้งานไปรวมไปถึงการเชื่อมโยงออนโทโลยีทั้งสองระดับเข้าด้วยกัน เพื่อสร้างเป็นเมตาดาต้าสำหรับตัวบริการเว็บที่สมบูรณ์

5.1 ความขัดแย้งกันในเชิงความหมายของข้อมูลบริการสำหรับโดเมน e-Learning

ในหัวข้อนี้จะแสดงตัวอย่างของความขัดแย้งกันของข้อมูลบริการจากผู้ให้บริการสองแหล่งที่มาลงทะเบียนกับระบบ LIS ได้แก่ ผู้ให้บริการจาก LMS1 และ ผู้ให้บริการจาก LMS2 ซึ่งแต่ละผู้ให้บริการที่มีการให้บริการที่เหมือนกัน อาจมีการประกาศข้อมูลบริการของตนเอง เช่น ชื่อบริการ ชื่อและที่อยู่ผู้ให้บริการ และตำแหน่งที่อยู่ของเอกสาร WSDL ซึ่งเป็นตำแหน่งของตัวบริการเว็บที่ถูกสร้างอยู่ในเว็บไซต์ของตน ซึ่งตัวอย่างข้อมูลบริการที่ประกาศอยู่ในเอกสาร WSDL ของแต่ละที่จะแสดงอยู่ในตารางที่ 5.1 ดังต่อไปนี้

ตารางที่ 5.1 แสดงตัวอย่างของข้อมูลบริการจากผู้ให้บริการที่แตกต่างกัน

LMS1: Web Service descriptions in WSDL file			
Operation: GetDesiredCourseDetails_Operation			
Inputs	Type	Outputs	Type
CourseCategory	String	CourseTitle	String
CourseTitle	String	CourseFileType	String
CourseKeyword	String	CourseFileSize	PositiveInteger
CourseFileType	String	CourseDescription	String
		CourseExpense	Decimal
LMS2: Web Service descriptions in WSDL file			
Operation: RequestSubject_Operation			
Inputs	Type	Outputs	Type
SubjectCategory	String	SubjectTitle	String
SubjectKeyword	String	SubjectFileType	String
SubjectFileType	String	SubjectFileSize	String
		SubjectDescription	String
		SubjectCost	Integer

จากข้อมูลในตารางที่ 5.1 จะแสดงถึงข้อมูลบริการจากผู้ให้บริการที่ 1 (LMS1) ที่ประกอบด้วยเมธอดหรือโอเปอเรชัน GetDesiredCourseDetails_Operation ซึ่งเป็นเมธอดที่ต้องการพารามิเตอร์นำเข้าสี่ตัวได้แก่

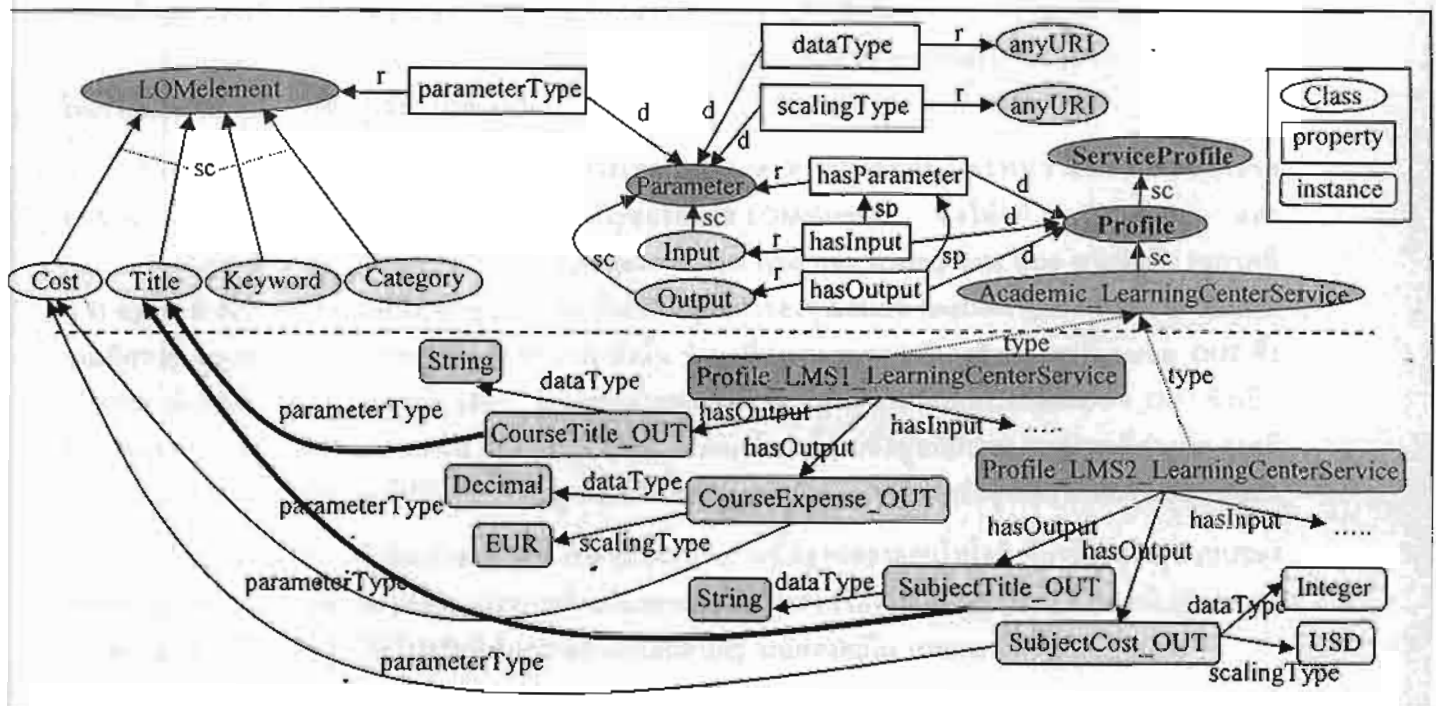
CourseCategory, CourseTitle, CourseKeyword, และ CourseFileType และมีพารามิเตอร์นำออกห้าตัว ได้แก่ CourseTitle, CourseFileType, CourseFileSize, CourseDescription, และ CourseExpense สำหรับผู้ให้บริการที่ 2 หรือ LMS2 จะมีโอเปอเรชัน RequestSubject_Operation ที่ประกอบด้วยพารามิเตอร์นำเข้าสามตัว ได้แก่ SubjectCategory, SubjectKeyword, และ SubjectFileType และมีพารามิเตอร์นำออกห้าตัว ได้แก่ SubjectTitle, SubjectFileType, SubjectFileSize, SubjectDescription, และ SubjectCost ซึ่งข้อมูลบริการจากทั้งสองผู้ให้บริการนี้จะนำไปสู่ความขัดแย้งกันของข้อมูลบริการสามประเภทด้วยกัน ประเภทแรก ได้แก่ ความขัดแย้งกันของพารามิเตอร์ที่เกิดขึ้นเมื่อแต่ละผู้ให้บริการมีการใช้ชื่อพารามิเตอร์ที่แตกต่างกันทั้งๆที่เป็นพารามิเตอร์ที่เหมือนกัน เช่น พารามิเตอร์นำเข้าชื่อCourseCategory จาก LMS1 และพารามิเตอร์นำเข้า SubjectCategory จาก LMS2 เป็นพารามิเตอร์ที่มีความหมายเหมือนกันแต่เขียนต่างกัน ความขัดแย้งประเภทที่สอง ได้แก่ ความขัดแย้งกันของประเภทข้อมูล ที่เกิดขึ้นเมื่อข้อมูลจากพารามิเตอร์ที่ส่งมาจากตัวบริการเว็บแต่ละแห่งมีประเภทข้อมูลที่แตกต่างกันทั้งๆที่เป็นข้อมูลแบบเดียวกัน เช่น พารามิเตอร์นำออก CourseFileSize จาก LMS1 จะมีประเภทข้อมูลเป็น "PositiveInteger" และ SubjectFileSize จาก LMS2 จะมีประเภทข้อมูลเป็น "String" ทั้งๆที่ข้อมูลจากพารามิเตอร์ทั้งสองจะเป็นข้อมูลขนาดไฟล์เหมือนกัน ความขัดแย้งประเภทที่สาม ได้แก่ ความขัดแย้งกันของหน่วยวัดข้อมูล ที่เกิดขึ้นเมื่อข้อมูลจากพารามิเตอร์ที่ส่งมาจากตัวบริการเว็บแต่ละแห่งมีหน่วยวัดข้อมูลที่แตกต่างกันทั้งๆที่เป็นข้อมูลแบบเดียวกัน เช่น พารามิเตอร์นำออก CourseExpense จาก LMS1 จะมีหน่วยวัดข้อมูลเป็น "EUR" และ SubjectCost จาก LMS2 จะมีหน่วยวัดข้อมูลเป็น "USD" เป็นต้น ตัวอย่างความขัดแย้งกันของข้อมูลบริการข้างต้นจะถูกใช้เป็นตัวอย่งในการออกแบบเมตาตาสำหรับตัวบริการเว็บดังจะแสดงในหัวข้อถัดไป

5.2 การออกแบบเมตาตาสำหรับตัวบริการเว็บบนพื้นฐานของออนโทโลยี

เมื่อเอกสาร WSDL ของผู้ให้บริการได้ถูกประกาศไว้โดยผ่านทางระบบ LIS เอกสาร WSDL จะถูกแปลงโดยผ่านโมดูลการแปลงข้อมูลบริการ ซึ่งจะทำการเชื่อมโยงข้อมูลบริการต่างๆในเอกสาร WSDL เช่น ชื่อเมธอด ชื่อพารามิเตอร์นำเข้าและออก ให้อยู่ในรูปแบบของเมตาตาสำหรับตัวบริการเว็บพื้นฐานที่ประกอบด้วยองค์ประกอบสำคัญสามส่วน ได้แก่ service profile, service model และ service grounding ที่อยู่ในรูปแบบของ OWL-S ซึ่งโครงสร้างของเมตาตาสำหรับตัวบริการเว็บนี้จะตั้งอยู่บนพื้นฐานการออกแบบที่ได้กล่าวไว้แล้วในบทที่ 4 สำหรับงานวิจัยนี้จะแสดงตัวอย่างของอินสแตนซ์ service profile จำนวนสองอินสแตนซ์ที่มีชื่อว่า Profile_LMS1_LearningCenterService และ Profile_LMS2_LearningCenterService ซึ่งเป็นอินสแตนซ์ของคลาส Academic_LearningCenterService ดังแสดงในภาพที่ 5.1 แต่ละ service profile อินสแตนซ์ จะสามารถเชื่อมโยงกับพารามิเตอร์อินสแตนซ์ทั้งที่เป็นพารามิเตอร์นำเข้าและพารามิเตอร์นำออกโดยผ่านทางคุณสมบัติ hasInput และ hasOutput ที่ถูกนิยามให้เป็นคุณสมบัติลูก (sub-property) ของคุณสมบัติ hasParameter อีกที ตัวอย่างเช่น พารามิเตอร์อินสแตนซ์ CourseTitle_OUT และ CourseExpense_OUT เป็นพารามิเตอร์อินสแตนซ์นำออกที่มีการประกาศโฆษณาอยู่ใน service profile ชื่อ Profile_LMS1_LearningCenterService ซึ่งทั้งสองพารามิเตอร์นี้จะอินสแตนซ์ของคลาส Output ซึ่งจะมีการเชื่อมความสัมพันธ์กับ Profile_LMS1_LearningCenterService โดยผ่านทางคุณสมบัติ hasOutput ในตารางที่ 5.2 จะแสดงตัวอย่างของกฎที่ได้มาตามนิยามของทฤษฎีเซตที่กล่าวไว้ในบทที่ 4

ตารางที่ 5.2. ตัวอย่างของกฎที่ได้มาตามนิยามของทฤษฎีเซต

นิยาม	กฎที่ได้มา
(Profile, <i>hasParameter</i> , Parameter) (hasInput, <i>subPropertyOf</i> , hasParameter) (hasOutput, <i>subPropertyOf</i> , hasParameter)	(Profile, <i>hasInput</i> , Parameter) (Profile, <i>hasOutput</i> , Parameter)
(Input, <i>subClassOf</i> , Parameter) (Output, <i>subClassOf</i> , Parameter)	(Profile, <i>hasInput</i> , Input) (Profile, <i>hasOutput</i> , Output)
(Academic_LearningCenterService, <i>subClassOf</i> , Profile)	(Academic_LearningCenterService, <i>hasInput</i> , Input) (Academic_LearningCenterService, <i>hasOutput</i> , Output)
(Profile_LMS1_LearningCenterService, <i>type</i> , Academic_LearningCenterService) (CourseTitle_OUT, <i>type</i> , Output) (CourseExpense_OUT, <i>type</i> , Output)	(Profile_LMS1_LearningCenterService, <i>hasOutput</i> , CourseTitle_OUT) (Profile_LMS1_LearningCenterService, <i>hasOutput</i> , CourseExpense_OUT)



ภาพที่ 5.1 แสดงบางส่วนของเมตาดาต้าสำหรับตัวบริการบนพื้นฐานของออนไลน์ ที่มีการเชื่อมโยงออนไลน์
ไต่ระดับข้อมูลบริการเข้ากับออนไลน์ไต่ระดับผู้ใช้ทั่วไป

พารามิเตอร์อินสแตนซ์ต่างๆที่ถูกประกาศอยู่ใน service profile นี้จะเป็นอินสแตนซ์ของคลาส Input และ Output ที่มีการสืบทอดคุณสมบัติ parameterType, dataType, และ scalingType จากคลาส Parameter ซึ่งความขัดแย้งกันของข้อมูลบริการก็จะถูกแก้ไขได้ด้วยวิธีของคุณสมบัติเหล่านี้ดังต่อไปนี้

เพื่อจะแก้ปัญหาความขัดแย้งกันของชื่อพารามิเตอร์อินสแตนซ์ กระบวนการเชื่อมโยงพารามิเตอร์ข้อมูลนำเข้าและออกจากservice profile ออนโทโลยี ไปสู่คลาสที่นิยามอยู่ในออนโทโลยีระดับผู้ใช้ทั่วไป จำเป็นต้องอาศัยตัวเชื่อมโยงข้อมูลบริการในการจัดหาเครื่องมือที่อยู่ในรูปแบบของอินเตอร์เฟส เพื่อช่วยให้ผู้ให้บริการสามารถเลือกการเชื่อมโยงพารามิเตอร์ดังกล่าว ให้เป็นข้อมูลบริการที่อยู่ในมุมมองของ LOM ซึ่งเป็นออนโทโลยีที่ผู้ใช่มองเห็นได้ ตัวอย่างของอินเตอร์เฟสการเชื่อมโยงข้อมูลบริการจะแสดงอยู่ในบทที่ 6

ดังนั้น จากตัวอย่างที่นำเสนอในหัวข้อข้างต้นจะพบว่าพารามิเตอร์อินสแตนซ์ CourseTitle_OUT จาก service profile ออนโทโลยี จะมีความคล้ายคลึงกับพารามิเตอร์ Title ที่เป็น subclass ของคลาส LOMelement ซึ่งอยู่ในออนโทโลยีสามัญ เนื่องจากทั้ง CourseTitle_OUT และ Title จะมีความหมายเหมือนกัน คือเก็บข้อมูลที่เป็นชื่อวิชา ดังนั้นจากนิยามที่ 7 ได้ว่า CourseTitle_OUT จะมีคุณสมบัติ parameterType ที่มีค่าเป็นคลาส Title ดังนี้

CourseTitle_OUT ~ Title $\longrightarrow$ (CourseTitle_OUT , parameterType, Title)

ดังนั้น จากตัวอย่างที่แสดงในภาพที่ 5.1 จะพบว่าทั้ง CourseTitle_OUT ซึ่งเป็นพารามิเตอร์อินสแตนซ์จาก Profile_LMS1_LearningCenterService และ SubjectTitle_OUT ซึ่งเป็นพารามิเตอร์อินสแตนซ์จาก Profile_LMS2_LearningCenterService จะเป็นอินสแตนซ์ของคลาส Output ซึ่งมีความสอดคล้องกับคลาส Title เพราะทั้งสองพารามิเตอร์จะมีความหมายเดียวกันกับคลาส Title ดังนั้นจึงเห็นได้ว่า การแก้ปัญหาความขัดแย้งกันในเชิงความหมายของข้อมูลบริการจะสามารถแก้ไขได้ด้วยการเชื่อมโยงพารามิเตอร์ของออนโทโลยีในระดับข้อมูลบริการให้เข้ากับออนโทโลยีในระดับผู้ใช้ทั่วไปนั่นเอง

ในการแก้ปัญหาความขัดแย้งกันของประเภทข้อมูลและหน่วยวัดข้อมูล จะพบว่าสับคลาสของคลาส LOMelement จะมีการสืบทอดคุณสมบัติที่สำคัญของคลาส LOMelement ซึ่งได้แก่ hasDataType และ hasScalingType ที่มีค่าของคุณสมบัติเป็นคลาส predefined data type และ scaling data type ตามลำดับ (ดูภาพที่ 4.4) ดังนั้น ถ้าสับคลาส Cost ที่นิยามอยู่บนออนโทโลยีสามัญ มีค่าของคุณสมบัติ hasDataType เป็นคลาส Decimal และมีค่าของคุณสมบัติ hasScaling เป็นคลาส USD ดังนั้น ข้อมูลที่ส่งมาจากพารามิเตอร์ CourseExpense_OUT ซึ่งเป็นพารามิเตอร์นำออกของ Profile_LMS1_LearningCenterService และพารามิเตอร์SubjectCost_OUT ซึ่งเป็นพารามิเตอร์นำออกของ Profile_LMS2_LearningCenterService ซึ่งอาจเป็นข้อมูลที่มีประเภทข้อมูลที่ต่างกัน และมีหน่วยวัดที่ต่างกัน ก็จะถูกแปลงให้มีประเภทข้อมูลเดียวกันคือ Decimal และมีหน่วยวัดข้อมูลเดียวกันคือ USD เป็นต้น

ตัวอย่างของโครงสร้างเมตาดาต้าสำหรับตัวบริการเว็บบนพื้นฐานของออนโทโลยี ที่แสดงอยู่ในรูปแบบของภาษา OWL-S สำหรับ service profile และการเชื่อมโยงพารามิเตอร์จากออนโทโลยีในระดับข้อมูลบริการไปยังคลาสที่อยู่ในออนโทโลยีในระดับผู้ใช้ทั่วไปหรือที่เรียกว่าออนโทโลยีสามัญ จะแสดงอยู่ใน ภาคผนวก ก ข และ ค

บทที่ 6

ผลการทดลอง

บทนี้จะอธิบายการดำเนินการทดลองเพื่อพัฒนาระบบ LIS รวมไปถึงการใช้งานระบบ LIS โดยในหัวข้อแรกจะแสดงถึงการออกแบบองค์ประกอบต่างๆของเครื่องมือที่จำเป็นที่ต้องใช้ในการพัฒนาระบบ LIS และการทำงานขององค์ประกอบต่างๆโดยรวมที่สอดคล้องกับสถาปัตยกรรมของระบบที่ได้นำเสนอในบทที่ 3 และหัวข้อถัดไปจะแสดงผลลัพธ์ที่ได้จากการทดลอง ซึ่งจะใช้ตัวอย่างจากกรณีศึกษาที่ได้นำเสนอไว้ในบทที่ 5 ซึ่งประกอบด้วยการใช้งานระบบ LIS ของผู้ให้บริการข้อมูล และการใช้งานของผู้ใช้งานทั่วไป ดังรายละเอียดต่อไปนี้

6.1 องค์ประกอบเครื่องมือต่างๆที่ใช้ในระบบ LIS

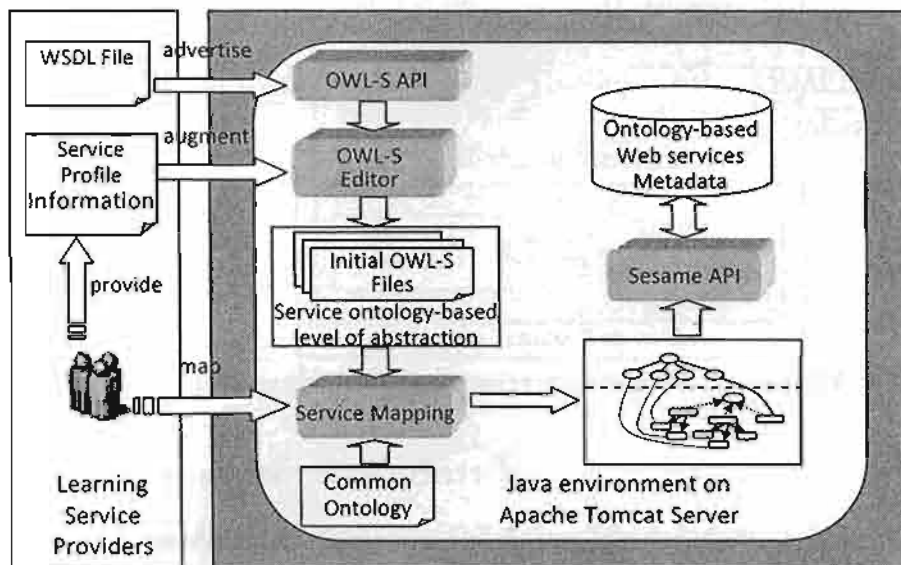
งานวิจัยนี้จะสร้างระบบที่ช่วยในการสืบค้นข้อมูลการเรียนรู้จากผู้ให้บริการข้อมูลการเรียนรู้ที่หลากหลาย โดยใช้เมตาดาต้าสำหรับตัวบริการเว็บบนพื้นฐานของออนโทโลยี เป็นเครื่องมือช่วยที่สำคัญ ระบบงาน LIS และเมตาดาต้าที่ได้ถูกออกแบบและพัฒนาโดยใช้ตัวอย่างจากกรณีศึกษาที่ได้กล่าวไว้ในบทที่ 5 เพื่อเป็นการพิสูจน์ว่าสามารถนำหลักการทางทฤษฎีของการออกแบบสถาปัตยกรรมของระบบ LIS และโครงสร้างเมตาดาต้า ไปสู่การปฏิบัติที่สามารถใช้งานได้จริง และสามารถแก้ปัญหาความขัดแย้งกันของข้อมูลบริการที่หลากหลายได้ ระบบงาน LIS ในปัจจุบันได้ถูกออกแบบและพัฒนาให้มีหน้าที่เป็นเหมือนตัวแทนผู้ให้บริการ (learning service broker) ที่คอยรับคำร้องขอจากผู้ใช้งานไป ในการสืบค้นข้อมูลการเรียนรู้ผ่านทางระบบ LIS เพื่อไปเรียกใช้บริการจากผู้ให้บริการที่มีการสร้างตัวบริการเว็บไว้ และยังสามารถรับคำร้องขอของผู้ให้บริการข้อมูลการเรียนรู้ ที่ต้องการมาประกาศข้อมูลบริการของตนที่มีการสร้างตัวบริการเว็บไว้

6.1.1 องค์ประกอบของเครื่องมือในการพัฒนาเมตาดาต้าสำหรับตัวบริการเว็บ

สำหรับการทำงานทางด้านผู้ให้บริการ งานวิจัยนี้ได้มีการทดลองสร้างตัวบริการเว็บให้กับระบบ LMS สองระบบ ที่ถูกพัฒนามนแพลตฟอร์มที่แตกต่างกัน ซึ่งเป็นผู้ให้บริการข้อมูลการเรียนรู้ โดยให้แต่ละระบบที่มีบริการที่เหมือนกัน มีการประกาศข้อมูลบริการที่หลากหลาย ซึ่งทำให้เกิดความขัดแย้งกันข้อมูลบริการขึ้น ตัวอย่างของภาพการทำงานโดยรวมขององค์ประกอบเครื่องมือที่สำคัญในระบบ LIS จะแสดงอยู่ในภาพที่ 6.1

เมื่อผู้ให้บริการได้ลงทะเบียนกับระบบ LIS โดยผ่านทางแบบฟอร์มที่ระบบ LIS ได้สร้างขึ้น ผู้ให้บริการที่เป็นสมาชิกของระบบ LIS จะสามารถกรอกตำแหน่งที่อยู่ของเอกสาร WSDL ที่สร้างจากตัวบริการเว็บของตนบนระบบ LIS โดยผ่านทางอินเตอร์เฟซสำหรับการติดต่อสื่อสาร (communication interface) ซึ่งจะทำการส่งข้อมูลเบื้องต้นนี้ให้กับโมดูลตัวแปลงข้อมูลบริการที่จะทำการอ่านเอกสาร WSDL และทำการแปลงคำอธิบายข้อมูลบริการที่อยู่ในเอกสาร WSDL ของผู้ให้บริการนั้นให้อยู่ในรูปแบบของเอกสาร OWL-S ซึ่งโมดูลตัวแปลงข้อมูลบริการนี้จะถูกพัฒนาโดยผ่าน OWL-S API [32] ซึ่งจะทำให้เกิดเอกสารพื้นฐานสามเอกสารขึ้น ได้แก่ service profile, service model และ service grounding ขึ้น หลังจากนั้นผู้ให้บริการสามารถกรอกข้อมูลบริการและข้อมูลผู้ให้บริการเพิ่มเติมเข้าไปในส่วนของ Service Profile โดยผ่านโมดูล OWL-S editor ที่ระบบสร้างขึ้น กระบวนการถัดไปคือการเชื่อมโยงพารามิเตอร์ที่อยู่ในออนโทโลยีระบบบริการให้เข้ากับคลาสต่างๆของ LOM ที่อยู่ในออนโทโลยีระดับสามัญ โดยผ่านโมดูลการเชื่อมโยงบริการ (service mapping process) ที่จะจัดหา interface สำหรับการเชื่อมโยงพารามิเตอร์นี้ให้กับผู้ให้บริการเลือกที่จะทำการเชื่อมโยงได้ และสำหรับกระบวนการสุดท้ายคือการทำงานที่เรียกว่า "OWL-S integration" ซึ่งจะใช้เครื่องมือ Sesame API [36] ในการรวมออนโทโลยีบริการที่

สร้างใหม่นี้ ให้เข้ากับเมตาดาต้าสำหรับตัวบริการเว็บที่ประกาศไว้แล้วในฐานความรู้เมตาดาต้า (metadata knowledge base) ของระบบ เพื่อสร้างเป็นเมตาดาต้าสำหรับตัวบริการเว็บที่สมบูรณ์และเก็บไว้ในฐานความรู้เมตาดาตาดังนี้



ภาพที่ 6.1 แสดงกระบวนการในการพัฒนาเมตาดาต้าสำหรับตัวบริการเว็บและเครื่องมือต่างๆที่เกี่ยวข้อง

6.1.2 องค์ประกอบของเครื่องมือในการสืบค้นข้อมูลบริการและข้อมูลการเรียนรู้

สำหรับการทำงานทางด้านผู้ใช้งานทั่วไป เมื่อผู้ใช้งานทั่วไปได้มาลงทะเบียนกับระบบ LIS และได้รับสิทธิในการเข้าใช้บริการการสืบค้นข้อมูลการเรียนรู้แล้ว ผู้ใช้งานทั่วไปสามารถส่งคำร้องที่อยู่ในรูปแบบมาตรฐาน LOM เข้ามาในระบบผ่านทางตัวสืบค้นข้อมูลบริการ (service discovery engine) ที่ถูกพัฒนาด้วย JSP [37] ที่อยู่ภายใต้การทำงานของ Java virtual machine และปฏิบัติงานอยู่บนเครื่องเซิร์ฟเวอร์ Apache Tomcat 5.5 ซึ่งตัวสืบค้นข้อมูลบริการนี้จะทำงานในลักษณะของ OWL Reasoner ที่มีการติดต่อกับฐานความรู้เมตาดาต้าโดยผ่านอินเทอร์เฟซ Jena 2.4 API [38] และ Java API [39] และจะส่งคำสั่งการสืบค้นที่อยู่ในรูปแบบของภาษา RDQL [40] ออกไปยังฐานความรู้เมตาดาต้า เพื่อสืบค้นข้อมูลบริการของผู้ให้บริการต่างๆที่มาลงทะเบียนจากฐานความรู้เมตาดาต้า เพื่อหาว่ามีข้อมูลบริการของผู้ให้บริการรายใดที่ตรงกับคำร้องขอของผู้ใช้งานทั่วไปหรือไม่ ข้อมูลบริการของผู้ให้บริการแต่ละรายที่สืบค้นได้จากฐานความรู้เมตาดาต้า จะถูกส่งให้กับโมดูลการเรียกใช้ตัวบริการ (Web service invocation) เพื่อจะทำการสร้างข้อความตามมาตรฐานของ SOAP และส่งไปยังผู้ให้บริการแต่ละรายที่สืบค้นได้นั้น องค์ประกอบในการสืบค้นข้อมูลบริการและข้อมูลการเรียนรู้จะแสดงอยู่ในภาพที่ 6.2

ผลลัพธ์จากการเรียกใช้ตัวบริการเว็บจากผู้ให้บริการจะถูกส่งกลับมายังในรูปแบบของมาตรฐาน SOAP เช่นเดียวกัน ก่อนที่จะถูกส่งไปยังโมดูลการแสดงผลบนเว็บในรูปแบบของเอกสาร HTML ที่สอดคล้องกับความต้องการของผู้ใช้ต่อไป

ในขั้นตอนถัดไปคือการลงทะเบียนของผู้ให้บริการ ซึ่งระบบ LIS จะแสดงหน้าจอสำหรับผู้ให้บริการข้อมูลการเรียนรู้ที่สามารถเข้ามาลงทะเบียนกับระบบ LIS โดยผู้ให้บริการจะต้องกรอกข้อมูลรายละเอียดเกี่ยวกับผู้ให้บริการเอง เช่น ชื่อ ที่อยู่ เบอร์โทรศัพท์ อีเมล และรหัสผู้ใช้ และรหัสผ่านที่ผู้ให้บริการจะสามารถ login เข้ามาในระบบ LIS ได้ หน้าจอสำหรับการลงทะเบียนผู้ให้บริการจะแสดงอยู่ในภาพที่ 6.4 ข้อมูลนี้จะถูกเก็บเพิ่มเติมอยู่ใน service profile ต่อไป

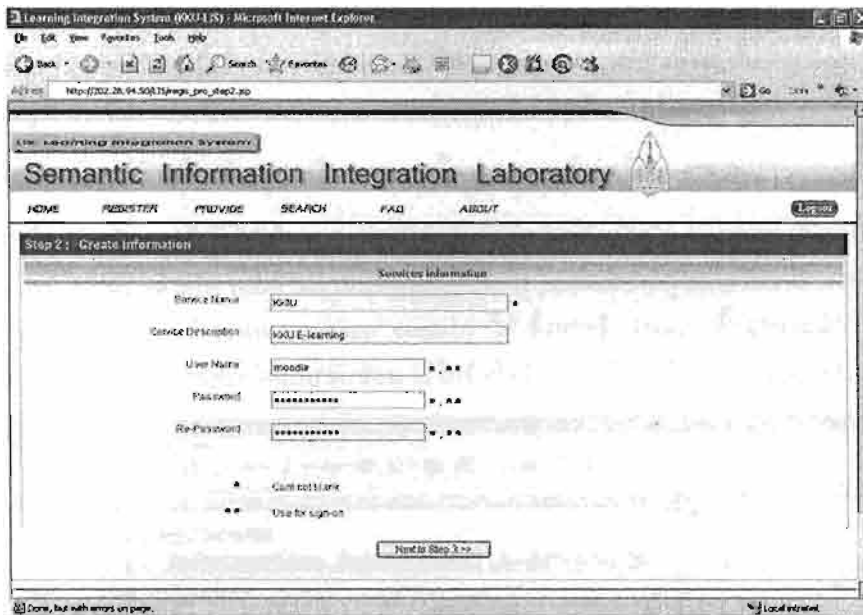
The screenshot shows a web browser window with the URL 'http://202.28.94.50/LIS/reg_provider.asp'. The page has a navigation bar with links: HOME, REGISTER, PROVIDER, SEARCH, FAQ, and ABOUT. The main content area is titled 'Provider Registration' and contains a form with the following fields:

- Provider Name: ngamnoi Aron wal
- Provider Description: KOU E-learning Creator
- Physical Address: Department of Computer Science Faculty of Science, Khon Kaen University
- Phone: 043-362188
- Email: ngamnoi@kku.ac.th
- User Name for login: ngamnoi
- Password for login: [masked]
- Re Password: [masked]

At the bottom of the form is a 'Submit' button. The status bar at the bottom of the browser window indicates 'Done, but with errors on page'.

ภาพที่ 6.4 แสดงหน้าจอการลงทะเบียนของผู้ให้บริการข้อมูลการเรียนรู้

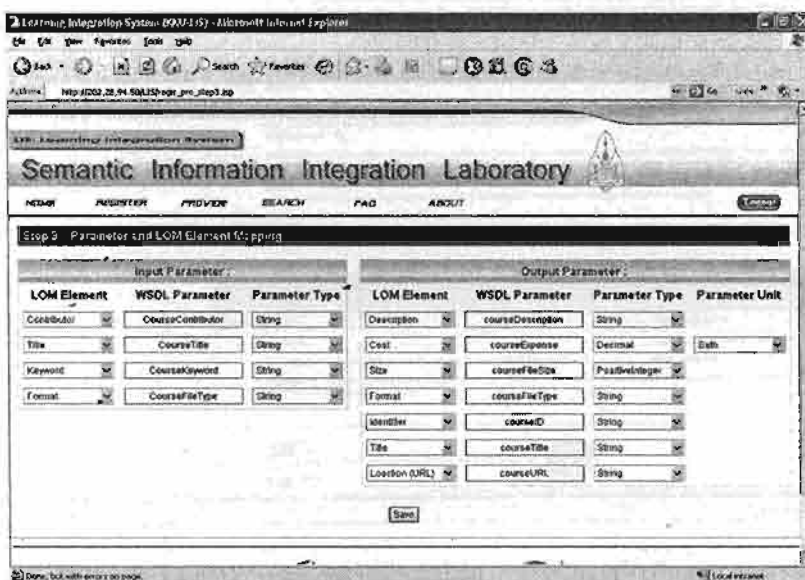
หลังจากที่ผู้ให้บริการทำการลงทะเบียนแล้ว ระบบจะทำการอ่านเอกสาร WSDL ของผู้ให้บริการและแสดงข้อมูลบริการต่างๆให้ผู้ให้บริการทราบ และแปลเป็นเอกสาร OWL-S เริ่มต้นให้ที่ประกอบด้วยเอกสาร service profile, service model และ service grounding ซึ่งเป็นเอกสารเริ่มต้นสำหรับการสร้างเมตาดาต้าสำหรับตัวบริการเว็บ เมื่อผู้ใช้คลิกปุ่มการทำงานถัดไป ก็จะมีแบบฟอร์มให้ผู้ให้บริการหรือผู้สร้าง LMS ทำการกรอกข้อมูลเพิ่มเติม อันได้แก่ ข้อมูลเกี่ยวกับบริการ ดังรายละเอียดในภาพที่ 6.5 เช่น ชื่อ Service คำอธิบายบริการ และรหัสผู้ใช้ และรหัสผ่าน ที่ผู้ให้บริการอนุญาตให้ผู้ใช้ทั่วไปของระบบ LIS สามารถเข้ามาใช้บริการเรียกดูข้อมูลการเรียนรู้ของ LMS นั้นได้โดยตรง โดยไม่ต้องผ่านการ login เข้ามาอีก ซึ่งข้อมูลนี้จะถูกเก็บอยู่ในฐานข้อมูลผู้ให้บริการต่อไป



ภาพที่ 6.5 แสดงหน้าจอการบันทึกข้อมูลรายละเอียดของผู้ให้บริการ

6.2.2 ตัวเชื่อมโยงบริการ

ในขั้นตอนถัดไปคือการเชื่อมโยงออนโทโลยีในระดับข้อมูลบริการให้เข้ากับออนโทโลยีในระดับผู้ใช้ทั่วไปหรือออนโทโลยีสามัญ ซึ่งในขั้นตอนนี้จะอาศัยตัวเชื่อมโยงบริการในการนำข้อมูลบริการจากฐานความรู้เมตาดาต้ามาแสดงแก่ผู้ให้บริการซึ่งประกอบด้วยพารามิเตอร์นำเข้าและนำออกแต่ละตัวของบริการที่ได้ลงทะเบียนไว้ ผู้ให้บริการสามารถเลือกการเชื่อมโยงของพารามิเตอร์แต่ละตัวกับคลาสต่างๆของ LOMelement ที่อยู่ในออนโทโลยีสามัญ ได้ด้วยตนเองดังตัวอย่างที่แสดงในภาพที่ 6.6



ภาพที่ 6.6 แสดงการเชื่อมโยงพารามิเตอร์ของออนโทโลยีในระดับข้อมูลบริการให้เข้ากับคลาส LOM ในออนโทโลยีสามัญ

การเชื่อมโยงออนโทโลยีจากทั้งสองออนโทโลยีนี้ นอกจากจะเป็นการแก้ปัญหาความขัดแย้งในเชิงความหมายของข้อมูลบริการ ซึ่งได้แก่ ชื่อพารามิเตอร์ด้วย ยังเปิดโอกาสให้ผู้ให้บริการสามารถกำหนดประเภทข้อมูลและหน่วยวัดข้อมูลของออนโทโลยีสามัญได้อีกด้วย เพื่อเป็นการแก้ปัญหาความขัดแย้งกันของประเภทข้อมูลและหน่วยวัดข้อมูล

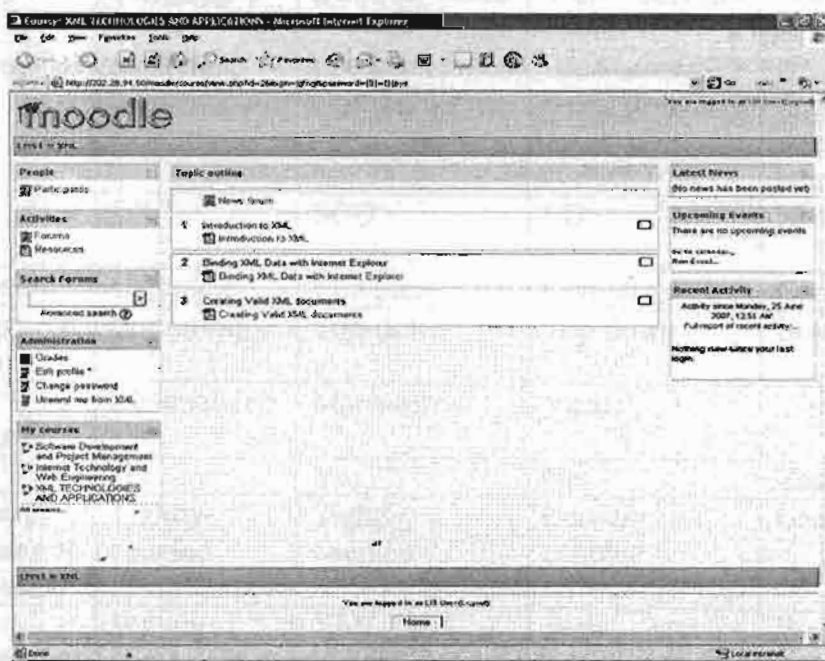
6.3 การสร้างผลลัพธ์การสืบค้นข้อมูลบริการและข้อมูลการเรียนรู้

ในขั้นตอนนี้จะเป็นขั้นตอนที่ผู้ใช้ทั่วไป สามารถส่งคำร้องขอในการสืบค้นข้อมูลการเรียนรู้บนพื้นฐานของมาตรฐาน LOM ที่จะระบุเป็นเงื่อนไขในการสืบค้น โดยผ่านตัวสืบค้นข้อมูลบริการ โดยในการนี้ผู้ใช้ทั่วไปสามารถเลือกการสืบค้นแบบพื้นฐาน (basic search) หรือแบบซับซ้อน (advance search) ได้ ซึ่งการสืบค้นแบบพื้นฐานจะมีการจำกัดอิลิเมนต์ของ LOM ที่จะทำการสืบค้น แต่ถ้าเป็นการสืบค้นแบบซับซ้อน จะไม่จำกัดจำนวนอิลิเมนต์ LOM ที่จะสืบค้น



ภาพที่ 6.7 แสดงผลลัพธ์การแสดงผลข้อมูลการเรียนรู้ที่สืบค้นได้จากตัวบริการเว็บที่หลากหลายที่สอดคล้องกับความต้องการของผู้ใช้ทั่วไป

ตัวอย่างเช่น ผู้ใช้ทั่วไปสามารถสืบค้นข้อมูลการเรียนรู้ โดยระบุเงื่อนไขบนมาตรฐาน LOM คือ Keyword คือ "XML" และ FileType คือ "pdf" ซึ่ง Keyword และ FileType นี้จะถูกเรียกว่าเป็น พารามิเตอร์นำเข้าที่ร้องขอ (requested parameters) และข้อมูล "XML" และ "pdf" เป็นข้อมูลเงื่อนไขของคำร้องขอ ซึ่งพารามิเตอร์ที่ร้องขอนี้จะถูกส่งไปยังตัวบริการสืบค้นเพื่อใช้ในการสืบค้นข้อมูลบริการจากเมตาดาต้าสำหรับตัวบริการเว็บในฐานะข้อมูลเมตาดาต้า ซึ่งอาจมีตัวบริการเว็บหลายแห่งที่มีพารามิเตอร์นำเข้าที่สอดคล้องกับพารามิเตอร์ร้องขอนี้ ข้อมูลจากตัวบริการเว็บที่สืบค้นได้นี้ รวมกับข้อมูลเงื่อนไขของคำร้องขอจากผู้ใช้ทั่วไปจะถูกนำมาสร้างเป็นเอกสาร SOAP เพื่อส่งไปยังตัวบริการเว็บที่สอดคล้องเหล่านั้นต่อไป ผลลัพธ์ทั้งที่เป็นพารามิเตอร์นำออกและข้อมูลผลลัพธ์ที่ถูกส่งมาจากตัวบริการเว็บจะถูกนำมาสร้างเป็นเอกสาร SOAP และส่งกลับมายังระบบ LIS เพื่อสร้างเป็นเอกสาร html เพื่อแสดงบนเว็บแก่ผู้ใช้ต่อไป ซึ่งแสดงได้ดังภาพที่ 6.7 ข้อมูลผลลัพธ์ที่ถูกส่งกลับมาแสดงจะประกอบด้วย ชื่อวิชา คำอธิบายรายวิชา คำเฉพาะ ประเภทของไฟล์บทเรียน(pdf, html, ฯลฯ) ขนาดของไฟล์ ตำแหน่งที่อยู่ URL ของเว็บไซต์ที่ให้บริการข้อมูลการเรียนรู้นั้น รวมไปถึงค่าใช้จ่ายในการเรียน บทเรียนนั้น เป็นต้น ซึ่งผู้เรียนทั่วไปจะสามารถคลิกที่ URL ของเว็บไซต์นั้น เพื่อเข้าสู่บทเรียนของเว็บไซต์ที่ให้บริการได้โดยตรง โดยไม่ต้องทำการ login เข้าสู่บทเรียนที่เว็บไซต์นั้นอีก เนื่องจากระบบจะนำ username และ password ที่ผู้ใช้บริการได้เข้ามาลงทะเบียนไว้ ส่งไปยัง URL ของเว็บไซต์นั้นให้โดยตรง ซึ่งเป็นการใช้คุณลักษณะการทำงานของ single sign on ที่ทำให้ผู้ใช้สามารถเข้าสู่บทเรียนของเว็บไซต์ที่ใดก็ได้ โดย login ผ่านทางระบบ LIS เพียงทีเดียว ดังตัวอย่างในรูปที่ 6.8 ที่ผู้ใช้คลิกเข้าสู่บทเรียนวิชา XML Technologies and Applications ของระบบ LMS Moodle ที่สร้างด้วยภาษา PHP ได้โดยตรง



ภาพที่ 6.8 แสดงหน้าจอการเข้าสู่ระบบ LMS Moodle ได้โดยตรงโดยผู้เรียนทั่วไป

บทที่ 7

อภิปรายผลการทดลอง

ในบทนี้จะทำการอภิปรายผลการทดลองโดยการเปรียบเทียบคุณลักษณะของระบบ LIS ในหลาย ๆ ด้านโดยเปรียบเทียบกับระบบงานอื่นๆ ที่แก้ปัญหาการสืบค้นข้อมูลจากแหล่งข้อมูลที่หลากหลายเหมือนกัน ทั้งในด้านการออกแบบสถาปัตยกรรม การประมวลผลการสืบค้น ประเภทของแหล่งข้อมูลที่ใช้ในการสืบค้น และ มุมมองข้อมูลของผู้ใช้ เป็นต้น

7.1 การเปรียบเทียบคุณลักษณะของระบบ LIS กับระบบอื่น

ระบบงาน LIS ที่ถูกสร้างขึ้นจากงานวิจัยนี้จะถูกเปรียบเทียบกับระบบงานอื่นๆ ที่ได้อ้างอิงถึงในวรรณกรรมที่เกี่ยวข้องของบทที่ 2 อันได้แก่ ระบบงาน e-Learning Broker [27] ระบบงาน Generic [3] และระบบงาน LearnServe [28] โดยจะมีการเปรียบเทียบกันในหลายแง่มุมดังแสดงในตารางที่ 7.1

ตารางที่ 7.1 : การเปรียบเทียบระบบ LIS กับระบบงานอื่น

	e-Learning Broker	Generic	LearnServe	LIS
Architecture	Semantic-based	SCORM-based	Web service-based	Semantic and Web Service-based
Data sources	Structure, Semantic structure	Unstructure, Semi-structure	Unstructure, Semi-structure	Unstructure, Structure, Semi-structure, Semantic structure
Common data model	OWL	SCO	LO	XML, OWL
Transparency	Complete	Complete	Complete	Complete
Bottom-up vs. Top-down	Bottom-up	Top-down	Top-down	Bottom-up
Virtual vs. materialized	Materialized	Materialized	Virtual	Virtual
Read-only	No	Yes	No	Yes
Tightly coupled vs. loosely coupled system	Tightly coupled	Tightly coupled	Loosely coupled	Loosely coupled
Mapping translation	Wrapper	XML-DB middleware	Web service interface	Web service interface
Machine-understandable	Yes	No	No	Yes
Semantic conflicts	Yes	No	No	Yes
Flexibility	Yes	No	Yes	Yes
Scalability	Yes	No	Yes	Yes
Interoperability	Yes	Yes	Yes	Yes

จากการเปรียบเทียบระบบ LIS กับระบบงานอื่นๆจะทำให้สามารถสรุปคุณลักษณะของระบบ LIS ได้ดังต่อไปนี้

- ระบบ LIS มีสถาปัตยกรรมเป็นแบบผสมผสานระหว่างเทคโนโลยีเว็บแบบสื่อความหมายและเทคโนโลยีตัวบริการเว็บ
- ระบบ LIS จะสามารถเข้าถึงแหล่งข้อมูล (data source) ที่หลากหลายทั้งข้อมูลที่ไม่มีโครงสร้าง เช่น text, ไฟล์ในรูปแบบต่างๆ (เช่น pdf, word, ฯลฯ) ข้อมูลที่มีโครงสร้างเช่น ฐานข้อมูล ข้อมูลแบบกึ่งโครงสร้าง เช่น html, XML และข้อมูลแบบสื่อความหมาย เช่น RDF, OWL เป็นต้น
- ระบบ LIS จะมีการสืบค้นข้อมูลการเรียนรู้และใช้โมเดลข้อมูลสามัญ อันได้แก่ XML ตามมาตรฐาน SOAP ในการรับส่งข้อมูลการเรียนรู้ระหว่างตัวบริการเว็บและระบบงาน LIS และมีการเก็บข้อมูลบริการโดยใช้โมเดลข้อมูลสามัญอันได้แก่ ภาษา OWL
- ระบบ LIS จะอำนวยความสะดวกในการสืบค้นข้อมูลการเรียนรู้โดยนำเสนอมุมมองข้อมูลที่อยู่ในรูปแบบของมาตรฐาน LOM ที่ผู้ใช้รู้จักโดยไม่ต้องไปยุ่งเกี่ยวกับตัวบริการเว็บแต่อย่างใด (user transparency)
- ระบบ LIS จะมีการออกแบบและพัฒนาแบบ Bottom-up โดยจะมีการพัฒนาจากแหล่งข้อมูลที่มีอยู่แล้ว (existing resources) ของผู้ให้บริการแต่ละที่มาสร้างเป็นตัวบริการเว็บ แล้วจึงไปทำการออกแบบองค์ประกอบต่างๆของระบบงาน LIS จนถึงการออกแบบออนไลน์สามัญสำหรับผู้ใช้ในระดับนสุด
- นอกจากนี้ระบบ LIS ยังถูกออกแบบบนพื้นฐานของสถาปัตยกรรมการบูรณาการแบบ virtual integration ซึ่งจะมีการสืบค้นข้อมูลจากตัวบริการเว็บ ณ เวลาที่ผู้ใช้มีการร้องขอทันทีทันใด จึงได้ข้อมูลที่มีความทันสมัยตลอด ซึ่งต่างกับบางระบบงานอื่นบางระบบที่เป็นแบบ materialized integration ซึ่งจะต้องมีการเก็บข้อมูลมาไว้ที่ระบบส่วนกลางก่อน จึงจะสามารถสืบค้นข้อมูลไปใช้ได้ ซึ่งบางครั้งอาจทำให้ได้ข้อมูลที่ไม่ทันสมัย
- ในการพัฒนาระบบ LIS ยังมีข้อจำกัดที่ระบบยังให้บริการแค่การสืบค้นข้อมูลการเรียนรู้เพียงอย่างเดียว ยังไม่ได้มีการพัฒนาบริการอย่างอื่น เช่น บริการการติดตามผลการเรียนของผู้เรียน ซึ่งจะเป็นหน้าที่ของ LMS ในแต่ละที่เมื่อผู้ใช้ทั่วไปสามารถเข้าไปใช้บริการใน LMS นั้นได้ หรือบริการการคิดค่าใช้จ่ายในการเรียนของผู้เรียน เป็นต้น
- เนื่องจากระบบ LIS จะมีออกแบบบนพื้นฐานของตัวบริการเว็บ ที่เปิดโอกาสให้แต่ละโปรแกรมประยุกต์สามารถติดต่อแลกเปลี่ยนข้อมูลกันได้ง่าย ดังนั้นจึงถือว่าเป็นระบบบูรณาการที่เป็นแบบ loosely coupled system ที่จะส่งผลให้ขยายระบบเมื่อมีผู้ให้บริการต้องการเข้าร่วมใช้งานกับระบบ LIS เพิ่มขึ้นจะสามารถทำได้ง่าย
- ระบบ LIS จะมีการสืบค้นข้อมูลการเรียนรู้ที่สืบค้นได้จากตัวบริการเว็บโดยผ่าน Web service interface และมาตรฐานโปรโตคอล SOAP
- ระบบ LIS จะมีเมตาดาต้าสำหรับตัวบริการเว็บ เพื่อเก็บคำอธิบายข้อมูลบริการด้วยภาษา OWL-S ซึ่งเป็นภาษาแบบสื่อความหมายที่ช่วยให้ซอฟต์แวร์เอเจนต์หรือโปรแกรมประยุกต์อื่นๆสามารถเข้าใจความหมายและสามารถสืบค้นข้อมูลบริการได้ (machine-understandable)
- เนื่องจากระบบ LIS จะมีการแทนเมตาดาต้าสำหรับตัวบริการเว็บด้วยภาษา OWL-S ซึ่งนอกจากจะช่วยให้การสืบค้นข้อมูลบริการสามารถทำได้ด้วยตัวสืบค้นข้อมูลบริการแล้ว ยังสามารถช่วยแก้ปัญหาความขัดแย้งกันในเชิงความหมายของข้อมูลบริการจากผู้ให้บริการที่หลากหลายได้อีกด้วย

- การออกแบบเมตาดาต้าสำหรับตัวบริการเว็บ จะมีความยืดหยุ่นในการเพิ่มเติมหรือแก้ไขข้อมูลบริการ โดยผ่านอินเตอร์เฟซที่ระบบ LIS ได้ออกแบบไว้ให้
- ระบบ LIS จะอำนวยความสะดวกในการขยายระบบ เมื่อมีผู้ให้บริการใหม่ต้องการเข้ามาใช้บริการของระบบ LIS เพิ่ม โดยผ่านระบบลงทะเบียนของระบบ LIS
- ระบบ LIS จะช่วยให้ระบบหรือโปรแกรมประยุกต์ที่มีความแตกต่างกัน สามารถติดต่อสื่อสารกัน และแลกเปลี่ยนข้อมูลร่วมกันได้ง่ายโดยผ่านทางเทคโนโลยีตัวบริการเว็บ

7.2 ข้อดีและข้อเสียของระบบ LIS

ในหัวข้อนี้จะสามารถสรุปข้อแตกต่างที่เป็นข้อเด่นของระบบ LIS ที่มีเหนือกว่าระบบอื่นดังต่อไปนี้

- (1) ระบบ LIS จะมีการสร้างฐานความรู้เมตาดาต้า ที่เก็บเมตาดาต้าสำหรับตัวบริการเว็บที่แทนด้วยภาษา OWL-S เพื่อเก็บข้อมูลบริการของตัวบริการเว็บแต่ละตัว ซึ่งทำให้ตัวสืบค้นข้อมูลบริการสามารถสืบค้นข้อมูลบริการได้ด้วยตนเอง โดยไม่ต้องมีคนมายุ่งเกี่ยว และการสืบค้นก็สามารถทำได้ตรงกับความต้องการของผู้ใช้ เนื่องจากข้อมูลบริการสามารถสืบค้นได้อย่างมีความหมาย อันเป็นการแก้ปัญหาความขัดแย้งกันของข้อมูลบริการได้
- (2) ผู้ใช้ทั่วไปสามารถส่งคำร้องขอในรูปแบบของอิลิเมนต์ LOM ที่ผู้ใช้รู้จัก โดยไม่ต้องยุ่งเกี่ยวกับข้อมูลบริการจริงของตัวบริการเว็บ ซึ่งคำร้องขอของผู้ใช้ทั่วไปนี้ถูกเชื่อมโยงไปยังข้อมูลบริการจริงของตัวบริการเว็บที่เก็บอยู่ในฐานความรู้เมตาดาต้าโดยอัตโนมัติโดยผ่านตัวเชื่อมโยงข้อมูลบริการเพียงที่เดียว โดยไม่ต้องไปหาข้อมูลบริการจากระบบ UDDI อื่นๆเอง
- (3) องค์ประกอบพื้นฐานของเมตาดาต้าสำหรับตัวบริการเว็บสามารถอธิบายได้ในรูปแบบของทฤษฎีเซตและ RDF model ที่สอดคล้องกับโครงสร้างภาษา OWL ที่ใช้แทนเมตาดาต้านี้
- (4) ระบบ LIS จะมีการออกแบบเมตาดาต้าสำหรับตัวบริการเว็บจะมีความยืดหยุ่นในการแก้ไขและเพิ่มเติมข้อมูลบริการ รองรับการขยายขนาดของระบบหรือการเพิ่มจำนวนผู้ให้บริการ และการใช้เทคโนโลยีตัวบริการเว็บจะช่วยให้การทำงานร่วมกันของโปรแกรมประยุกต์ที่แตกต่างกันของผู้ให้บริการที่หลากหลายสามารถทำได้ง่าย

อย่างไรก็ตาม ระบบ LIS ก็ยังมีข้อที่ยังต้องปรับปรุงเพิ่มเติมให้มีประสิทธิภาพในการทำงานมากยิ่งขึ้นในด้านต่างๆดังต่อไปนี้

- (1) ควรมีการสร้างบริการอื่นๆเพิ่มเติมอีก เช่น ระบบการคิดค่าบริการจากผู้ให้บริการ การสืบค้นข้อมูลการเรียนรู้ที่อยู่ในรูปแบบของออบเจกต์การเรียนรู้ เพื่อให้สามารถติดตามการเรียนรู้ของผู้เรียนได้ เป็นต้น
- (2) การเชื่อมโยงของออนโทโลยีในระดับผู้ใช้ทั่วไปหรือออนโทโลยีสามัญให้เข้ากับออนโทโลยีในระดับข้อมูลบริการ ควรสามารถทำได้อย่างอัตโนมัติ โดยไม่ต้องมีผู้ให้บริการต้องทำการเลือกข้อมูลการเชื่อมโยง ระบบควรจะสามารถแนะนำข้อมูลบริการที่มีความสอดคล้องกันให้กับผู้ให้บริการในการตัดสินใจการเชื่อมโยงได้
- (3) บางครั้งการสืบค้นข้อมูล ณ เวลาที่ผู้ใช้ส่งคำร้องขอ อาจต้องใช้เวลาในการสืบค้น เนื่องจากการสืบค้นข้อมูลจากแหล่งข้อมูลผู้ให้บริการโดยตรง ซึ่งจะขึ้นกับเครือข่ายว่ามีปัญหาหรือความล่าช้าหรือไม่ ซึ่งระบบควรมีการเก็บสำรองข้อมูลการเรียนรู้ของผู้ให้บริการที่ผู้ใช้เคยสืบค้นมาแล้วไว้ในระบบกลางเพื่อช่วยเพิ่มความเร็วในการสืบค้นได้อีกในระดับหนึ่ง

- (4) การเพิ่มระบบรักษาความปลอดภัยของการเข้าสู่ระบบของผู้ใช้ทั่วไป ควรพัฒนาให้เป็นไปตามมาตรฐานของ single sign on จริงๆ

บทที่ 8

สรุปผลงานวิจัย

งานวิจัยนี้จะนำเสนอสถาปัตยกรรมระบบบูรณาการข้อมูลการเรียนรู้จากระบบจัดการการเรียนรู้ที่หลากหลายเพื่อการแก้ปัญหาการทำงานร่วมกันของระบบจัดการการเรียนรู้ที่ต่างแพลตฟอร์มกัน ทำให้สามารถแลกเปลี่ยนข้อมูลร่วมกันได้ โดยใช้เทคโนโลยีตัวบริการเว็บร่วมกับออนโทโลยี ซึ่งงานวิจัยที่นำเสนอนี้มีการนำเสนอทั้งทางด้านทฤษฎีและทางปฏิบัติ เพื่อแสดงให้เห็นว่าสิ่งที่ได้ศึกษาจากงานวิจัยสามารถนำไปใช้ในทางปฏิบัติได้จริง จึงสามารถสรุปสิ่งที่ได้จากงานวิจัยดังต่อไปนี้

8.1 สรุปผลงานวิจัย

งานวิจัยนี้ได้มีการนำเสนอทั้งในด้านทฤษฎีและการประยุกต์ใช้ในทางปฏิบัติดังต่อไปนี้

- (1) ได้นำเสนอสถาปัตยกรรมระบบบูรณาการข้อมูลการเรียนรู้ที่เรียกว่า LIS เพื่อทำให้ระบบจัดการการเรียนรู้หรือ LMS ที่แตกต่างกันสามารถทำงานร่วมกันได้ และแก้ปัญหาความขัดแย้งกันของข้อมูลบริการ
- (2) ได้นำเสนอสถาปัตยกรรมการเชื่อมโยงตัวบริการเว็บที่ประกอบด้วยองค์ประกอบต่างๆที่ช่วยในการสร้างเมตาตาต้าสำหรับตัวบริการเว็บ และสถาปัตยกรรมการสืบค้นตัวบริการเว็บของระบบ LIS เพื่อช่วยให้การสืบค้นข้อมูลบริการจากตัวบริการเว็บสามารถทำได้อย่างอัตโนมัติ
- (3) ได้นำเสนอเทคนิคการโมเดลให้ได้มาซึ่งโครงสร้างเมตาตาต้าสำหรับตัวบริการเว็บบนพื้นฐานของออนโทโลยีเพื่อใช้เป็นองค์ประกอบสำคัญของระบบ LIS ในการแก้ปัญหาความขัดแย้งกันของข้อมูลบริการ และการสืบค้นข้อมูลบริการ
- (4) ได้นำเสนอองค์ประกอบพื้นฐานของเมตาตาต้าในรูปแบบของทฤษฎีเซต และโมเดล RDF ที่ใช้ในการอธิบายโครงสร้างพื้นฐานของเมตาตาต้า
- (5) ได้นำเสนอโครงสร้างเมตาตาต้าสำหรับตัวบริการเว็บที่อยู่ในรูปแบบของภาษา OWL-S ที่มีโครงสร้างพื้นฐานสอดคล้องกับโครงสร้างที่แสดงในรูปแบบของทฤษฎีเซตและโมเดล RDF จึงเป็นการพิสูจน์ถึงการนำไปใช้งานได้จริงของเมตาตาต้า การใช้ภาษา OWL-S ในการแทนโครงสร้างเมตาตาตาดังนี้ ยังช่วยในการสืบค้นข้อมูลของซอฟต์แวร์เอเจนต์หรือโปรแกรมประยุกต์ที่ทำให้สามารถเข้าใจความหมายร่วมกันของข้อมูลได้
- (6) โครงสร้างเมตาตาต้าสำหรับตัวบริการเว็บที่ได้ถูกนำมาประยุกต์ใช้แก้ปัญหาของระบบ e-Learning ในปัจจุบันนี้ ยังช่วยให้ผู้ใช้ทั่วไปสามารถส่งคำร้องขอที่อยู่ในรูปแบบของมาตรฐาน LOM บนออนโทโลยีสามัญ โดยไม่ต้องทราบว่าข้อมูลบริการนั้นถูกเก็บอยู่ในที่ใด ทั้งนี้เนื่องจากการสร้างตัวเชื่อมโยงข้อมูลบริการที่ถูกใช้เป็นเครื่องมือในการเชื่อมโยงออนโทโลยีในระดับข้อมูลบริการ ให้เข้ากับออนโทโลยีในระดับผู้ใช้ทั่วไป อีกทั้งยังทำให้สามารถแก้ปัญหาความขัดแย้งกันของข้อมูลบริการจากผู้ให้บริการที่หลากหลายได้อีกด้วย
- (7) การสร้างตัวสืบค้นข้อมูลบริการจะช่วยให้ระบบ LIS สามารถสืบค้นข้อมูลบริการของตัวบริการเว็บจากที่ต่างๆได้อย่างอัตโนมัติโดยอาศัยข้อมูลจากเมตาตาต้าสำหรับตัวบริการเว็บ อันทำให้การสืบค้นข้อมูลการเรียนรู้จากผู้ให้บริการผ่านทางตัวบริการเว็บสามารถทำได้อย่างอัตโนมัติและสอดคล้องกับความต้องการของผู้ใช้ทั่วไปมากที่สุด

8.2 ผลลัพธ์ที่ได้จากงานวิจัย

จากการดำเนินงานวิจัยมาเป็นระยะเวลา 2 ปีเต็ม สามารถสรุปผลงานวิจัยที่ได้โดยแบ่งเป็นผลงานตีพิมพ์และผลงานที่เป็นโปรแกรมระบบจากงานวิจัยดังต่อไปนี้

8.2.1 ผลงานตีพิมพ์

ผลงานที่เป็นบทความจากงานวิจัยที่ได้รับการตีพิมพ์แล้วจากงานประชุมทางวิชาการทั้งในระดับนานาชาติและระดับชาติจะมีดังต่อไปนี้

- Ngamnilj Arch-int, Chidchanok Lursinsup, and Peraphon Sophatsathit , " A Reference Architecture for Interoperating the existing e-Learning System using Metadata and Web Services Model ", *Proceedings of IEEE International Conference on Intelligent Agents, Web Technology, and Internet Commerce -IAWTIC'2005, Vienna, Austria (28-30 Nov, 2005)*, pp. 891-896. (แสดงอยู่ในภาคผนวก ก)
- Ngamnilj Arch-int, Chidchanok Lursinsup, Peraphon Sophatsathit, and Teerarat Amonpongkul , " E-Learning Systems Integration using Metadata and Web Services Model ", *Proceedings of the 2nd National Conference on Computing and Information Technology (NCCIT'2006)*, King Mongkut's Institute of Technology North Bangkok, Thailand, May 19-20,2006. (แสดงอยู่ในภาคผนวก ข)

ผลที่เป็นบทความที่กำลังอยู่ในระหว่างการพิจารณาตีพิมพ์ในวารสารทางวิชาการระดับนานาชาติ ได้แก่

- Ngamnilj Arch-int, Chidchanok Lursinsup, and Peraphon Sophatsathit , " Semantic Information Integration for e-Learning System using Common Ontology and Semantic Web Services Model ", *International Journal of Computers and Their Applications*, USA, 2007 (submitted).

8.2.2 ผลงานที่เป็นโปรแกรมระบบจากงานวิจัย

เมื่อเสร็จสิ้นโครงการงานวิจัย ผู้ดำเนินงานวิจัยได้พัฒนาโปรแกรมระบบบูรณาการข้อมูลการเรียนรู้ หรือที่เรียกว่าระบบ LIS ขึ้นมา เพื่อเป็นการทดสอบว่าสามารถนำความรู้ที่ได้จากงานวิจัยไปประยุกต์ใช้งานในเชิงปฏิบัติได้ โดยระบบที่ได้รับการพัฒนาขึ้นมานี้ จะถูกติดตั้งเพื่อให้ผู้ใช้ทั่วไปเข้ามาทดลองใช้งานได้ที่ <http://202.28.94.50/LIS/index.jsp>

โดยในเบื้องต้นนี้ จะให้ผู้ใช้ทั่วไปสามารถเข้ามาสืบค้นข้อมูลการเรียนรู้จากระบบ LMS ที่แตกต่างกันของภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น ที่มีการพัฒนาระบบ LMS สำหรับวิชาการระดับปริญญาตรี และระบบ LMS สำหรับวิชาการระดับปริญญาโท โดยอยู่แยกกันคนละระบบ แต่ระบบ LMS จะถือว่าเป็นผู้ให้บริการที่มีการสร้างตัวบริการเว็บไว้ ซึ่งระบบงาน LIS จะถูกพัฒนาเพื่อเป็นตัวกลางเชื่อมโยงระหว่างผู้ใช้ทั่วไปและระบบ LMS ต่างๆ เพื่อให้บริการสืบค้นข้อมูลวิชาได้ตามเงื่อนไขความต้องการของผู้ใช้ได้

8.3 ข้อเสนอแนะเพิ่มเติม

งานวิจัยนี้ยังมีส่วนที่สามารถทำเพิ่มเติมได้อีกในหลายจุดคือ

- (1) ในการเชื่อมโยงออนไลน์ในระดับข้อมูลบริการให้เข้ากับออนไลน์สามัญ ถึงแม้ในงานวิจัยนี้จะมีการพัฒนาตัวเชื่อมโยงข้อมูลบริการช่วยการสร้างอินเทอร์เน็ตเฟสของการเชื่อมโยงให้ แต่ก็ยังต้องให้ผู้ให้บริการเป็นผู้ตัดสินใจที่จะเชื่อมโยงพารามิเตอร์ตัวใดให้เข้ากับคลาสใน LOElement ซึ่งในอนาคตระบบควรพัฒนาตัว

เชื่อมโยงข้อมูลบริการให้สามารถแนะนำได้ว่ามีพารามิเตอร์ใดในออนโทโลยีระดับข้อมูลบริการที่มีความสอดคล้องกับคลาสที่อยู่ในออนโทโลยีสามัญได้ เพื่อช่วยแบ่งเบาภาระของผู้ให้บริการ

- (2) ตัวสืบค้นข้อมูลบริการสามารถปรับปรุงประสิทธิภาพการสืบค้นด้วยการพัฒนาอัลกอริทึมของการสืบค้นเพื่อการค้นหาวบริการเว็บที่มีพารามิเตอร์ทั้งนำเข้าและนำออกที่สอดคล้องกับพารามิเตอร์ที่ผู้ใช้ทั่วไปร้องขอซึ่งอาจมีทั้งพารามิเตอร์นำเข้าและนำออกด้วยเช่นกัน เพื่อให้ได้ตัวบริการเว็บที่สอดคล้องมากที่สุด
- (3) สามารถพัฒนาการเข้าสู่ระบบของสมาชิก ให้มีความปลอดภัยสูงขึ้นไปอีกตามมาตรฐานของเทคโนโลยี single sign on
- (4) งานวิจัยนี้ยังสามารถที่จะนำไปประยุกต์กับงานทางด้านอื่นๆที่ต้องการทำให้ระบบงานที่พัฒนาด้วยโปรแกรมประยุกต์ที่แตกต่างกันสามารถแลกเปลี่ยนข้อมูลร่วมกันได้ นอกเหนือจากการแก้ปัญหาของระบบ e-Learning ที่ได้นำเสนออยู่ในงานวิจัยนี้

บรรณานุกรม

- [1] ADL (Advanced Distributed Learning), *SCORM[®] 2nd Edition Overview* (2004). Available at: <http://www.adlnet.org>.
- [2] ADL (Advanced Distributed Learning), *ADL and the Sharable Content Object Reference Model (2002)*. Available at: <http://www.JointADLCoLab.org>.
- [3] P. Arapi, N. Moumoutzis, and S. Christodoulakis, Supporting Interoperability in an Existing e-Learning Platform using SCORM. In: *Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies, ICALT-03*, Greece (2003), 388-390.
- [4] ADL Technical Team, *SCORM Specification v1.2* (2003). Available at: <http://www.adlnet.org/index.cfm?fuseaction=scormabt>.
- [5] IEEE P1484.12.1, *Learning Object Metadata (LOM) Standard* (2002). Available at: <http://ltsc.ieee.org/wg12/>.
- [6] IMS Global learning Consortium Inc., *IMS Learning Resource Meta-data Information Model* (2001). Available at: <http://www.imsproject.org/>.
- [7] W3C, *Web Services Architecture* (2004). Available at: <http://www.w3.org/TR/ws-arch/>.
- [8] OASIS, *Universal Description, Discovery and Integration (UDDI)* (2001). Available at: <http://www.uddi.org/>.
- [9] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, *Web Services Description Language (WSDL) 1.1* (2001). Available at: <http://www.w3.org/TR/wsdl>.
- [10] W3C, *SOAP version 1.2 Part 0: Primer* (2003). Available at: <http://www.w3.org/TR/soap12-part0/>.
- [11] T.R. Gruber, A translation approach to portable ontology specifications, *Knowledge Acquisition* 4(2) (1993) 199-220.
- [12] D. Fensel, *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, Springer-Verlag, Berlin, 2001.
- [13] M. Uschold, and M. Gruninger, ONTOLOGIES: Principles, Methods and Applications, *The Knowledge Engineering Review*, 11(2) (1996) 93-155.
- [14] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara, *Owl-s: Semantic markup for web services* (2004). Available at: <http://www.daml.org/services/owl-s/1.1/overview/>.
- [15] CETIS (the Centre For Educational Technology Interoperability Standards), *What is IEEE Learning Object Metadata / IMS Learning Resource Metadata?* (2005). Available at: <http://www.cetis.ac.uk>.
- [16] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web, *Scientific American*, 284(5) (2001) 34-43.
- [17] H.S. Nwana, M. Wooldridge, Software Agent Technologies, *BT Technology Journal*, 14(4) (1996) 68-78.

- [18] G. Antoniou and F.V. Harmelet, *A Semantic Web Primer*, MIT press, USA, 2004.
- [19] W3C, *Extensible Markup Language (XML) 1.0* (1998). Available at: <http://www.w3.org/TR/REC-xml>.
- [20] O. Lassila and R. R. Swick. *Resource description framework (rdf) model and syntax specification*. W3C Recommendation (22 February 1999). Available at: <http://www.w3.org/RDF>.
- [21] D. Brickley, R.V. Guha, eds. *Resource Description Framework Schema (RDFS)*, W3C Proposed Rec. (03-Mar-1999). Available at: <http://www.w3.org/TR/PR-rdf-schema/>.
- [22] D. L. McGuinness and F. van Harmelen. *Owl: web ontology language overview*. W3C Recommendation (10 February 2004). Available at: <http://www.w3.org/TR/owl-ref/>.
- [23] F. van Harmelen, P. Patel-Schneider, and I. Horrocks. *Reference description of the DAML+OIL ontology markup language* (2001). Available at: <http://www.daml.org/2001/03/daml+oil-index.html>.
- [24] C. M. Daconta, J. L. Obrst, T. K. Smith. *The Semantic Web: A guide to the future of XML, Web Services, and Knowledge Management*, John Wiley & Sons, 2003.
- [25] US Census Bureau, *North American Industry Classification System (NAICS)* (1997). Available at: <http://www.census.gov/epcd/www/naics.html>.
- [26] United Nations and Dun & Bradstreet Co, *United Nations Standard Products and Services Codes (UNSPSC)* (1999). Available at: <http://eccma.org/unspsc>.
- [27] J.M. Santos, L. Anido, M. Llamas, Design of a Semantic Web-based Brokerage Architecture for the E-learning Domain. A Proposal for a Suitable Ontology, *Frontiers in Education*. In: *Proceedings of the 35th Annual Conference FIE '05*, Pontevedra, Spain (2005), S3H-18- S3H-23.
- [28] G. Vossen, P. Westerkamp, Maintenance and Exchange of Learning Objects in a Web Services Based E-Learning System, *Electronic Journal of e-Learning*, 2 (2) (2004) 292-304.
- [29] IMS Global Learning Consortium Inc., *IMS Content Packaging Specification v1.1.2* (1998). Available at: <http://www.imsproject.org/content/packaging/index.html>.
- [30] IMS Global Learning Consortium Inc., *IMS Learning Resource Metadata Specification v1.2.1* (1998). Available at: <http://www.imsproject.org/metadata/index.html>.
- [31] Z. Xu, Z Yin, A.E Saddik, A Web Services Oriented Framework for Dynamic E-Learning Systems, 2003, In: *Proceeding of the IEEE Canadian Conference on Electrical and Computer Engineering CCECE-2003*, Ottawa Univ., Canada (2003), 943 – 946.
- [32] MindSwap, “*OWL-S API*,” Available at: <http://www.mindswap.org/2004/owl-s/api/index.shtml>, (2004).
- [33] W3C, *XML Schema Part 2: Datatypes* (2001). W3C Recommendation. Available at <http://www.w3.org/TR/xmlschema-2>.

- [34] P.E.V. Vet and N.J.I. Mars, Bottom-Up Construction of Ontologies, *IEEE Transactions on Knowledge and Data Engineering* 10 (4) (1998) 513-526.
- [35] SemWebCentral, *OWL-S Editor - A tool to read, edit, and write OWL-S ontologies* (2004). Available at: <http://projects.semwebcentral.org/projects/owlseeditor/>.
- [36] J. Broekstra, A. Kampman, F. H. Van, *Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema*. In: I. Horrocks, J. Hendler, (eds.), *Proceedings of the 1st International Semantic Web Conference, ISWC-2002*, Sardinia, Italy, Springer (2002) 54-68.
- [37] Sun Microsystems Inc., *Java Server Page (JSP)* (2000). Available at: <http://java.sun.com/products/jsp/>.
- [38] Hewlett-Packard, *Jena: a Java framework for writing Semantic Web applications* (2003). Available at <http://www.hpl.hp.com/semweb/jena.htm>.
- [39] SUN Microsystems, *Java Development Kit 1.1.x*, 1999 <http://www.javasoft.com/products/jdk/1.1>.
- [40] A. Seaborne, *RDQL - A Query Language for RDF* (2004). Available at: <http://www.w3.org/Submission/RDQL/>.

ภาคผนวก

ภาคผนวก ก

บทความที่ได้รับการตีพิมพ์ในงานประชุมทางวิชาการระดับนานาชาติ

IEEE International Conference on Intelligent Agents, Web Technology, and Internet Commerce -IAWTIC'2005,

กรุงเวียนนา ประเทศออสเตรีย 28-30 พฤศจิกายน 2548

A Reference Architecture for interoperating existing e-Learning Systems using Metadata and Web Services Model*

Ngamnij Arch-int¹

Chidchanok Lursinsup²

Peraphon Sophatsathit²

¹Department of Computer Science, Faculty of Science, Khon Kaen University, Thailand 40002

²Department of Mathematics, Faculty of Science, Chulalongkorn University, Bangkok, Thailand

Email: ngamnij@kku.ac.th, lchidcha@chula.ac.th, Peraphon.S@chula.ac.th

Abstract

The increasing demand for interoperability existing heterogeneous e-learning systems to support accessibility, reusability, and adaptability has always been one of the most challenging researches. A number of research efforts have been proposed to address this problem, ranging from development of a SCORM conformant courseware to Web services-based e-learning systems. This paper proposes a reference architecture for interoperating the existing e-learning system with the help of Web services and a metadata-UDDI model. The proposed metadata-UDDI model is designed as a core component of the architecture and provided the service information for generating the lightweight communication protocols between heterogeneous e-learning platforms. As a consequence, the semantic service heterogeneity is eliminated. An XML-based data model is employed to manipulate and express the metadata-UDDI contents. The inherent flexibility of XML technology permits system-wide interoperability suitable for a Web-based environment.

1. Introduction

Currently, the educational system has been evolved into electronic learning (or e-learning) that enriches the traditional learning system by providing the autonomous learning for learners and allowing learner to learn anywhere and anytime. The e-learning system also provides a means to deliver skills and knowledge to the learner through the Learning Management

System (or LMS) bringing the learning system toward the student-center learning. With the greater benefits and growing acceptance of e-learning, most educational institutions develop e-learning systems according to their own requirements. Therefore, the heterogeneous e-learning platforms have been emerged that lead to lacking of flexibility, interoperability, and reusability. A lot of learning resources have been developed with redundancy and high cost investment.

Until recently there has been a significant effort to set up a unit of learning into Learning Object (LO) [1, 2] based on Sharable Content Object Reference Model (SCORM) [3, 4]. The objective of LO is to define a standard and appropriate description (or metadata) for LO in which LOs created from multiple sources can be easily discovered and reused. In order to support exchangeability and reusability of LO between different LMSs and flexibility to plug LO into a content package [5, 6], learning object should exhibit three properties, which are, cohesive, decoupled and context-free [7].

However, describing a course through SCORM needs reconstruction of the course almost from the beginning or finding a mechanism to map existing resources to SCORM content model components [8]. Hence, the SCORM conformant courseware will take more time and cost in developing processes. Beside, it is rather difficult to discover the relevant LOs which have been developed by different authors to be plugged into the courseware.

This research proposed a reference architecture for interoperating the existing e-learning systems. With the emergence of a promising Web services

* This work is part of the research project of Semantic Information Integration for e-Learning Systems using Ontology and Web Services model which is supported by the Thailand Research Fund (TRF), Thailand.

technology [9, 10], Web services are defined as the open standards that provide a flexible solution for integrating the heterogeneous applications and enabling the dynamic interoperability between different systems. The salient characteristics of Web Services render themselves attractive to be incorporated as an assistant mechanism for e-learning interoperability. The proposed reference architecture is designed specifically to enable content authors, instructors, learners and administrators operating in different organizations to discover and invoke services dispersed over heterogeneous systems, as well as management learning information, such as, LOs, and learning courses (e.g., HTML, documents, JPEG, GIF, etc.). The architecture incorporates the metadata-UDDI as a central directory for services registration and services discovery. In order to solve the semantic conflicts of service names provided by heterogeneous service providers, the metadata-UDDI also serves as a repository to store service data descriptions. These service data descriptions are subsequently used for automatic generating SOAP [11] messages facilitating communication between applications.

In order to support system-wide interoperability suitable for a Web-based environment, we choose XML as the language for expressing the metadata-UDDI contents, as well as providing flexibility and scalability in building and manipulating the service data descriptions.

The remainder of the paper is structured as follows. Section 2 describes closely related works. Section 3 furnishes an overview and details of reference architecture components. Section 4 presents the structural design of the XML-based metadata-UDDI which is a core component of the architecture. These include XML-DTD and a valid XML document structure. Section 5 concludes the paper and suggests further research extension.

2. Related Work

The interoperating of existing e-Learning systems is a challenging research area. Recent research endeavors furnish numerous approaches and methodologies for exchanging learning data between e-learning systems.

For example, the research of [8] proposed a generic architecture to support exporting and importing the SCORM 1.2 Content Packages called Package Interchange Files (PIF) between relational databases of the preexistent e-learning platform. The core component of the architecture is middleware tier responsible for finding all learning resources to be stored as SCO's and Assets in the Content Package, as

well as mapping and transforming between relational database schemas and SCORM XML Schemas.

The research of [12] proposed an architecture called LearnServe by applying the Web service paradigm to e-learning such that the learning objects can be exchanged and maintained between e-learning systems. The entire functionality of an e-learning system is decomposed into groups of activities which can be implemented as a variety of services. In order to support exchangeability and reusability learning contents, all LOs, classes, and courses that are stored on different servers need to be registered in a central repository together with meta-information. The Web service can be employed to read the metadata and store the information in the database provided by the repository. Hence, users can transfer the dynamic call of Web services to the call of learning contents.

The research of [13] proposed a Web services oriented framework for e-learning systems aiming to facilitate the discovery of learning components and applications and the loosely connection. The framework enables different LMSs to interact with each other and share their services via the Web service interface. Service providers can publish their learning services in the UDDI [14] that can be discovered by other service requester. The framework also provides a single sign on process for learners in accessing courses from one institution to another.

Although, a number of researches aim to propose the architectures or frameworks to interoperate existing e-learning systems such that the learning contents can be exchanged and reused, none of these approaches emphasize on solving semantic conflicts of the service names provided by different service providers.

3. Reference Architecture

In this section the proposed reference architecture for interoperating the existing e-learning systems is depicted in Figure 1. The architecture consists of multiple tiers, namely, web client components, learning services integration center, metadata-UDDI, and learning service providers, as described below:

(1) **Web Client Components** consist of learners, and content authors (i.e., teachers and instructional designers) whose serve as the learning service requesters. The web client components communicate with learning service integration center in a Web-based environment. This tier provides location transparency to users for maximal flexibility. The users can search requests without any priori knowledge of local sources or remote sources to be accessed through Web services.

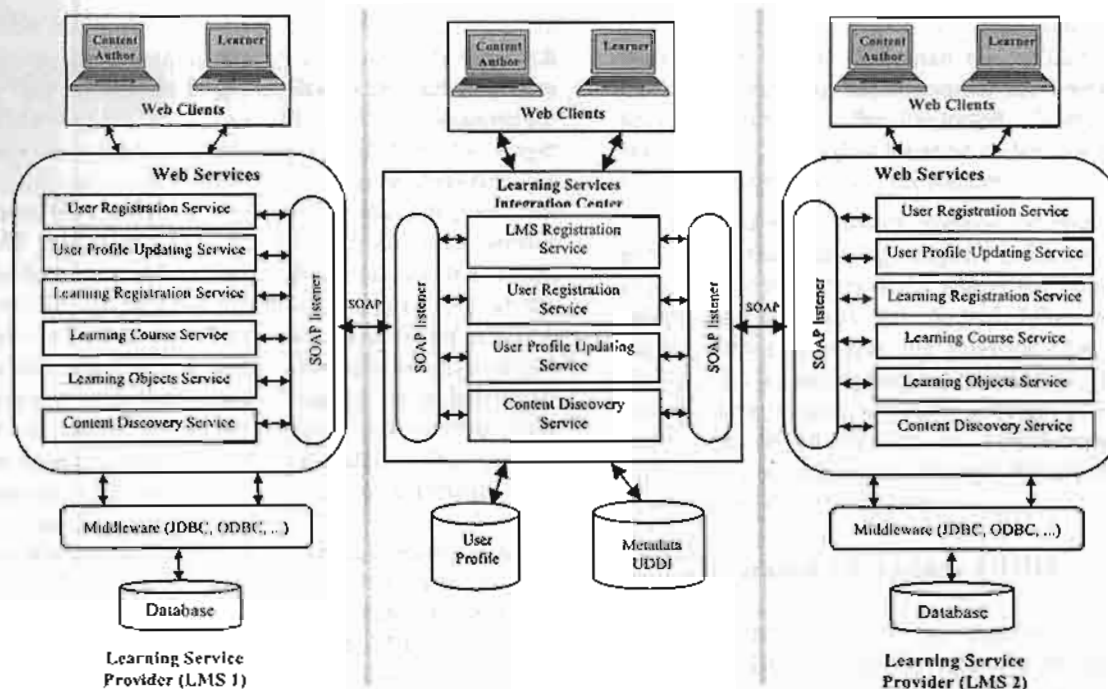


Figure 1. A reference architecture for interoperating the existing e-learning systems.

Learners can search for learning courses by specifying a part of course name or topic in the course as a keyword for searching. Upon submission of a learner query, the relevant learning courses that can be arbitrarily distributed in different LMSs will be consolidated and returned to the learner via the content discovery service. On the other hand, the content authors who need to create their courses from existing LOs can search for LOs and plug them into the course so that the developing time could be shortened.

(2) **Learning Service Integration Center** is a local SCORM compatible LMS and serves as a learning service broker which obtains the user requests from web client components and subsequently invokes the corresponding services to serve the user requests. This tier is made up of four principle services, namely, the LMS registration, user registration, user profile updating and content discovery services as described below:

(2.1) *LMS registration service* enables other learning service providers in different remote LMSs to register their learning service information described by Web Service Description Language (WSDL) [15], such as, service creator, service names, service methods, location of service providers, and input/output parameters, etc. This service information

is registered in the metadata-UDDI directory that are subsequently discovered and invoked according to the user requests.

(2.2) *User registration service* enables the new web client components register to the Learning Service Integration Center. The user registration service will log the user information (e.g., username, password, registry course, and learning status, etc.) into the user profile. These user information are subsequently used by content discovery service to decide the suitable course or LO for each user during the learning session.

In order to provide the maximal flexibility and loosely connection for users in accessing learning resources from anywhere, the user profile information can be exchanged with other user registration services of different learning service providers. This can be achieved with the help of Web service technology.

(2.3) *User profile updating service* is responsible for tracking of user activities and updating the user profile during the learning process, such as, the registry courses, completed courses, and learning status.

(2.4) *Content discovery service* enables users to search for the required learning courses and LOs. Upon submission of a user query, the content discovery service needs to search for the user information in the user profile, as well as the prerequisite courses such that the appropriate courses

or LOs can be selected and provided to the users. In addition, the content discovery service needs to search for the corresponding service information from metadata-UDDI, as well as initiate SOAP messages for transmitting service request information to the target learning service providers. In the reverse direction, the content discovery service collects and integrates the SOAP response results obtained from each service providers and subsequently forwards to the users, where the presentation format is carried out at the upper web client tier. Upon the presentation of list of available learning courses or LOs, the web clients can register to a desired course locating on a distributed LMS by calling the learning registration service of the corresponding LMS service provider. This can be achieved with the help of service information in metadata-UDDI designs.

(3) **Metadata-UDDI** is designed and incorporated as a core component of the middle tier in order to cope with the service name conflicts provided by different service providers. The metadata-UDDI contents are systematically implemented by means of flexible XML data model. As a consequence, the service information from metadata will be wrapped into a SOAP message such that the communication protocol between local LMS and distributed service providers can be carried out. In so doing, cross systems interoperability can be achieved with the help of XML flexible metadata-UDDI model. The details of metadata-UDDI contents will be described in the next section.

(4) **Learning Service Providers** consist of existing distributed LMSs located at different educational sites. This tier aims to provide various kinds of services, such as, the user registration, user profile updating, course registration, learning courses, learning objects, and content discovery services. As such, the services of user registration, user profile updating, and content discovery provide the same responsibility as described in the aforementioned learning service integration center. The rest of services are described below.

- **Learning registration service.** Once the web clients select a course or LO to be consumed, the learning registration service will log the user information and learning course or LO information into the registry information database. This information will also be used to updating the user profile repository, as well as be transmitted to the learning course service (or learning object service) where the learning information retrieval is carried out and provided to the web client components.

- **Learning course service** is responsible for providing course information (e.g., course id, course name, and prerequisites courses, etc) to the content

discovery service where the selection of suitable courses for learner is carried out. Once the learner decides to register for a course, the learning course service provides the selected learning course information (e.g., url or location of the learning course, course description, etc.) for learner.

- **Learning object service** is responsible for providing the learning object information to the content discovery service. Once the content author selects and registers the desired LOs, the learning object service provides the relevant LOs to content author. As a consequent, these LOs will be plugged into the new course to render greater reusability and adaptability to the LO purpose. Searching for relevant LOs can be accomplished through the IEEE's learning object metadata (LOM) [16].

4. XML-based Metadata-UDDI

This section focuses on designing XML-based metadata-UDDI in order to cope with the service name conflicts provided by different service providers. Due to the fact that each service provider designs and creates their service information independently, catering to the needs of their own requirements, it is inevitable that the problem of semantic name conflicts has been arisen. The metadata-UDDI is designed to be a repository for storing the service data descriptions. These service data descriptions are subsequently used to generating SOAP requests/responses as the communication messages between the heterogeneous LMSs where interrogating process is initiated. As a consequence, the metadata-UDDI design also provides greater benefits to solving service name conflicts in distributed environment.

The fact that a XML-based metadata-UDDI consisting of XML-DTD incorporated with XML documents makes it imperative for an XML document needs to be validated by rules defined through XML-DTD (document type definition) representing XML data schematic description. The structural design of a XML-DTD is set up as depicted in Figure 2 and the example of a valid XML document conforming to XML-DTD is depicted in Figure 3.

The XML-DTD structure in Figure 2 illustrates elements and attributes for storing the service data description. The element Metadata-UDDI starts as the root element of a document and consists of two sub-elements, that is, Metadata and UDDI elements. Each of these elements can be described as follows:

```

<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Metadata-UDDI (Metadata, UDDI)>
<!ELEMENT Metadata (searchBy+)>
<!ELEMENT searchBy (service+)>
<!ATTLIST searchBy name (subject_name | keyword) #REQUIRED>
<!ELEMENT service (wsdlLocation, method, input, outputs, provider)>
<!ELEMENT wsdlLocation (#PCDATA)>
<!ELEMENT method (#PCDATA)>
<!ELEMENT input (#PCDATA)>
<!ELEMENT outputs (output)>
<!ELEMENT output (#PCDATA)>
<!ELEMENT provider (#PCDATA)>
<!ATTLIST provider providerKey IDREF #IMPLIED>

<!ELEMENT UDDI (serviceProviders)>
<!ELEMENT serviceProviders (serviceProvider+)>
<!ELEMENT serviceProvider (name, url, address, description, contacts)>
<!ATTLIST serviceProvider providerID ID #REQUIRED>
<!ELEMENT name (#PCDATA)>
<!ELEMENT url (#PCDATA)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT contacts (personName, email)>
<!ELEMENT personName (#PCDATA)>
<!ELEMENT email (#PCDATA)>

```

Figure 2. XML-DTD metadata-UDDI structure.

(1) The Metadata element

In order to provide several ways for searching the learning courses or LOs, such as, search by subject name (e.g., course name or LO name) or by keyword (e.g., topics within the course name). The element Metadata is designed to contain one or more sub-elements searchBy whose value is atomic value (e.g., subject_name, or keyword, etc.).

The design purpose of this element is to solve the semantic name conflicts which occur when there is a disagreement on the meaning, interpretation, or intended use of the same or related data [17]. In such a way that the semantically equivalent service data descriptions, such as, service names, service methods, input and output parameters, and wsdl location are defined by different names. Therefore, each searchBy element is designed to hold one or more sub-elements service, which in turn consists of several sub-elements, that is, wsdlLocation, method, input, outputs, and provider. These sub-elements are synonymous service data descriptions provided by different service providers. As such, the element provider contains attribute providerKey defined as *idref* to reference the corresponding service provider name defined as *id* of providerID.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Metadata-UDDI SYSTEM "Metadata-UDDI.dtd">
<Metadata-UDDI>
  <Metadata>
    <searchBy name="subject_name">
      <service>
        <wsdlLocation>http://10.177.113.14:8090/lms1/
          searchsubjectname.php?wsdl</wsdlLocation>
        <method>getCourseName</method>
        <input>courseName</input>
        <outputs>
          <output>searchCourseNameResult</output>
        </outputs>
        <provider providerKey="P000001"/>
      </service>
    </searchBy>
    <searchBy name="keyword">
      <service>
        <wsdlLocation>http://10.177.113.14:8090/lms1/
          searchkeyword.php?wsdl</wsdlLocation>
        <method>getKeyword</method>
        <input>keyword</input>
      </service>
    </searchBy>
  </Metadata>
  <UDDI>
    <serviceProviders>
      <serviceProvider providerID="P000001">
        <name>Khon Kaen University</name>
        <url>http://www.kku.ac.th</url>
        <address>123 Mittraphap Road, Khon Kaen 40002</address>
        <description>e-learning services</description>
      </serviceProvider>
    </serviceProviders>
  </UDDI>
</Metadata-UDDI>

```

Figure 3. XML document conforming to the XML-DTD.

(2) The UDDI element

The design purpose of an UDDI element is to hold a sub-element serviceProviders which in turn consists of one or more sub-element serviceProviders. As such, each element serviceProvider holds sub-elements name, url, address, description, and contact of service provider to provide service provider information.

5. Conclusion and Future Work

The main objective of the proposed reference architecture is to render maximal interoperability of the existing e-learning across heterogeneous systems. The architecture also enhances greater benefits of e-learning, namely, the accessibility, reusability, adaptability, affordability and flexibility. With the salient characteristics of Web services technology, we combine this technology in the reference architecture to facilitate e-learning interoperability. Additional benefits precipitated from this work will encompass the actual implementation of metadata-UDDI by means of XML technology which, in turn, offers system scalability by virtue of XML constructs. The metadata-UDDI is a significant component of the architecture that will be used to generate the standard SOAP messages to bridge the communication gap between heterogeneous e-learning platforms. Learning information querying and retrieving from heterogeneous LMSs can be realistically accomplished through this metadata-UDDI model. As a consequence, the semantic service heterogeneity is thus be eliminated.

Our further work will focus on extension of other services, such as, the sequencing service, course management service, and assessment service, etc. We will also enrich the proposed metadata-UDDI with advanced ontology language such as OWL [18] to enhance XML universal expressive power and syntactic interoperability toward machine-processable Semantic Web [19].

6. References

- [1] T. Boyle, "Design principles for authoring dynamic, reusable learning objects," *Australian Journal of Educational Technology*, vol. 19 (1), pp. 46-58.
- [2] D. Wiley, "Learning Object Design and Sequencing Theory," *Doctoral Dissertation, Dept. of Instructional Psychology and Technology*, Brigham Young University, 2000.
- [3] Advanced Distributed Learning (ADL) (2004), "SCORM® 2nd Edition Overview," available at <http://www.adlnet.org>.
- [4] ADL Technical Team (2003), "SCORM Specification v1.2," available at <http://www.adlnet.org/index.cfm?fuseaction=scormabt>.
- [5] Advanced Distributed Learning (ADL) (2004), "SCORM Content Aggregation Model Version 1.3.1," available at <http://www.adlnet.org/>
- [6] Advanced Distributed Learning (ADL), (2002), "ADL and the Sharable Content Object Reference Model," available at <http://www.JointADLCoLab.org>.
- [7] R. Jones, "Designing Adaptable Learning Resources with Learning Object Patterns," *Journal of Digital Information*, vol. 6 (1), 2004.
- [8] P. Arapi, N. Moutoutzis, and S. Christodoulakis, "Supporting Interoperability in an Existing e-Learning Platform using SCORM," *In the proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies (ICALT 2003)*, Greece, July 2003.
- [9] W3C, "Web Services Architecture," Available from: <http://www.w3.org/TR/ws-arch/>, W3C Working Group Note [11-Feb-2004].
- [10] W. Hoschek, "The Web Service Discovery Architecture," *Proceeding of the IEEE/ACM Supercomputing Conference (SC 2002)*, Maryland, pp.1-15.
- [11] W3C, "SOAP version 1.2 Part 0: Primer," Available from: <http://www.w3.org/TR/soap12-part0/>, W3C Recommendation [24-Jun-2003].
- [12] G. Vossen and P. Westerkamp, "Maintenance and Exchange of Learning Objects in a Web Services Based e-Learning System," *Electronic Journal of e-Learning*, vol. 2 (2), pp. 292-304, 2004.
- [13] Z. Xu, Z. Yin, and A. E. Saddik, "A Web Services Oriented Framework for Dynamic E-Learning Systems," *Proceeding of the IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2003)*, Montreal, Canada, vol. 1, pp. 943-946, 4-7 May 2003.
- [14] OASIS, "Universal Description, Discovery and Integration (UDDI) (2001)," Available from: <http://www.uddi.org/>.
- [15] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL) 1.1," Available from: <http://www.w3.org/TR/wsdl>, W3C Note [15-Mar-2001].
- [16] IEEE P1484.12, "Learning Object Metadata (LOM) Standard", Available from: <http://ltsc.ieee.org/wg12/>.
- [17] A.P. Sheth and J.A. Larson, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases," *ACM Computing Surveys*, vol. 22 (3), pp. 183-236, 1990.
- [18] F. V. Harmelen, J. Hendler, I. Horrocks, L. A. Stein, D. L. McGuinness, P. F. Patel-Schneider, "Web Ontology Language (OWL) Reference Version 1.0," *W3C Working Draft*, available at <http://www.w3.org/TR/owl-ref/>, February 2003.
- [19] S. Decker, S. Melnik, F. Van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann and I. Horrocks, "The Semantic Web: The Roles of XML and RDF," *IEEE Internet Computing* 4, pp. 63-74, September/October, 2000.

ภาคผนวก ข

บทความที่ได้รับการตีพิมพ์ในงานประชุมทางวิชาการระดับชาติ

Proceedings of the 2nd National Conference on Computing and Information Technology (NCCIT'2006),

สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ กรุงเทพฯ 19-20 พฤษภาคม 2549

การบูรณาการระบบการเรียนรู้ E-learning โดยวิธีเมตาดาต้าและโมเดลตัวบริการเว็บ E-Learning Systems Integration using Metadata and Web Services Model

งามนิจ อาจอินทร์¹ ชิดชนก เหลือสินทรัพย์² พีระพนธ์ ไสพัสสกิจ² อีรารัตน์ อมรพงษ์กุล¹

¹ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น 40002

²ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย 10330

Email: ngamnij@kku.ac.th, Ichidcha@chula.ac.th, Peraphon.S@chula.ac.th, ateerarat@yahoo.com

บทคัดย่อ

ในปัจจุบันมีความต้องการที่จะทำให้ระบบ e-learning ที่ถูกพัฒนาด้วยแพลตฟอร์มที่แตกต่างกัน ให้มีความสามารถที่จะทำงานร่วมกันได้ แลกเปลี่ยนข้อมูลการเรียนรู้ร่วมกันได้ และการนำข้อมูลการเรียนรู้ที่ถูกสร้างไว้แล้วกลับมาใช้ใหม่ หลายงานวิจัยได้มีการนำเสนอวิธีการแก้ปัญหาดังกล่าว ตั้งแต่การพัฒนาข้อมูลการเรียนรู้บนมาตรฐาน SCORM ไปจนถึงการพัฒนาตัวบริการเว็บ บนระบบจัดการการเรียนรู้ที่หลากหลายงานวิจัยนี้ ได้นำเสนอสถาปัตยกรรมการทำงานร่วมกันของระบบ e-learning ที่แตกต่างกัน โดยวิธีเมตาดาต้า และโมเดลตัวบริการเว็บ ซึ่งเมตาดาต้านี้ ถูกออกแบบให้เป็นองค์ประกอบหลักของสถาปัตยกรรม เพื่อช่วยในการสืบค้นบริการต่างๆ จากตัวบริการเว็บ และแก้ปัญหาความหลากหลายทางด้านความหมายของบริการต่างๆ จากตัวบริการเว็บ ที่ถูกสร้างโดยผู้ให้บริการที่หลากหลาย โครงสร้างของเมตาดาต้าจะถูกแทนด้วยภาษา XML เพื่อให้มีความยืดหยุ่นในการจัดการเมตาดาต้า และการเข้าใจความหมายของข้อมูลร่วมกัน อันเป็นการสนับสนุนให้เกิดการทำงานร่วมกันของระบบ e-learning อย่างแท้จริง

Abstract

Due to the increasing demands in interoperating the heterogeneous e-learning systems to support the learning data exchangeability, and reusability, a number of researches have been proposed to address this problem ranging from development of a SCORM conformant courseware to Web services-based e-learning systems. This research proposes the architecture for interoperating the heterogeneous e-learning systems using metadata and Web services model. The metadata

has been designed to be a core component of the architecture to provide service discovery, as well as elimination of the semantic service heterogeneity. An XML-based data model is employed to manipulate and express the metadata contents. The inherent flexibility of XML technology permits system-wide interoperability suitable for a Web-based environment.

Keywords: Web Services, Metadata, e-learning Interoperability

1. บทนำ

ระบบการศึกษาในปัจจุบัน ได้มีการพัฒนาไปสู่ระบบการเรียนรู้ผ่านสื่ออิเล็กทรอนิกส์ หรือที่เรียกว่า e-learning เพื่อให้ผู้เรียนสามารถศึกษาเรียนรู้ได้ด้วยตนเองในทุกที่ และทุกเวลา โดยผ่านระบบจัดการการเรียนรู้หรือที่เรียกว่า LMS (Learning Management System) ที่ทำหน้าที่บริหารและจัดการเกี่ยวกับการสร้างข้อมูลการเรียนรู้ และการเข้าใช้งานข้อมูลการเรียนรู้ของผู้เรียนแต่ละคน อันเป็นการนำไปสู่ระบบการเรียนรู้ที่มีผู้เรียนเป็นศูนย์กลางได้

จากประโยชน์ของระบบ e-learning จึงทำให้หลายสถาบันการศึกษาที่มีความต้องการที่จะมีระบบ e-learning ที่เป็นของตนเอง โดยการพัฒนาขึ้นมาเองหรือการนำเข้าจากต่างประเทศ จึงทำให้มีความหลากหลายของระบบ e-learning เกิดขึ้น เนื่องจากถูกพัฒนาด้วยแพลตฟอร์ม (platform) ที่แตกต่างกัน อันนำไปสู่ปัญหาการขาดความยืดหยุ่นในการพัฒนา (flexibility) การทำงานร่วมกัน (interoperability) การแลกเปลี่ยนข้อมูลการเรียนรู้ร่วมกัน (exchangeability) และการนำเนื้อหาที่มีอยู่แล้วกลับมาใช้ใหม่ (reusability) อันทำให้เกิดความซ้ำซ้อนในการพัฒนาข้อมูลการเรียนรู้ของแต่ละ

*งานวิจัยนี้ได้รับทุนสนับสนุนจากสำนักงานกองทุนสนับสนุนการวิจัยแห่งชาติ (สกว.) และสำนักงานคณะกรรมการการอุดมศึกษา (สกอ.)

สถาบัน และนำไปสู่การสูญเสียค่าใช้จ่ายที่สูงมากในการพัฒนาข้อมูลการเรียนรู้

ถึงแม้ว่าในปัจจุบัน จะมีความพยายามในการออกแบบข้อมูลการเรียนรู้ ให้อยู่ในรูปแบบของออบเจกต์การเรียนรู้ (learning object หรือ LO) [1, 2] บนพื้นฐานของ SCORM (Sharable Content Object Reference Model) [3, 4] ที่มีคำอธิบายข้อมูลของ LO ที่เรียกว่า LOM (Learning Object Metadata) [5] เป็นองค์ประกอบ โดยมีวัตถุประสงค์เพื่อให้สามารถสืบค้น LO ได้ตรงกับความต้องการ ให้สามารถแลกเปลี่ยน LO ระหว่าง LMS ที่แตกต่างกันได้และการนำ LO ที่มีผู้สร้างไว้แล้วมาใช้ใหม่เพื่อประกอบกันเป็นชุดวิชาใหม่ในลักษณะของ content package ได้ [6] อย่างไรก็ตามการสร้างข้อมูลการเรียนรู้บนมาตรฐาน SCORM จำเป็นต้องมีการปรับข้อมูลการเรียนรู้ที่มีอยู่แล้วทั้งหมด ไปสู่มาตรฐานของ SCORM ซึ่งยังต้องใช้เวลาและค่าใช้จ่ายในการพัฒนาคอนข้างสูง [7] อีกทั้งยังมีความยากในการค้นหา LO ที่ถูกเก็บอยู่ในแหล่งเก็บข้อมูลที่หลากหลายบนเครือข่ายอินเทอร์เน็ต ให้สอดคล้องกับความต้องการของผู้ใช้

งานวิจัยนี้ ได้นำเสนอสถาปัตยกรรมการบูรณาการระบบการเรียนรู้ e-learning โดยวิธีคำอธิบายข้อมูลหรือเมตาดาต้า (metadata) และโมเดลตัวบริการเว็บ (Web services) [8, 9] เนื่องจากคุณลักษณะที่โดดเด่นของเทคโนโลยีตัวบริการเว็บ ที่ถูกพัฒนาให้เป็นมาตรฐานเปิด ที่สนับสนุนให้โปรแกรมประยุกต์ที่แตกต่างกัน สามารถทำงานร่วมกันได้ เทคโนโลยีตัวบริการเว็บจึงสามารถถูกนำมาประยุกต์ใช้งานให้เข้ากับระบบ e-learning ได้ สถาปัตยกรรมที่ได้นำเสนอนี้ จะเปิดโอกาสให้ผู้ใช้งานที่อาจเป็น ผู้พัฒนาข้อมูลการเรียนรู้ ผู้สอน หรือผู้เรียน ที่มีภาระลงทะเบียน ไว้กับระบบบูรณาการข้อมูลการเรียนรู้ สามารถเรียกใช้บริการต่างๆ จากตัวบริการเว็บที่ผู้ให้บริการในแต่ละ LMS จัดเตรียมไว้ให้ เช่น การสืบค้นข้อมูลการเรียนรู้ได้ นอกจากนี้ผู้ใช้งานที่เป็นสมาชิกของระบบบูรณาการ ก็ไม่จำเป็นต้องทำการสมัครเป็นสมาชิกใน LMS แต่ละที่ เพื่อขอสืบค้นหรือเข้าใช้งานข้อมูลการเรียนรู้ในแต่ละ LMS จึงเป็นการอำนวยความสะดวกให้กับผู้ใช้งาน ที่สามารถเข้าสู่ระบบเพียงครั้งเดียว (single sign-on) แต่สามารถค้นหาข้อมูลการเรียนรู้ที่ต้องการได้จากทุก LMSs

สถาปัตยกรรมนี้ยังประกอบด้วย เมตาดาต้าที่กำกับข้อมูลการลงทะเบียนบริการต่างๆจากผู้ให้บริการตัวบริการเว็บ โดยมีวัตถุประสงค์เพื่อช่วยในการค้นหาบริการต่างๆจากผู้ให้บริการ และการแก้ปัญหาความขัดแย้งกันของข้อมูลการบริการต่างๆ (semantic conflicts of service data) ของผู้ให้บริการที่หลากหลายกันด้วย ทั้งนี้โครงสร้างของเมตาดาต้าจะถูกแทนด้วยภาษา XML [10] เพื่อใช้เป็นข้อมูลในการสร้างข้อความ SOAP [11] ในการเรียกใช้บริการ หรือรับผลลัพธ์ของบริการจากตัวบริการเว็บในแต่ละแห่ง

งานวิจัยนี้จะประกอบด้วย ส่วนแรกเป็นการกล่าวถึงงานวิจัยอื่นที่เกี่ยวข้อง ส่วนที่สองเป็นการออกแบบสถาปัตยกรรมของระบบบูรณาการข้อมูลการเรียนรู้ และองค์ประกอบต่างๆของระบบ ส่วนที่สามเป็นการออกแบบโครงสร้างเมตาดาต้า และการแทนที่เมตาดาต้าด้วยภาษา XML ส่วนที่สี่จะเป็นการแสดงขั้นตอนการสืบค้นข้อมูลการเรียนรู้ผ่านระบบบูรณาการข้อมูล ส่วนที่ห้าเป็นผลการวิจัยเพื่อทดสอบความถูกต้องของสถาปัตยกรรมที่นำเสนอ อันนำไปสู่การประยุกต์ใช้งานจริงของระบบบูรณาการ และส่วนสุดท้ายจะเป็นบทสรุปและงานวิจัยในอนาคต

2. งานวิจัยที่เกี่ยวข้อง

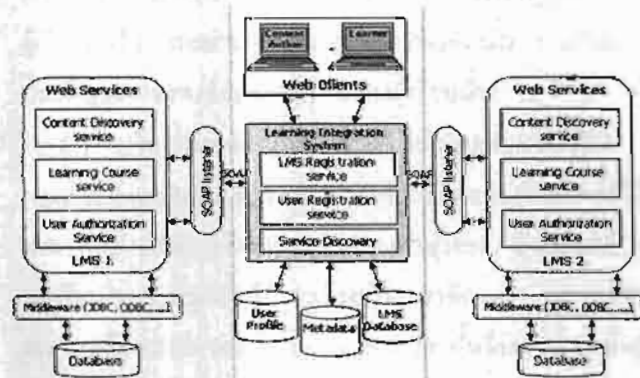
หลายงานวิจัยมีความพยายามที่จะทำให้ระบบ e-learning ที่แตกต่างกัน สามารถแลกเปลี่ยนข้อมูลการเรียนรู้ระหว่างกันได้ ตัวอย่างเช่น งานวิจัยของ [7] ได้นำเสนอสถาปัตยกรรมทั่วไปในการนำเข้าและส่งออกแฟกเกจเนื้อหา บนมาตรฐานของ SCORM 1.2 ในรูปแบบของ PIF (Package Interchange File) ระหว่างฐานข้อมูลเชิงสัมพันธ์ของระบบ e-learning ที่แตกต่างกัน โดยมีส่วนประกอบในระดับกลางของสถาปัตยกรรมทำหน้าที่ค้นหาทรัพยากรการเรียนรู้และทำให้อยู่ในรูปของหน่วยการเรียนรู้ที่เรียกว่า SCO และ Assets เพื่อเก็บอยู่ในแฟกเกจเนื้อหาต่อไป งานวิจัยของ [12] ได้นำเสนอสถาปัตยกรรมที่เรียกว่า LearnServe ที่มีการประยุกต์ใช้งานร่วมกับตัวบริการเว็บ เพื่อให้สามารถสืบค้น และแลกเปลี่ยนออบเจกต์การเรียนรู้ระหว่างระบบ e-learning ได้ หน้าที่หรือกิจกรรมต่างๆของระบบจัดการการเรียนรู้ จะถูกแบ่งออกเป็นกลุ่มของบริการต่างๆ นอกจากนี้จะมีการลงทะเบียนข้อมูลการเรียนรู้ที่เก็บอยู่ในแหล่งเก็บข้อมูลต่างๆ และสร้างคำอธิบาย

ข้อมูลการเรียนรู้เหล่านั้นไว้ในแหล่งเก็บข้อมูลกลางของระบบ เพื่อช่วยในการสืบค้นข้อมูลการเรียนรู้เหล่านั้น ให้ได้ตรงตามความต้องการของผู้ใช้ และสามารถแลกเปลี่ยนข้อมูลการเรียนรู้ร่วมกันได้

ถึงแม้ว่างานวิจัยส่วนใหญ่ จะมุ่งเน้นไปที่การสร้างสถาปัตยกรรมหรือกรอบงานที่ทำให้ระบบ e-learning สามารถทำงานร่วมกันได้ และแลกเปลี่ยนข้อมูลการเรียนรู้ระหว่างกันได้ แต่ยังไม่มียานวิจัยใดที่มุ่งเน้นการแก้ปัญหาความขัดแย้งกันในทางความหมายของข้อมูลการบริการ จากผู้ให้บริการที่หลากหลาย ซึ่งยังเป็นปัญหาคriticalที่เกิดขึ้นในปัจจุบัน

3. สถาปัตยกรรมการทำงานร่วมกันของระบบ e-learning

ในหัวข้อนี้จะนำเสนอสถาปัตยกรรมการทำงานร่วมกันของระบบ e-learning ที่มีความแตกต่างกัน ให้สามารถสืบค้นและแลกเปลี่ยนข้อมูลการเรียนรู้ จากแหล่งเก็บข้อมูลของระบบ LMS ที่แตกต่างกันได้ โดยผ่านระบบบูรณาการข้อมูลการเรียนรู้เพียงที่เดียว สถาปัตยกรรมนี้จะประกอบด้วยสี่องค์ประกอบด้วยกันดังภาพที่ 1 ได้แก่ ผู้ใช้งาน (Web clients) ตัวบริการเว็บ (Web services) ระบบบูรณาการข้อมูลการเรียนรู้ (Learning Integration System-LIS) และแหล่งเก็บข้อมูล (data repository) ต่างๆ เช่น แหล่งเก็บข้อมูลคำอธิบายข้อมูลบริการหรือเมตาดาต้า (metadata) แหล่งเก็บข้อมูลผู้ให้บริการ (user profile) และแหล่งเก็บข้อมูลผู้ให้บริการ (LMS database) เป็นต้น ดังรายละเอียดดังต่อไปนี้



ภาพที่ 1: สถาปัตยกรรมการบูรณาการระบบ e-learning โดยวิธีเมตาดาต้าและโมเดลตัวบริการเว็บ

3.1 ผู้ใช้งานทั่วไป ทำหน้าที่เป็นผู้เรียกใช้บริการ (Service requesters) ได้แก่ ผู้พัฒนาข้อมูลการเรียนรู้ ผู้สอน และผู้เรียน ที่สมัครเป็นสมาชิกของระบบบูรณาการข้อมูลการเรียนรู้ เพื่อเรียกใช้บริการการสืบค้นข้อมูลการเรียนรู้จากตัวบริการเว็บของระบบ LMS ต่างๆที่เป็นสมาชิกของระบบบูรณาการ โดยผู้เรียกใช้บริการไม่จำเป็นต้องทราบว่าข้อมูลการเรียนรู้ถูกเก็บอยู่ใน LMS ใด

3.2 ระบบบูรณาการข้อมูลการเรียนรู้

ระบบบูรณาการข้อมูลการเรียนรู้ หรือเรียกสั้นๆว่าระบบ LIS เป็นระบบงานกลางที่ทำหน้าที่เป็น ตัวแทนผู้ให้บริการ (Service Brokers) โดยจะเป็นตัวกลางที่เชื่อมการทำงานระหว่างผู้ใช้งานและผู้ให้บริการ ซึ่งจะมีการให้บริการในส่วน ดังต่อไปนี้

(1) บริการการลงทะเบียนผู้ให้บริการข้อมูลการเรียนรู้ (LMS Registration service) เป็นบริการที่ผู้ให้บริการข้อมูลการเรียนรู้เช่น ผู้บริหารและจัดการระบบ LMS ต่างๆสามารถเข้ามาลงทะเบียน เพื่อสมัครเป็นสมาชิกของระบบ LIS และประกาศบริการต่างๆที่มีอยู่ในระบบ LMS นั้น เช่น บริการการสืบค้นข้อมูลการเรียนรู้ตามชื่อวิชา คำสำคัญ หรือชื่ออาจารย์ผู้สอน เป็นต้น ซึ่งจะต้องมีการบันทึกข้อมูลชื่อบริการ (service names) ชื่อเมธอด (method) พารามิเตอร์ของข้อมูลนำเข้า (input parameters) พารามิเตอร์ของข้อมูลผลลัพธ์ (output parameters) และตำแหน่งที่อยู่ URL ของไฟล์ WSDL [13] ที่สร้างโดยผู้ให้บริการ LMS นั้น ข้อมูลบริการเหล่านี้จะถูกเก็บบันทึกไว้ในเมตาดาต้า เพื่อใช้ในการสืบค้นข้อมูลบริการต่อไป นอกจากนี้แล้วผู้ให้บริการยังต้องบันทึกรหัสการเข้าสู่ระบบ (login name) ของ LMS นั้น และรหัสผ่าน (password) ไว้ในฐานข้อมูล LMS ด้วย ซึ่งเป็นรหัสที่ยอมให้สมาชิกของระบบ LIS สามารถเข้าสู่บทเรียนของ LMS นั้นได้โดยตรง โดยไม่ต้องมาสมัครเป็นสมาชิกที่ระบบ LMS นั้นอีก

(2) บริการการลงทะเบียนผู้ใช้งานทั่วไป (User registration service) เป็นบริการเกี่ยวกับการลงทะเบียนสมัครเป็นสมาชิกของผู้ใช้งานทั่วไป เพื่อเก็บประวัติของผู้ใช้งานแต่ละคนลงในฐานข้อมูลผู้ใช้งานทั่วไป ก่อนการเข้าใช้งานระบบ

(3) การสืบค้นบริการต่างๆ (Service discovery) เป็นบริการที่รับคำร้องขอเรียกใช้บริการจากผู้ใช้งานและทำการสืบค้น

บริการจากผู้ให้บริการต่างๆที่สอดคล้องกับคำร้องขอ นั้น ทั้งนี้ จะอาศัยข้อมูลที่เกี่ยวข้องในเมตาดาต้า ช่วยในการสืบค้นข้อมูล บริการดังกล่าว เช่น การสืบค้นว่าคำร้องขอนั้นมีอยู่ในบริการ ของ LMS ไດบ้าง ต้องเรียกใช้ชื่อเมธอดใด ต้องส่งพารามิเตอร์ อะไรไปบ้าง หลังจากนั้นจะทำการสร้างข้อความ SOAP เพื่อ ขอเรียกใช้บริการไปยัง LMS เหล่านั้น และรวบรวมผลลัพธ์ที่ สืบค้นได้จากแต่ละ LMS ที่อยู่ในรูปของข้อความ SOAP เช่นกัน มาทำการจัดรูปแบบและส่งกลับไปแสดงแก่ผู้ใช้ต่อไป

3.3 ตัวบริการเว็บ ทำหน้าที่เป็นผู้ให้บริการ (Service providers) ซึ่งได้แก่ ระบบ LMS ของสถาบันการศึกษาต่างๆ บนเครือข่ายอินเทอร์เน็ต ที่มีการลงทะเบีย่นกับระบบ LIS โดยแต่ละระบบ LMS อาจถูกพัฒนาด้วยภาษาโปรแกรมที่แตกต่างกัน บนเครื่องคอมพิวเตอร์ที่มีระบบปฏิบัติการที่ต่างกัน และมีการสร้างตัวบริการเว็บไว้ให้บริการในด้านต่างๆ ดังต่อไปนี้

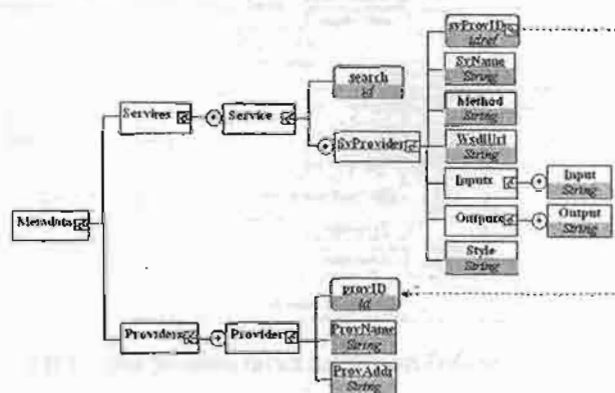
(1) บริการการสืบค้นข้อมูลการเรียนรู้ (content discovery service) เป็นบริการที่รับคำร้องขอการสืบค้นข้อมูล ที่ส่งมาจากระบบ LIS ที่อยู่ในรูปแบบของข้อความ SOAP และทำการสืบค้นข้อมูลการเรี ยนรู้ นั้นจากฐานข้อมูลการเรียนรู้ โดยผ่านตัวกลางอินเตอร์เฟส เช่น JDBC (Java Database Connection) หรือ ODBC (Open Database Connectivity) เป็นต้น และในทางกลับกันก็จะรวบรวมผลลัพธ์ที่สืบค้นได้ เช่น ชื่อวิชา ตำแหน่ง URL ของวิชานั้น และคำอธิบายวิชาโดยย่อ ส่งกลับไปยังระบบ LIS ในรูปแบบของข้อความ SOAP เช่นกัน

(2) บริการจัดการข้อมูลการเรียนรู้ (learning course service) เป็นบริการที่ช่วยให้ผู้เรียนสามารถเข้าสู่บทเรียนที่ผู้เรียนสนใจ และนำเสนอเนื้อหาของบทเรียนนั้น รวมทั้งการบ้านที่กและรายงานข้อมูลการเรียนรู้ของผู้เรียนนั้น

(3) บริการตรวจสอบสิทธิการเข้าใช้งานระบบของผู้ใช้ (user authorization service) เป็นบริการที่ตรวจสอบรหัสผู้ใช้ และรหัสผ่านที่มีการล็อกอิน (login) เข้าสู่ระบบ ซึ่งผู้ที่จะเข้าระบบเรียนใน LMS นั้นได้ จะต้องเป็นสมาชิกของระบบ LMS เอง หรือสมาชิกของระบบ LIS เท่านั้นจึงจะมีสิทธิเข้าระบบเรียนของ LMS นั้นได้

4. โครงสร้างเมตาดาต้า (Metadata Structure)

เมตาด้าถูกออกแบบให้เป็นองค์ประกอบหลักของสถาปัตยกรรม เพื่อทำหน้าที่เก็บข้อมูลการให้บริการของแต่ละ LMS ซึ่งช่วยในการค้นหาบริการและแก้ปัญหาความขัดแย้งในเชิงความหมายของข้อมูลบริการ เช่น บริการในแต่ละ LMS อาจมีชื่อบริการ ชื่อเมทอด หรือชื่อพารามิเตอร์ที่แตกต่างกันไป แต่มีความหมายที่เหมือนกัน (synonymous service data) การออกแบบโครงสร้างเมตาด้าจะแทนด้วยภาษา XML ที่อธิบายโครงสร้างได้ด้วยโครงสร้างแบบต้นไม้ (tree structure) ที่แสดงในภาพที่ 2 ที่ประกอบด้วยข้อมูลบริการต่างๆของผู้ให้บริการ และรายละเอียดของผู้ให้บริการแต่ละแห่ง



ภาพที่ 2 แสดงโครงสร้างเมตาดาต้า

โดยรูปสี่เหลี่ยมผืนผ้าแต่ละรูปจะแทนอิลิเมนต์ (Element) ของ XML ส่วนรูปสี่เหลี่ยมผืนผ้าที่มีขอบบนและมีเส้นขอบสองเส้นจะแทนด้วยแอททริบิวต์ของ XML บริเวณสีขาวภายในรูปสี่เหลี่ยมผืนผ้า จะเก็บชื่ออิลิเมนต์หรือแอททริบิวต์ ส่วนบริเวณที่มีการแรเงา จะแสดงประเภทข้อมูลที่แต่ละอิลิเมนต์หรือแอททริบิวต์นั้นเก็บ เครื่องหมายวงเล็บซ้อนกันจะแสดงว่า อิลิเมนต์นั้นมีอิลิเมนต์ย่อยภายในอีก เครื่องหมายบวกที่อยู่ข้างหน้าอิลิเมนต์ จะแสดงว่าอิลิเมนต์ย่อยที่อยู่หลังเครื่องหมายบวกนี้ จะมีตั้งแต่หนึ่งอิลิเมนต์ขึ้นไป อิลิเมนต์ย่อยที่ไม่มีเครื่องหมายบวกอยู่ข้างหน้าแสดงว่า จะมีอิลิเมนต์ย่อยนั้นเพียงหนึ่งอิลิเมนต์เท่านั้น แอททริบิวต์ที่ทำหน้าที่เป็นตัวชี้เฉพาะ (unique identifier) จะถูกกำหนดให้เป็น *id* และแอททริบิวต์ที่ทำหน้าที่เป็นตัวอ้างอิง (identifier reference) ถึงตัวชี้เฉพาะนั้น จะถูกกำหนดให้เป็น *idref* ซึ่งสามารถแสดงการอ้างอิงด้วยเส้นลูกศรประจํา ตัวอย่างของโครงสร้างเมตาภาษาคำที่

แทนด้วย XML-DTD ที่สอดคล้องกับโครงสร้างในภาพที่ 2 จะแสดงอยู่ในภาพที่ 3

```
<?xml version="1.0" encoding="UTF-8"?>
<ELEMENT Metadata (Services, Providers)>
<ELEMENT Services (Service)+>
<ELEMENT Service (SvProvider)+>
<ATTLIST Service search ID #REQUIRED>
<ELEMENT SvProvider (SvName, Method, WsdUrl, Inputs, Outputs, Style)>
<ATTLIST SvProvider svProvID IDREF #REQUIRED>
<ELEMENT SvName (#PCDATA)>
<ELEMENT Method (#PCDATA)>
<ELEMENT WsdUrl (#PCDATA)>
<ELEMENT Style (#PCDATA)>
<ELEMENT Inputs (Input)+>
<ELEMENT Input (#PCDATA)>
<ELEMENT Outputs (Output)+>
<ELEMENT Output (#PCDATA)>
<ELEMENT Providers (Provider)+>
<ELEMENT Provider (ProvName, ProvAddr)>
<ATTLIST Provider provID ID #REQUIRED>
<ELEMENT ProvName (#PCDATA)>
<ELEMENT ProvAddr (#PCDATA)>
```

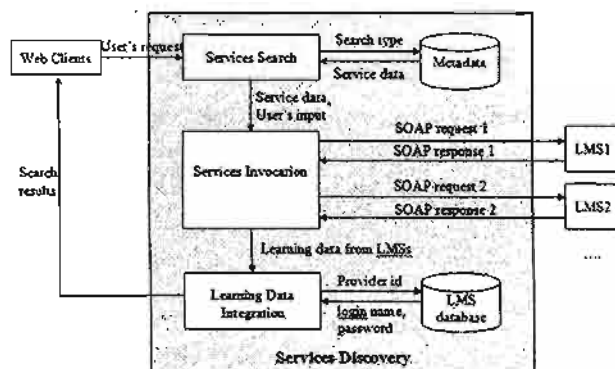
ภาพที่ 3 แสดงโครงสร้างเมตาดาต้าที่แทนด้วย XML-DTD

ความหมายของอิลิเมนต์และแอททริบิวต์ต่างๆในเมตาดาต้า มีดังต่อไปนี้ อิลิเมนต์ *Metadata* เป็นอิลิเมนต์ root ของเอกสาร ซึ่งประกอบด้วยอิลิเมนต์ย่อยสองอิลิเมนต์คือ *Services* และ *Providers* โดยอิลิเมนต์ *Services* เป็นอิลิเมนต์ที่ประกอบด้วยอิลิเมนต์ย่อยคือ *Service* ตั้งแต่หนึ่งอิลิเมนต์ขึ้นไป ที่ประกอบด้วยแอททริบิวต์ *search* ที่ทำหน้าที่เก็บข้อมูลประเภทการค้นหาบริการในดวับริการเว็บของแต่ละ LMS เช่น บริการการสืบค้นบทเรียนแบบระบุคำสำคัญ (keyword) หรือแบบระบุชื่อวิชา (title) เป็นต้น และอิลิเมนต์ย่อยชื่อ *SvProvider* ที่ประกอบด้วยแอททริบิวต์ *svProvID* ที่เก็บหมายเลขอ้างอิงของผู้ให้บริการ และอิลิเมนต์ย่อยอื่นๆที่ทำหน้าที่บอกข้อมูลบริการของผู้ให้บริการแต่ละแห่ง อันได้แก่ ชื่อบริการของผู้ให้บริการ (*SvName*) ชื่อของเมธอดหรือฟังก์ชันที่ให้บริการ (*Method*) ตำแหน่งที่อยู่ของเอกสาร WSDL ของดวับริการเว็บ (*WsdUrl*) พารามิเตอร์ซึ่งข้อมูลเข้า ที่จะต้องส่งไปประมวลผลยังดวับริการเว็บ (*Inputs*) พารามิเตอร์ข้อมูลผลลัพธ์ . ที่ได้จากการประมวลผลจากดวับริการเว็บ (*Outputs*) และ รูปแบบของการติดต่อกับดวับริการเว็บ (*Style*) ซึ่งมีหลายรูปแบบ เช่น document style, rpc style และ rpc style แบบมี complex type เป็นต้น ทั้งนี้ แอททริบิวต์ *svProvID* จะเก็บหมายเลขอ้างอิงของผู้ให้บริการ ที่มีการประกาศเป็น 'idref' ซึ่งอ้างอิงไปยังแอททริบิวต์ *provID* ของอิลิเมนต์ *Provider* ที่มีการประกาศเป็น 'id' และสำหรับอิลิเมนต์ *Providers* เป็นอิลิเมนต์ที่เก็บอิลิเมนต์ย่อย *Provider* ซึ่งทำหน้าที่เก็บรายละเอียดของผู้ให้บริการแต่ละแห่ง เช่น หมายเลขอ้างอิงของผู้ให้บริการ (*ProvID*) ชื่อผู้ให้บริการ (*ProvName*) และ ที่อยู่ผู้ให้บริการ (*ProvAddr*) เป็นต้น

อิลิเมนต์ *Providers* เป็นอิลิเมนต์ที่เก็บอิลิเมนต์ย่อย *Provider* ซึ่งทำหน้าที่เก็บรายละเอียดของผู้ให้บริการแต่ละแห่ง เช่น หมายเลขอ้างอิงของผู้ให้บริการ (*ProvID*) ชื่อผู้ให้บริการ (*ProvName*) และ ที่อยู่ผู้ให้บริการ (*ProvAddr*) เป็นต้น

5. การสืบค้นข้อมูลการเรียนรู้

ในการสืบค้นข้อมูลการเรียนรู้จากผู้ให้บริการหรือ LMS จากที่ต่างๆ ที่มีการลงทะเบียนไว้กับระบบ LIS จะสามารถสืบค้นโดยอาศัยข้อมูลจากเมตาดาต้า ซึ่งสามารถแสดงแผนภาพการทำงานของบริการการสืบค้นข้อมูลการเรียนรู้ (service discovery) ได้ดังภาพที่ 4



ภาพที่ 4: แผนภาพการทำงานในส่วนบริการการสืบค้นข้อมูลการเรียนรู้

จากภาพที่ 4 สามารถอธิบายการทำงานได้ดังต่อไปนี้

(1) การค้นหาบริการ (Services search) เป็นส่วนค้นหาข้อมูลบริการที่เก็บอยู่ในเมตาดาต้า โดยการทำงานในส่วนนี้จะมีการรับคำร้องขอในการสืบค้นข้อมูลการเรียนรู้จากผู้ใช้งานมา อันได้แก่ ประเภทการค้นหา (search type) เช่น การสืบค้นแบบระบุคำสำคัญ ชื่อวิชา หรือ ชื่ออาจารย์ เป็นต้น รวมทั้งข้อความที่ผู้ใช้งานต้องการค้นหา (user's input) เพื่อนำข้อมูลคำร้องขอไปค้นหาบริการในเมตาดาต้าว่ามี LMS ใดบ้างที่ให้บริการเกี่ยวกับการค้นหาข้อมูลการเรียนรู้ตามประเภทการค้นหาที่ผู้ใช้ระบุ และทำการรวบรวมข้อมูลบริการต่างๆที่สืบค้นได้ เช่น หมายเลขอ้างอิงผู้ให้บริการ, ชื่อบริการ, URL ของเอกสาร WSDL, เมธอดที่จะเรียกใช้บริการ, พารามิเตอร์ข้อมูลเข้าและออก, และ รูปแบบของการติดต่อกับดวับริการเว็บ เป็นต้น

(2) การเรียกใช้บริการจากผู้ให้บริการ (Services invocation) เป็นส่วนที่ทำหน้าที่รับข้อมูลบริการที่สืบค้นได้จากเมตาดาต้าและข้อมูลร้องขอจากผู้ใช้งาน มาสร้างเป็นข้อความ

SOAP เพื่อเรียกใช้บริการจากผู้ให้บริการที่เป็น LMS ต่างๆที่เกี่ยวข้อง และรอรับผลลัพธ์ที่สืบค้นได้จากแต่ละ LMS ที่อยู่ในรูปแบบของข้อความ SOAP เช่นกัน อันได้แก่ ชื่อวิชา ตำแหน่ง URL ของวิชานั้น และคำอธิบายเนื้อหาวิชาโดยย่อ เป็นต้น ซึ่งเป็นผลลัพธ์ที่สอดคล้องกับพารามิเตอร์ข้อมูลผลลัพธ์ (output parameters) จากเมตาดาต้า และทำการนำส่งข้อมูลผลลัพธ์นี้ไปยังส่วนการบูรณาการข้อมูลต่อไป

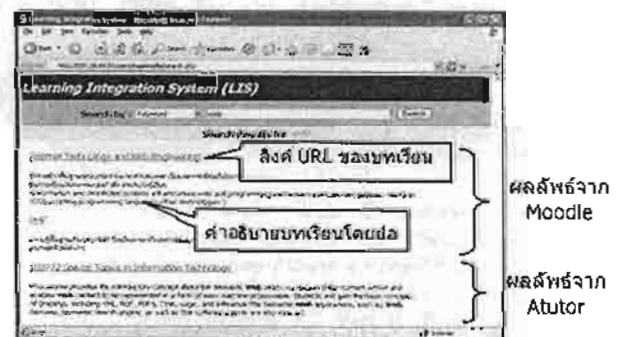
(3) การบูรณาการข้อมูลการเรียนรู้ (Learning data integration) เป็นส่วนที่ทำหน้าที่รวบรวมข้อมูลผลลัพธ์จาก LMS ต่างๆที่ส่งมาจากส่วนการเรียกใช้บริการที่อยู่ในรูปแบบของข้อความ SOAP มาทำการจัดรูปแบบของผลลัพธ์ให้อยู่ในรูปแบบของเอกสาร html เพื่อแสดงผลแก่ผู้ใช้ โดยสร้างลิงค์ (link) เชื่อมโยงชื่อวิชากับตำแหน่ง URL ของวิชาในแต่ละ LMS ให้บริการ รวมทั้งการแสดงผลคำอธิบายรายวิชานั้น และเพื่อให้ผู้ใช้ที่เป็นสมาชิกของระบบ LIS สามารถเข้าสู่บทเรียนของ LMS ต่างๆได้โดยตรงเมื่อมีการคลิกที่ลิงค์ของบทเรียนใน LMS ใดๆ โดยไม่ต้องเข้าไปสมัครเป็นสมาชิกของ LMS ใหม่ ในส่วนของการบูรณาการข้อมูลการเรียนรู้ จะต้องมีการค้นหารหัสผู้ใช้และรหัสผ่านที่ถูกเข้ารหัส (encrypt) ไว้จากฐานข้อมูล LMS ที่ระบบ LMS แต่ละระบบได้มาลงทะเบียนไว้กับระบบ LIS ซึ่งเป็นรหัสผู้ใช้ที่ระบบ LMS นั้นๆอนุญาตหรือให้สิทธิในการเข้าสู่บทเรียนต่างๆใน LMS นั้นได้ ข้อมูลรหัสผู้ใช้และรหัสผ่านนี้จะถูกนำไปเชื่อมต่อกับตำแหน่ง URL ของวิชาที่ถูกสร้างเป็นลิงค์ไว้ก่อนหน้านี้แล้ว และเมื่อผู้ใช้มีการคลิกที่ลิงค์ของบทเรียนนั้น ข้อมูลรหัสผู้ใช้และรหัสผ่านนี้จะถูกส่งไปยังระบบ LMS นั้นโดยผ่านส่วนบริการการตรวจสอบสิทธิการเข้าใช้งานระบบ LMS ของผู้ใช้ ซึ่งด้านเป็นสมาชิกของระบบ LIS นี้ก็จะเข้าสู่บทเรียนนั้นได้โดยอัตโนมัติ

6. ผลการวิจัย

งานวิจัยนี้ได้พัฒนาระบบ LIS ที่เป็นระบบงานกลางที่เชื่อมการทำงานระหว่างผู้ใช้ทั่วไป และผู้ให้บริการที่เป็นระบบ LMS ที่แตกต่างกัน โดยระบบ LIS จะประกอบด้วยบริการต่างๆ ดังที่กล่าวไว้ในหัวข้อ 3.2 ซึ่งถูกพัฒนาด้วยภาษา PHP บนเว็บเซิร์ฟเวอร์ Apache และใช้ฐานข้อมูล MySQL สำหรับเก็บฐานข้อมูล LMS และ ข้อมูลสมาชิกในระบบ LIS และเพื่อเป็นการทดสอบว่าระบบ LMS ที่แตกต่างกันจะสามารถทำงาน

ร่วมกันได้โดยผ่านตัวบริการเว็บ จึงได้ทำการสร้างตัวบริการเว็บไว้กับระบบ LMS ที่แตกต่างกัน อันได้แก่ LMS ที่ถูกพัฒนาด้วยภาษา PHP และมีการสร้างตัวบริการเว็บด้วย PHP nusoap เช่น ระบบ Moodle [14] และระบบ Atutor [15] ที่ติดต่อกับฐานข้อมูล MySQL นอกจากนี้ยังมีระบบ LMS ที่ถูกสร้างด้วย ASP โดยมีการสร้างตัวบริการเว็บด้วย ASP.NET ที่ติดต่อกับฐานข้อมูล Access

จากการทดสอบการทำงานของระบบ จะสามารถแสดงผลลัพธ์ของการเรียกใช้บริการการสืบค้นข้อมูลการเรียนรู้ของระบบ LIS ดังแสดงในภาพที่ 5 โดยผู้ใช้มีการเลือกประเภทการสืบค้นแบบระบุคำสำคัญ (keyword) พิมพ์ข้อความที่ต้องการสืบค้นว่า “Web” เมื่อกดปุ่ม Search ระบบ LIS จะสืบค้นบริการการค้นหายาเรียนจากเมตาดาต้า เพื่อค้นหาว่ามี LMS ใดให้บริการการสืบค้นด้วยคำสำคัญนี้บ้าง และทำการสร้างข้อความ SOAP เพื่อขอเรียกใช้บริการจาก LMS เหล่านั้น



ภาพที่ 5: หน้าจอแสดงผลจากการสืบค้นบริการข้อมูลการเรียนรู้



ภาพที่ 6: หน้าจอแสดงการเข้าสู่บทเรียนของระบบ LMS Moodle

ผลลัพธ์ที่ได้จากการสืบค้นในแต่ละ LMS จะถูกนำมารวบรวมและจัดรูปแบบการแสดงผลที่ประกอบด้วยชื่อวิชา คำอธิบายรายวิชาแบบย่อ และลิงค์ที่มีการเชื่อมโยงตำแหน่ง URL ของวิชานั้นเข้ากับชื่อวิชา ซึ่งจะมีการแทรกรหัสผู้ใช้และรหัสผ่านสำหรับระบบ LMS นั้นๆที่มีการเข้ารหัสไว้ เพื่อให้

ผู้ใช้งานสามารถคลิกที่ลิงค์บทเรียนนั้นและเข้าสู่การใช้งานบทเรียนต่างๆของ LMS แต่ที่นั่นได้โดยอัตโนมัติ ดังภาพที่ 6

7. บทสรุปและงานวิจัยในอนาคต

งานวิจัยนี้ได้นำเสนอสถาปัตยกรรมการบูรณาการระบบการเรียนรู้ e-learning ที่สามารถครอบคลุมคุณสมบัติที่สำคัญของระบบ e-learning อันได้แก่ ความสามารถในการค้นหาข้อมูลการเรียนรู้ได้จากทุกที่ (accessability) การทำให้ LMS ของแต่ละหน่วยงานที่ถูกพัฒนา โดยต่างแพลตฟอร์มกัน สามารถติดต่อสื่อสาร แลกเปลี่ยนทรัพยากรกัน และทำงานร่วมกันได้ (interoperability) การนำเสนอเมตาดาต้าซึ่งเป็นองค์ประกอบสำคัญของระบบ LIS เพื่อเก็บคำอธิบายข้อมูลบริการจากผู้ให้บริการหรือ LMSs ต่างๆ ซึ่งนอกจากจะช่วยให้สามารถสืบค้นบริการจาก LMS ต่างๆได้พร้อมกันโดยตรง และได้ข้อมูลที่เป็นปัจจุบันแล้ว ยังช่วยแก้ปัญหาความขัดแย้งกัน ทางด้านความหมายของข้อมูลบริการจากตัวบริการเว็บต่างๆ อีกทั้งยังเป็นการช่วยลดภาระงานของผู้สร้างเนื้อหา ที่ไม่ต้องสร้างเนื้อหาขึ้นมาใหม่ทั้งหมดแต่สามารถใช้เนื้อหาที่สืบค้นได้จากผู้สร้างไว้แล้ว อันเป็นการสนับสนุนหลักการการนำกลับมาใช้ใหม่ (reusability) งานวิจัยนี้ยังอำนวยความสะดวกให้แก่ผู้ใช้ที่เป็นสมาชิกของระบบ LIS ให้สามารถล็อกอินเพียงครั้งเดียว แต่สามารถเข้าสู่ระบบ LMS แต่ละแห่งได้โดยอัตโนมัติ นอกจากนี้การแทนโครงสร้างเมตาดาต้าด้วยภาษา XML ทำให้มีความยืดหยุ่น (flexibility) ต่อการจัดการโครงสร้างเมตาดาต้า และการเข้าใจความหมายของข้อมูลร่วมกันระหว่างระบบงานที่แตกต่างกัน

อย่างไรก็ตาม ในการสืบค้นข้อมูลโดยระบบงาน LIS นี้ อาจมีความล่าช้าบ้างในการสืบค้น เนื่องจากเป็นการสืบค้นจากแหล่งข้อมูลในระบบ LMS แต่ละแห่งโดยตรง นอกจากนี้ เพื่อเพิ่มประสิทธิภาพ ในการสืบค้นข้อมูลแบบสื่อความหมายให้มากยิ่งขึ้น และสามารถแก้ปัญหาคความหลากหลายของข้อมูลในเชิงความหมาย (semantic heterogeneity) ได้อย่างแท้จริง โครงสร้างเมตาดาต้ายังสามารถถูกปรับปรุง โดยการแทนด้วยภาษาออนโทโลยี เช่น RDF [16] หรือ OWL [17] ได้ อันนำไปสู่ความเข้าใจความหมายของข้อมูลร่วมกัน ของคอมพิวเตอร์และโปรแกรมประยุกต์ที่แตกต่างกันได้ยิ่งขึ้นต่อไป (machine-understandable Semantic Web) [18]

8. เอกสารอ้างอิง

- [1] T. Boyle, "Design principles for authoring dynamic, reusable learning objects," *Australian Journal of Educational Technology*, vol. 19 (1), pp. 46-58.
- [2] D. Wiley, "Learning Object Design and Sequencing Theory," *Doctoral Dissertation, Dept. of Instructional Psychology and Technology*, Brigham Young University, 2000.
- [3] Advanced Distributed Learning (ADL) (2004), "SCORM® 2nd Edition Overview," available from <http://www.adlnet.org>.
- [4] ADL Technical Team (2003), "SCORM Specification v1.2," available from <http://www.adlnet.org/index.cfm?fuseaction=scormabt>.
- [5] IEEE P1484.12, "Learning Object Metadata (LOM) Standard", Available from: <http://ltsc.ieee.org/wg12/>.
- [6] Advanced Distributed Learning (ADL) (2004), "SCORM Content Aggregation Model Version 1.3.1," available from <http://www.adlnet.org/>.
- [7] P. Arapi, N. Moutoutzis, and S. Christodoulakis, "Supporting Interoperability in an Existing e-Learning Platform using SCORM," *In the proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies (ICALT 2003)*, Greece, July 2003.
- [8] W3C, "Web Services Architecture," available from: <http://www.w3.org/TR/ws-arch/>, W3C Working Group Note [11-Feb-2004].
- [9] W. Hoschek, "The Web Service Discovery Architecture," *Proceeding of the IEEE/ACM Supercomputing Conference (SC 2002)*, Maryland, pp.1-15.
- [10] W3C, "World Wide Web Consortium, Extensible Markup Language (XML) 1.0 (Second Edition)," available from: <http://www.w3.org/TR/2000/REC-xml-20001006/>, W3C Recommendation [6-Oct-2000].
- [11] W3C, "SOAP version 1.2 Part 0: Primer," available from: <http://www.w3.org/TR/soap1.2-part0/>, W3C Recommendation [24-Jun-2003].
- [12] G. Vossen and P. Westerkamp, "Maintenance and Exchange of Learning Objects in a Web Services Based e-Learning System," *Electronic Journal of e-Learning*, vol. 2 (2), pp. 292-304, 2004.
- [13] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL) 1.1," available from: <http://www.w3.org/TR/wsdl/>, W3C Note [15-Mar-2001].
- [14] Moodle, available at <http://moodle.org/> or <http://www.thaimoodle.com>.
- [15] ATutor – Learning Content Management System, available at <http://www.atutor.ca/>.
- [16] Frank Manola and Eric Miller, RDF Primer. W3C Recommendation. WWW, available at: <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>, 10 February 2004.
- [17] Harmelen F. V., Hendler J., Horrocks I., Stein L. A., McGuinness D. L., Patel-Schneider P. F., Web Ontology Language (OWL) Reference Version 1.0. W3C Working Draft, available at <http://www.w3.org/TR/owl-ref/>, February 2003.
- [18] Decker S., Melnik S., Van H. F., Fensel D., Klein M., Broekstra J., Erdmann M. and Horrocks I., The Semantic Web: The Roles of XML and RDF, *IEEE Internet Computing* 4, September/October, 2000, 63-74.