

บทที่ 2

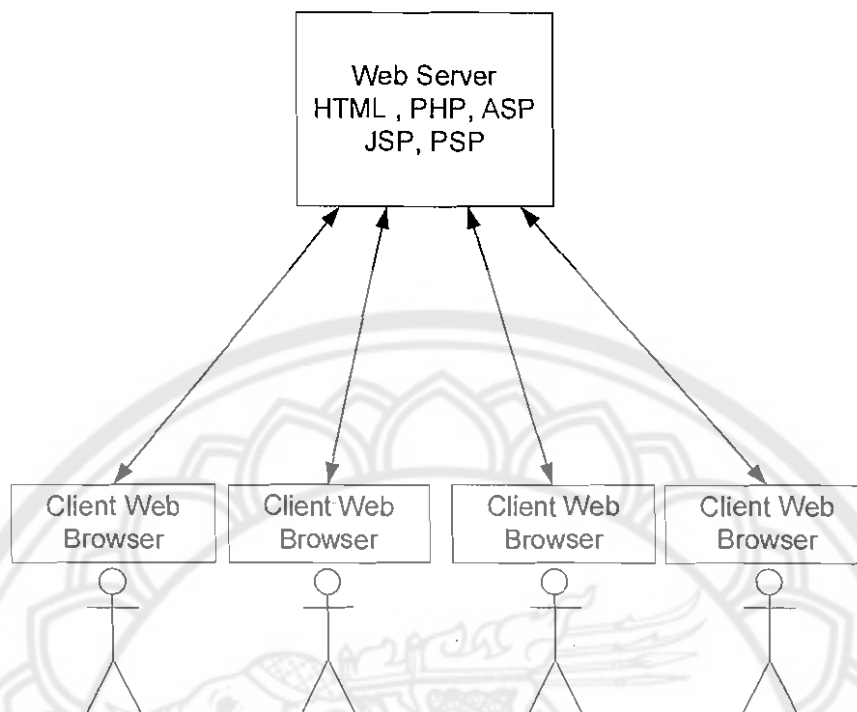
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในงานวิจัยนี้ มีเอกสารและงานวิจัยที่เกี่ยวข้อง ดังนี้

1. ทฤษฎีเว็บเซอริวิส
2. การวิเคราะห์ ออกแบบและพัฒนาเชิงวัตถุ
3. คุณภาพการบริการของเว็บเซอริวิส
4. หลักสถิติที่นำมาใช้ในงานวิจัย
5. งานวิจัยที่เกี่ยวข้อง

ทฤษฎีเว็บเซอริวิส

อินเทอร์เน็ตเป็นสื่อกลางการเผยแพร่ข้อมูลที่กำลังได้รับความนิยมในปัจจุบันผ่านทางเว็บไซต์และภาษาที่ถูกนำมาใช้ในการพัฒนาในลักษณะนี้ซึ่งสนับสนุนให้มีการทำงานได้ทั้งฝั่งเครื่องลูกข่ายและเครื่องแม่ข่าย ได้แก่ VBScript, JavaScript, ASP, PHP, JSP และ PSP เป็นต้น เว็บไซต์ในปัจจุบันมีลักษณะคล้ายแอปพลิเคชันมากขึ้น จึงมีการเรียกเว็บไซต์เหล่านั้นว่า เว็บแอปพลิเคชัน (จัดรัชย์ สุขสะอาด, ม.ป.ป.) ในบรรดาภาษาที่ทำงานบนเว็บนี้ สามารถแบ่งเป็นภาษาที่ทำงานบนฝั่ง Client และภาษาที่ทำงานบนฝั่ง Server ได้ ดังนั้น VBScript และ JavaScript เป็นภาษาที่ทำงานบนฝั่ง Client และ ASP, PHP, JSP และ PSP เป็นภาษาที่ทำงานบนฝั่ง Server เป็นต้น การทำงานของเว็บรูปแบบนี้เป็นการทำงานระหว่างเครื่องเซิร์ฟเวอร์และบุคคลหรือเว็บเบราว์เซอร์ (Web Browser) ดังภาพ 5



ภาพ 5 แสดงรูปแบบการให้บริการของเว็บแอปพลิเคชัน

เว็บเซอร์วิสถูกออกแบบมาให้สนับสนุนการทำงานระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ หรือแอปพลิเคชันกับแอปพลิเคชันผ่านระบบเครือข่าย และติดต่อสื่อสารกันได้โดยใช้ XML และ WSDL เป็นตัวกลางในการติดต่อสื่อสารโดยใช้ SOAP เป็นโปรโตคอลในการติดต่อสื่อสาร ซึ่งเป็นมาตรฐานในการติดต่อสื่อสารระหว่างระบบเครือข่ายโดยผ่านโปรโตคอลต่างๆ ในการส่งข้อมูลบนเว็บไซต์ เช่น HTTP เป็นต้น (จักรชัย สุขสะอาด, ม.ป.ป., ไม่มีเลขหน้า; W3C Working Group, 2003)

ในการพัฒนาเว็บเซอร์วิสนั้นสามารถเลือกที่จะพัฒนาแบบ SOAP หรือแบบ REST ถ้าเลือกพัฒนาแบบ SOAP เว็บเซอร์วิสจะต้องมีการส่งข้อความ XML ตามรูปแบบที่กำหนดไว้โดยโปรโตคอล SOAP อีกทั้งต้องมีเอกสารอธิบายการเรียกใช้เว็บเซอร์วิสประกอบ ซึ่งเอกสารที่อธิบายนี้จะเขียนโดยใช้ภาษา WSDL ในกรณีของผู้เรียกใช้ จะต้องมีการเข้าใจเอกสารที่อธิบายการเรียกใช้ SOAP เว็บเซอร์วิสหรือมีเครื่องมือที่จะเข้าใจและเรียกใช้ได้อย่างถูกต้อง

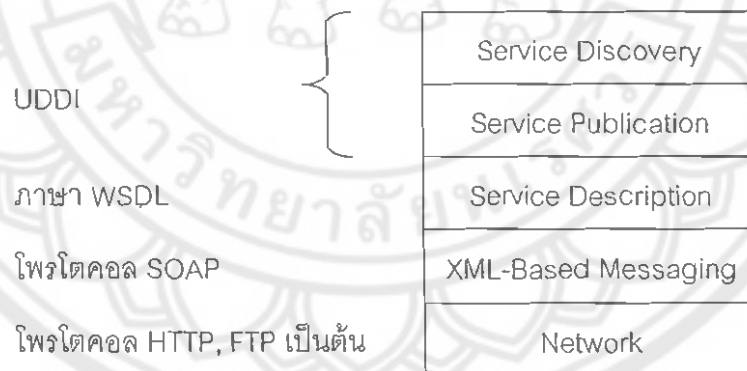
ส่วน REST เว็บเซอร์วิสจะเป็นรูปแบบของซอฟต์แวร์ที่มองว่าข้อมูลต่างๆ เป็นทรัพยากร (Resource) ซึ่งสามารถเรียกใช้ได้ผ่านทางโปรโตคอล HTTP และข้อมูลที่ส่งกลับมาให้ผู้ใช้เป็นข้อมูลรูปแบบ XML ในกรณีของผู้เรียกใช้ REST เว็บเซอร์วิสเพียงทราบ URL ของ REST เว็บ

เซอร์วิสและการอ่านข้อมูล XML ก็จะต้องข้อมูลที่ต้องการได้ (Weerawarana, et al, 2005) ในงานวิจัยนี้ผู้วิจัยจะพัฒนาเว็บเซอร์วิสแบบ SOAP เว็บเซอร์วิส

สถาปัตยกรรมของเว็บเซอร์วิส

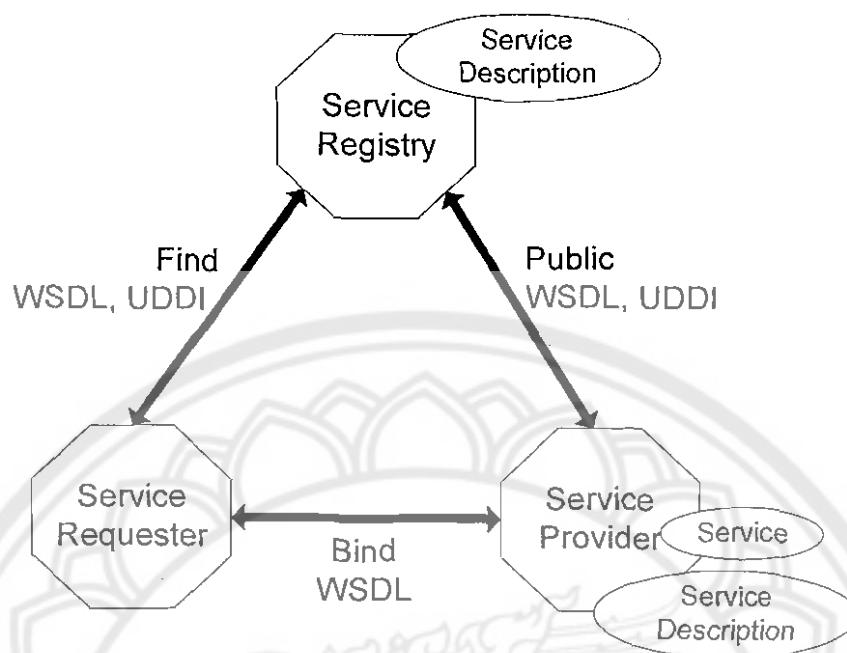
โครงสร้างสถาปัตยกรรมของเว็บเซอร์วิสนั้นจะเน้นการให้บริการเป็นส่วนใหญ่เรียกว่า Service Oriented Architecture (SOA) ประกอบด้วย องค์ประกอบต่างๆ ดังภาพ 6 และแต่ละองค์ประกอบมีความสัมพันธ์ดังภาพ 7

1. ผู้ให้บริการ (Service Provider) ทำหน้าที่ประกาศ (Public) บริการต่างๆ ขององค์กรไปยังไดเรกทอรีที่เก็บทะเบียนการบริการหรือเรียกว่า ตัวแทนผู้ให้บริการ (Service Broker /Repository) โดยผู้ให้บริการต้องลงทะเบียนโดยระบุชื่อบริการและพารามิเตอร์หรือเงื่อนไขต่างๆ ที่สามารถใช้บริการนั้นด้วยมาตรฐาน WSDL เพื่ออธิบายการเรียกใช้งานของโปรแกรมต่างๆ
2. ตัวแทนผู้ให้บริการ (Service Registry) จะจัดเก็บบริการต่างๆ โดยใช้มาตรฐานที่เรียกว่า UDDI ที่ช่วยกำหนดการลงทะเบียนการค้นหา และการเข้าถึงบริการที่อยู่ในไดเรกทอรีได้
3. ผู้เรียกใช้บริการ (Service Requester) คือเครื่องคอมพิวเตอร์ปลายทางที่ต้องการค้นหาบริการจากตัวแทนผู้ให้บริการ เมื่อพบว่าบริการนั้นมีอยู่จริง และอยู่ที่ผู้บริการใดก็จะเรียกใช้บริการ (Blind) ไปยังผู้ให้บริการนั้นโดยผ่านมาตรฐาน WSDL



ภาพ 6 แสดงความสัมพันธ์ระหว่าง WSDL, SOAP และ UDDI

(สราวุธ อ้อยศรีสกุล, 2544)



ภาพ 7 แสดงโครงสร้างสถาปัตยกรรมของเว็บเซอร์วิส
(Weerawarana, et al, 2005)

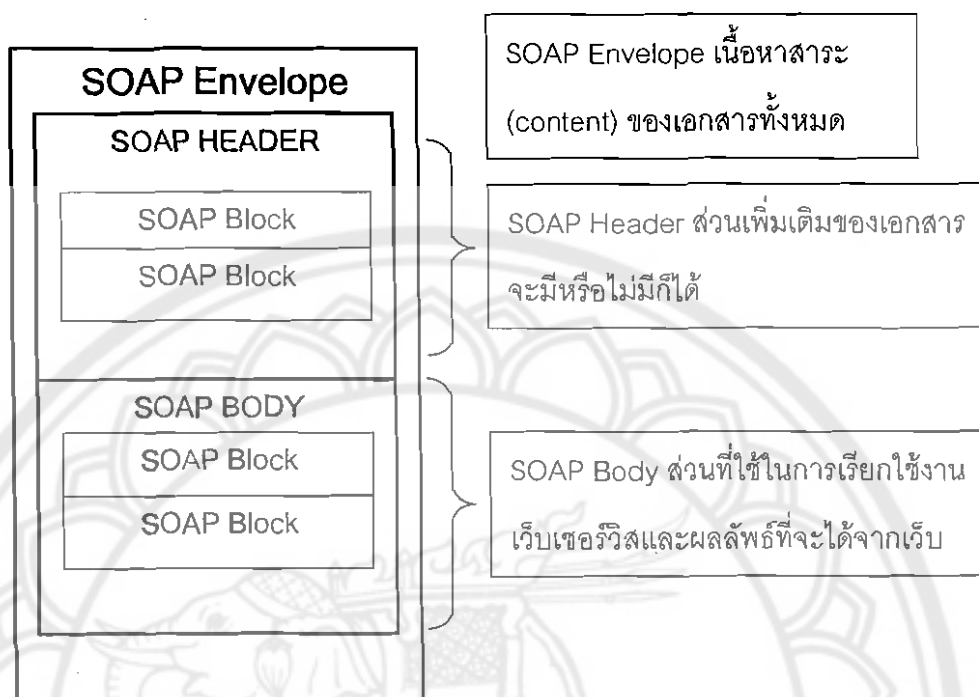
องค์ประกอบของเว็บเซอร์วิส ประกอบไปด้วย 4 ส่วนหลัก

จากแนวความคิดข้างต้นปัจจัยพื้นฐานที่นำมาใช้ในการสร้างและพัฒนาเว็บเซอร์วิส ประกอบไปด้วยเทคโนโลยีเหล่านี้

1. โพรโตคอล SOAP

เป็นการยากที่เทคโนโลยีที่ใช้กันในปัจจุบันนี้จะทำให้แอปพลิเคชันกับแอปพลิเคชันสามารถติดต่อสื่อสารกันได้เพราะแต่ละเทคโนโลยีมีการพัฒนาที่แตกต่างกัน สถาปัตยกรรมของแต่ละเทคโนโลยีก็ถูกออกแบบมาให้แตกต่างกัน จึงทำให้มีความยืดหยุ่นในการทำงานน้อย ดังนั้นบริษัทจำนวนมากรวมทั้งไมโครซอฟท์และยูลเซอร์แลนด์ซอฟต์แวร์ได้ร่วมกันสร้างโพรโตคอล SOAP ขึ้นมาเพื่อกระตุ้นให้เทคโนโลยีทั้งหลายสามารถติดต่อสื่อสารกันได้ซึ่ง SOAP เป็นโพรโตคอลที่สร้างขึ้นมานับโพรโตคอลมาตรฐานที่ได้รับการยอมรับ SOAP จะใช้ฟอร์มเมตของ XML ในการอธิบายเมธอดและพารามิเตอร์ทั้งหลายเพื่อสร้างการเรียกใช้งานระยะไกลข้ามเครือข่ายได้ (สุวัฒน์ สุขสมจินตร์, 2546)

โครงสร้างของ SOAP มีโครงสร้างที่อยู่ในรูปภาษา XML ซึ่งสามารถแบ่งส่วนต่างๆ ของเอกสารได้เป็น 3 ส่วน ดังภาพ 8



ภาพ 8 แสดงโครงสร้างภายในเอกสาร SOAP (นิตยา เอี่ยมคง, 2549)

ตัวอย่างสคริปต์ SOAP โดยภายในจะมีสคริปต์ SOAP เพื่ออธิบายบริการดังภาพ 9 เป็น การสอบถามราคาดอกกุหลาบจำนวน 100 ดอก

```
<soap:Envelop>
<soap:Body>
<GetPrice>
<Item>Rose</Item>
<Quantity>100</Quantity>
</GetPrice>
</soap:Body>
<soap:Envelop>
```

ภาพ 9 แสดงตัวอย่างสคริปต์ SOAP

2. ภาษา WSDL

WSDL เป็นภาษาที่ใช้อธิบายคุณลักษณะบริการของเว็บเซอร์วิส วิธีการติดต่อกับเว็บเซอร์วิส (Method Call) ซึ่งจะอธิบายลักษณะของบริการไว้อย่างชัดเจน ชื่อเว็บเซอร์วิส ข้อมูลพารามิเตอร์ของเว็บเซอร์วิส ชนิดของพารามิเตอร์ที่ส่งค่าผ่านให้เว็บเซอร์วิส บอกรหัสของพารามิเตอร์ที่ส่งค่าคืนกลับให้เว็บเซอร์วิส และบอกตำแหน่งของไฟล์ที่ทำหน้าที่คอยรับการเรียกใช้เว็บเซอร์วิส (SOAP Listener) (สราวุธ อ้อยศรีสกุล, 2544; มหาวิทยาลัยเกษตรศาสตร์, 2545; W3C Working Group, 2003; Kopack and Potts, 2003)

2.1 โครงสร้างภาษา WSDL

เนื่องจาก WSDL เป็นภาษาที่อยู่ในความดูแลขององค์กร W3C (World Wide Web Consortium) เวอร์ชันที่ใช้อยู่ในปัจจุบัน คือ WSDL 1.1 บริการของเว็บเซอร์วิสจะมีเอกสาร WSDL เป็นเครื่องมือช่วยสร้างบริการอย่างอัตโนมัติสิ่งที่สำคัญภายในเอกสาร WSDL นั้นจะอธิบายเกี่ยวกับการติดต่อและเรียกใช้บริการของเว็บเซอร์วิส โดย WSDL จะมีส่วนประกอบหรืออิลิเมนต์หลักดังตาราง 1 และตัวอย่างเอกสารพร้อมคำอธิบายในภาพ 10

ตาราง 1 แสดงอิลิเมนต์ของเอกสาร WSDL

อิลิเมนต์	คำจำกัดความ
<portType>	เป็นส่วนที่สำคัญที่สุดของ WSDL Element เป็นการอธิบายโอบเปอร์เรชันของเว็บเซอร์วิสที่มีการให้บริการ และเมสเสจที่เกี่ยวข้องเทียบได้กับ Function Library หรือ โมดูล หรือ Class ในการเขียนโปรแกรม
<operation>	อธิบายเมธอดที่บริการเว็บเซอร์วิสหนึ่งๆ จะมีเมธอดจำนวนกี่เมธอดก็ได้
<message>	อธิบาย Data Element ของโอบเปอร์เรชัน แต่ละเมสเสจอาจมีมากกว่าหนึ่งส่วนเทียบได้กับพารามิเตอร์ของฟังก์ชันในการเขียนโปรแกรม
<type>	อธิบายชนิดข้อมูลที่เว็บเซอร์วิส ใช้เพื่อความเป็นกลาง WSDL ใช้ XML Schema Syntax ในการระบุชนิดข้อมูล
<binding>	อธิบายฟอร์แมตของเมสเสจแต่ละ protocol Details ในแต่ละพอร์ท
<service>	สำหรับเว็บเซิร์ฟเวอร์จะมีเว็บเซอร์วิสให้บริการเท่าไรก็ได้และชื่อเว็บเซอร์วิสก็เป็นตัวจำแนกและบ่งบอกแต่ละบริการซึ่งห้ามมีชื่อซ้ำกัน

<pre><?xml version="1.0" encoding="UTF-8" ?> <wsdl:definitions targetNamespace="http://localhost:8080/axis/services/Version" xmlns:apacheSOAP="http://xml.apache.org/xml-soap" xmlns:impl="http://localhost:8080/axis/services/Version" xmlns:intf="http://localhost:8080/axis/services/Version" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/" xmlns:wSDLsoap="http://schemas.xmlsoap.org/wSDL/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"></pre>	<<definition>
<pre><wsdl:message name="getVersionRequest" /> <wsdl:message name="getVersionResponse"> <wsdl:part name="getVersionReturn" type="xsd:string" /> </wsdl:message></pre>	<<message>
<pre><wsdl:portType name="Version"> <wsdl:operation name="getVersion"> <wsdl:input message="impl:getVersionRequest" name="getVersionRequest" /> <wsdl:output message="impl:getVersionResponse" name="getVersionResponse" /> </wsdl:operation> </wsdl:portType></pre>	<<portType>
<pre><wsdl:binding name="VersionSoapBinding" type="impl:Version"> <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" /> <wsdl:operation name="getVersion"> <wsdlsoap:operation soapAction="" /> <wsdl:input name="getVersionRequest"> <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://axis.apache.org" use="encoded" /> </wsdl:input> <wsdl:output name="getVersionResponse"> <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://localhost:8080/axis/services/Version" use="encoded" /> </wsdl:output> </wsdl:operation> </wsdl:binding></pre>	<<binding>
<pre><wsdl:service name="VersionService"> <wsdl:port binding="impl:VersionSoapBinding" name="Version"> <wsdlsoap:address location="http://localhost:8080/axis/services/Version" /> </wsdl:port> </wsdl:service> </wsdl:definitions></pre>	<<service>

ภาพ 10 อธิบายเอกสาร WSDL

3. มาตรฐาน UDDI

UDDI เป็นมาตรฐานชุดพื้นฐาน APIs (Application Programming Interface) ของ SOAP สามารถนำไปใช้ในการพัฒนาตัวแทนของผู้ให้บริการได้ UDDI เป็นการระบุนายละเอียดภายในของการให้บริการลูกค้า เกี่ยวข้องกับด้านศักยภาพและข้อมูลที่จำเป็นในการติดต่อกับเว็บไซต์ด้วยการดาวน์โหลด การเข้าใช้งานสามารถทำได้ทั้งแบบทั่วไป (Public Directory) คือ อนุญาตให้ทุกคนเข้าถึงข้อมูลได้โดยการลงทะเบียน, แบบส่วนตัว (Private Registry) คือ มีการป้องกันด้วยไฟร์วอลล์ซึ่งในการเข้าถึงข้อมูลนั้นจะกระทำเฉพาะภายในองค์กรเท่านั้นหรือจะมีเพียงสมาชิกเท่านั้นที่สามารถเข้าใช้งานได้ และแบบที่ต้องการความเป็นส่วนตัวในรูปแบบหุ้นส่วน

(Semiprivate Registry) คือเข้าถึงได้ในจำนวนที่จำกัดจากภายนอก เช่น สามารถเข้าถึงได้เฉพาะหุ้นส่วนทางการค้าที่เป็นบุคคลสำคัญเท่านั้น เป็นต้น

ดังนั้น UDDI จึงเปรียบเสมือนเป็นสมุดหน้าเหลืองที่มีมาตรฐานที่ตั้งขึ้นโดยบริษัทไอบีเอ็ม บริษัทไมโครซอฟท์ และบริษัทอริบา (Ariba) ปัจจุบันมีบริษัทร่วมกันกำหนดมาตรฐานให้กับ UDDI มากกว่า 70 บริษัททั่วโลก (ฉัตรชัย สุขสอาด, ม.ป.ป., ไม่มีเลขหน้า)

4. ภาษา XML

ภาษา XML เป็นเครื่องมือที่ถูกนำไปใช้ในการสร้างรายละเอียดในเอกสารซึ่งเป็นมาตรฐานในการแลกเปลี่ยนข้อมูลบนอินเทอร์เน็ตโดยผู้ที่ทำหน้าที่รับผิดชอบและกำหนดมาตรฐานคือ W3C รายละเอียดของไวยากรณ์จะอยู่ในรูปของ XML Schemas ภายใน XML Schemas จะประกอบด้วยแท็ก (Tags) ต่างๆ และความสัมพันธ์ภายในแต่ละส่วนของเอกสารเหล่านี้ SOAP, WSDL และ UDDI จะมีพื้นฐานมาจากภาษา XML ทั้งสิ้น ตัวอย่างสคริปต์เอกสาร XML มีดังนี้ (มหาวิทยาลัยเกษตรศาสตร์, 2545; Kopack and Potts, 2003)

```
<computerBook>
  <book>
    <name>Web Service</name>
    <price>10.00$</price>
  </book>
  <book>
    <name>XML</name>
    <price>20.00$</price>
  </book>
</computerBook>
```

ภาพ 11 แสดงตัวอย่างสคริปต์ XML

จากภาพ 11 แสดงตัวอย่างสคริปต์ XML สามารถอธิบายได้ว่า ภายในแท็กของหนังสือคอมพิวเตอร์นั้นประกอบด้วยหนังสือ 2 เล่ม คือ หนังสือชื่อเว็บเซอร์วิส ราคา 10 เหรียญ และหนังสือชื่อ XML ราคา 20 เหรียญ ภายในเอกสาร XML ไม่ได้บอกถึงวิธีการแสดงผล หากจะกำหนดการแสดงผลจะต้องใช้ควบคู่กับภาษา HTML และเอกสารสามารถสื่อความหมายให้คอมพิวเตอร์เข้าใจได้รวมถึงนำค่าที่ได้ไปประมวลผลต่อได้ (ฉัตรชัย สุขสอาด, ม.ป.ป.)

ตัวอย่างการใช้งานเว็บเซอร์วิส

องค์กรต่างๆ สามารถนำเว็บเซอร์วิสมาใช้ได้หลายลักษณะ เช่น ใช้เชื่อมต่อระหว่างบริษัทสองแห่งที่เป็นหุ้นส่วนกัน หรือใช้ในสถานการณ์ของแอปพลิเคชันตัวแทนจำหน่ายสินค้าโอท็อป สมมติว่าตัวแทนจำหน่ายสินค้าโอท็อปต้องการดึงข้อมูลเกี่ยวกับสินค้าโอท็อปของแต่ละตำบลมาไว้ในร้านจำหน่ายออนไลน์ ก็สามารถสร้างเว็บเซอร์วิส (Implement Web Service) ของแต่ละตำบล และสร้างแอปพลิเคชันตัวแทนจำหน่ายให้เรียกใช้บริการของเว็บเซอร์วิสของแต่ละตำบล ตัวอย่างเว็บไซต์ที่ให้บริการเว็บเซอร์วิส คือ เว็บไซต์ของกรมสรรพากร ที่ให้บริการข้อมูลเกี่ยวกับทางด้านการเงิน การลงทุน อัตราการแลกเปลี่ยน และภาษี และ กูเกิล เว็บไซต์ค้นหายอดนิยม ให้บริการเกี่ยวกับการค้นหาต่างๆ (Search Engine) และการตรวจสอบคำที่สะกดผิด (Spell Checked) ลักษณะรูปแบบขององค์กรที่เหมาะสมกับการนำเว็บเซอร์วิสมาใช้ คือ องค์กรที่มีข้อมูลขององค์กรอื่นต้องการใช้ โดยที่องค์กรอื่นไม่สามารถเข้าถึงข้อมูลนั้นโดยตรง ส่วนใหญ่องค์กรของภาครัฐบาล เหมาะกับการสร้างเว็บเซอร์วิสแบบให้บริการ (Service Provider) และองค์กรของภาคเอกชน เหมาะกับการนำบริการของเว็บเซอร์วิสไปใช้ (Service Requester)

ข้อดีของเว็บเซอร์วิส สามารถสรุปเป็นข้อๆ ได้ดังนี้ (Kopack and Potts, 2003)

1. เว็บเซอร์วิสสามารถทำงานร่วมกับระบบเก่าได้ (Legacy System)
2. ต้นทุนในการปรับเปลี่ยนจากระบบงานเก่ามาเป็นระบบงานที่ใช้เว็บเซอร์วิสนั้นมีมูลค่าต่ำ (Lower Operational Costs)
3. ต้นทุนในการพัฒนาซอฟต์แวร์ต่ำ (Lower Software Development Cost)
4. เป็นการพัฒนาระบบที่รวดเร็วกว่า (Faster System Development)
5. โครงสร้างการเชื่อมโยงกับลูกค้าดีกว่า (Better Interfaces with Customers)
6. มีการติดต่อกับหุ้นส่วนทางธุรกิจที่ดีกว่า (Better Integration With External Business Partners)
7. ลดปัญหาความซ้ำซ้อนของรายจ่ายที่ไม่เหมาะสมจึงทำให้เกิดรายได้ใหม่ที่เหมาะสม (New Revenue Opportunities)

8. เพิ่มความทันสมัยให้กับองค์กรทำให้เกิดความสำเร็จในธุรกิจรูปแบบใหม่
(Completely New Business Models)

ข้อเสียของเว็บเซอร์วิส สามารถสรุปเป็นข้อๆ ได้ดังนี้

1. ทำงานช้าเพราะในขั้นตอนการพัฒนานั้นได้แยกโค้ดในส่วนเชื่อมต่อออกจากฐานข้อมูล ทำให้มีความล่าช้าในการค้นหาฐานข้อมูลที่ต้องการในแต่ละครั้งของการร้องขอบริการ
2. ในการใช้งานเว็บเซอร์วิสนั้นผู้เรียกใช้บริการจะต้องกำหนดโค้ดในส่วนที่ใช้เชื่อมต่อกับเว็บเซอร์วิสนั้นให้ตรงกับเอกสาร WSDL

การวิเคราะห์และออกแบบเชิงวัตถุ

แนวความคิดเชิงวัตถุได้กลายเป็นที่นิยมในปัจจุบัน ด้วยเหตุที่มีการมองปัญหาและสิ่งที่เกิดขึ้นภายในระบบเป็นออบเจกต์หรือวัตถุ ซึ่งตรงตามสภาพความเป็นจริง โดยที่ในแต่ละออบเจกต์จะประกอบไปด้วยคุณสมบัติและพฤติกรรม (Property and Behavior) ของการทำงาน การพัฒนาเชิงวัตถุช่วยให้ง่ายที่จะทำความเข้าใจ ประหยัดเวลาในการทำงานเนื่องจากสามารถนำกลับมาใช้ใหม่ได้และง่ายต่อการสื่อสาร ด้วยเหตุผลนี้จึงทำให้ประหยัดค่าใช้จ่ายในการพัฒนาระบบ รวมถึงเป็นวิธีที่มีประสิทธิภาพ และภาษาที่นำแนวคิดเชิงวัตถุมาใช้ในปัจจุบัน ได้แก่ ภาษา JAVA, C++, Visual Studio.NET และภาษา Ruby เป็นต้น (สุนทริน วงศ์ศิริกุล, ม.ป.ป.; โสภาส เขียมสิริวงศ์, 2548; ศิริลักษณ์ โรจนกิจอำนวย, 2545)

แนวความคิดแบบออบเจกต์ประกอบด้วย ออบเจกต์ (Object), คลาส (Class), วิธีการ (Method), เอนแคปซูเลชัน (Encapsulation) และ การสืบทอดคุณสมบัติ (Inheritance) สามารถอธิบายได้ดังนี้ (สุนทริน วงศ์ศิริกุล, ม.ป.ป.; โสภาส เขียมสิริวงศ์, 2548; ศิริลักษณ์ โรจนกิจอำนวย, 2545)

ออบเจกต์ คือ สิ่งที่เราให้ความสนใจเป็นเหตุการณ์ที่เกิดขึ้นภายในระบบ แต่ละออบเจกต์จะประกอบไปด้วย 3 ส่วนหลัก คือ สถานะ (State), พฤติกรรม (Behavior) และ คุณลักษณะ (Identity)

คลาส เป็นการระบุว่าในแต่ละออบเจกต์ประกอบไปด้วย แอททริบิวต์ (Attribute) และ โอเปอเรชัน (Operation) คลาสเปรียบเสมือนแม่แบบซึ่งไม่สามารถนำไปใช้งานได้ แต่สามารถนำไปเป็นแม่แบบเพื่อสร้างออบเจกต์เพื่อนำไปใช้งานต่อไป

วิธีการ เป็นตัวประสาน (Interface) ในคลาสและเป็นการนำโอเปอเรชันไปใช้ โดยกำหนดวิธีการขึ้นมาเพื่อดำเนินการกับออบเจกต์

เอนแคปซูเลชัน เป็นการนำเอาคุณสมบัติและพฤติกรรมรวมไว้ด้วยกัน เพื่อป้องกันไม่ให้คลาสอื่น เรียกใช้คุณสมบัติ (Attribute) ได้โดยตรง แต่ต้องเรียกใช้ผ่านพฤติกรรม (method) ของคลาสนั้น

การสืบทอดคุณสมบัติ เป็นคุณสมบัติที่คลาสๆ หนึ่ง สามารถสืบลักษณะของแอททริบิวท์และเมธอดของ อีกคลาสหนึ่งได้ การทำเช่นนี้ทำให้คุณสามารถสร้างคลาสใหม่ขึ้นโดยนำสาระสำคัญ ที่เหมือนกันทั้งคุณสมบัติและพฤติกรรมจากคลาสนั้นมาใช้ได้

ในการพัฒนาโปรแกรมเชิงวัตถุ นั้น ส่วนมากจะใช้ภาษาที่ช่วยในการพัฒนาและสื่อสารกับบุคคลอื่นได้ง่ายยิ่งขึ้น ซึ่งภาษาที่เป็นที่นิยมในปัจจุบัน คือ ภาษา UML

ภาษา UML (Unified Modeling Languages)

ในการพัฒนาและออกแบบเชิงวัตถุ นั้น ภาษา UML ถือเป็นเครื่องมือหนึ่งที่จะช่วยในการวิเคราะห์และออกแบบตามแนวคิดเชิงวัตถุ (บรรจง หะรังษี และญาณวรรณ สิ้นสุภิญญา, ม.ป.ป.) ได้ให้ความหมายว่า

"โมเดล (Modeling Language) คือ ความพยายามในการที่จะอธิบายปัญหาของซอฟต์แวร์ที่จะดำเนินการพัฒนาขึ้นมา ตัวโมเดลจะแสดงให้เห็นถึงออบเจกต์ต่างๆที่เกี่ยวข้องและความสัมพันธ์ระหว่างออบเจกต์เหล่านั้น นอกจากนั้นโมเดลยังแสดงให้เห็นถึงวิธีการที่จะแก้ไขปัญหา โดยใช้แผนภาพ เนื้อความ (Text) หรือรูปแบบอื่นๆ ซึ่งเป็นที่ยอมรับกันระหว่างผู้พัฒนาและผู้ใช้ระบบ ในการนำเสนอโมเดลหนึ่ง UML จะใช้ทั้งไดอะแกรมและเนื้อความเพื่อทำการนำเสนอโมเดล และเมื่อกล่าวถึง Modeling Language ความหมายคือ ภาษาที่ใช้ในการอธิบายโมเดลนั่นเอง"

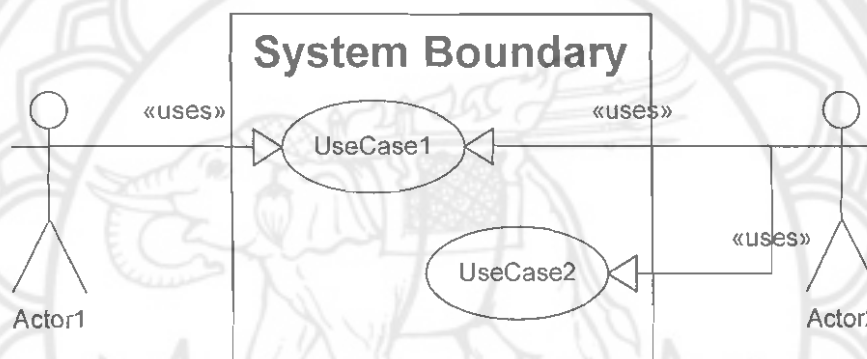
การวิเคราะห์และออกแบบเชิงวัตถุด้วย UML มีสัญลักษณ์ด้วยกัน 9 ไดอะแกรม ดังนี้

1. Activity Diagram คือ ไดอะแกรมที่แสดงกิจกรรมทั้งหมดที่เกิดขึ้นในระบบ
2. Class Diagram คือ ไดอะแกรมที่แสดงคลาสและความสัมพันธ์ระหว่างคลาส
3. Collaboration Diagram คือ ไดอะแกรมที่แสดงลำดับการทำงานของ วัตถุ ผู้เกี่ยวข้อง และกิจกรรม โดยลำดับการทำงานไม่ขึ้นกับเวลา
4. Component Diagram คือ ไดอะแกรมที่แสดงถึงโครงสร้างทางกายภาพของระบบ
5. Deployment Diagram คือ แสดงระบบสถาปัตยกรรมของ Hardware/Software ตลอดจนความสัมพันธ์ระหว่าง hardware/software
6. Object Diagram คือ ไดอะแกรมที่แสดงออบเจกต์และความสัมพันธ์ระหว่างออบเจกต์
7. Sequence Diagram คือ ไดอะแกรมที่แสดงลำดับการทำงานของระบบ
8. State Diagram คือ ไดอะแกรมที่แสดงถึงสถานะของออบเจกต์

9. Use-Case Diagram คือ ไดอะแกรมที่บอกถึงความต้องการของระบบ

จากทั้ง 9 ไดอะแกรมข้างต้นผู้วิจัยได้นำ UML มาช่วยในการวิเคราะห์และออกแบบระบบ โดยเลือกใช้โมเดลที่เป็นที่นิยมและถูกใช้บ่อย ซึ่งมีด้วยกัน 3 ไดอะแกรม คือ User-Case Diagram, Class Diagram และ Sequence Diagram ดังนี้

Use Case Diagram เป็นไดอะแกรมที่แสดงภาพรวมการทำงานของระบบทั้งหมดของระบบ แสดงให้เห็นถึงการกระทำ การโต้ตอบกันระหว่างบุคคล หน่วยงาน หรือซอฟต์แวร์ที่ปฏิสัมพันธ์กันในระบบ โดยทั่วไปประกอบไปด้วย 2 ส่วนหลัก คือ actor และ Use Case ดังภาพ 12 (สุนทริน วงศ์ศิริกุล, ม.ป.ป.; Triyawatt, 2001)



ภาพ 12 แสดงสัญลักษณ์ที่เป็นส่วนประกอบ Use Case Diagram

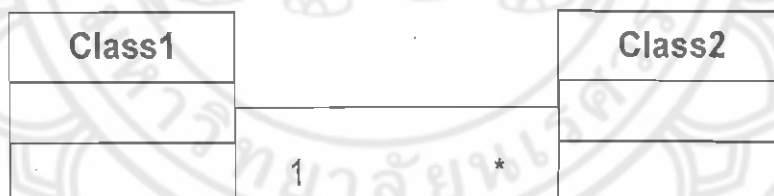
Class Diagram เป็นไดอะแกรมที่แสดงถึงส่วนประกอบของคลาส ซึ่งคลาสประกอบด้วย 3 ส่วน คือ ชื่อคลาส แอททริบิวต์ และโอเปอเรชัน สัญลักษณ์ที่ใช้เป็นรูปสี่เหลี่ยมผืนผ้า ดังภาพ 13 (สุนทริน วงศ์ศิริกุล, ม.ป.ป.; Triyawatt, 2001)

ClassName
-Attribute1
-Attribute2
-Attribute3
+Operation1()
+Operation2()
+Operation3()

ภาพ 13 แสดงส่วนประกอบภายในคลาส

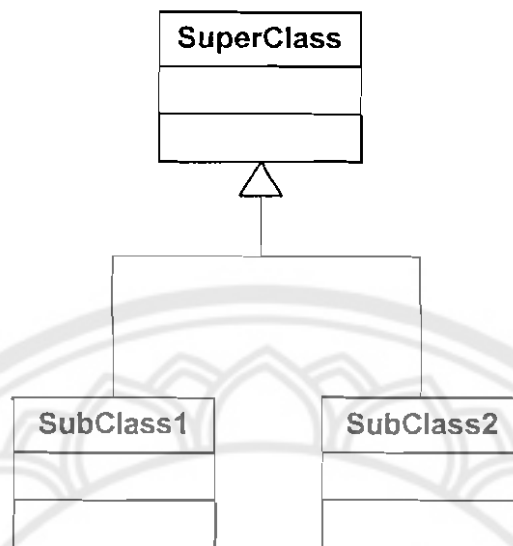
ความสัมพันธ์ระหว่างคลาส (Class Relationship) แบ่งออกเป็น 3 ประเภท

1. Association เป็นการเชื่อมโยงระหว่างคลาสที่มีความสัมพันธ์ในลักษณะที่เป็นเอ็นทิตีอ่อนแอ (Weak Entity) หรือที่เรียกว่า Qualified Association ดังภาพ 14 (สุนทริน วงศ์ศิริกุล, ม.ป.ป.)



ภาพ 14 แสดงความสัมพันธ์ระหว่างคลาสแบบ Association

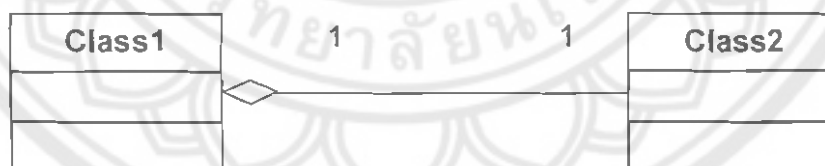
2. Generalization (หรือ Inheritance) เป็นการเชื่อมต่อระหว่างคลาสที่มีความสัมพันธ์แบบสืบทอดดังเช่น การถ่ายทอดคุณสมบัติจากคลาสแม่ (Superclass) ไปยังคลาสลูก (Subclass) ดังภาพ 15



ภาพ 15 แสดงความสัมพันธ์ระหว่างคลาสแบบ Generalization

3. Aggregation เป็นการเชื่อมโยงระหว่างคลาสที่เป็นส่วนหนึ่งของอีกคลาสหนึ่งแบ่งเป็น 2 ประเภท

3.1 Shared Aggregation เป็นการแสดงความสัมพันธ์แบบไม่เหนียวแน่นสามารถแยกออกจากกันได้ ไม่จำเป็นต้องอยู่ด้วยกัน ความสัมพันธ์แบบนี้สามารถอธิบายด้วยสัญลักษณ์สี่เหลี่ยมข้าวหลามตัดโปร่ง ดังภาพ 16 (ศิริลักษณ์ โรจนกิจอำนวย, 2545; Triyawatt, 2001)



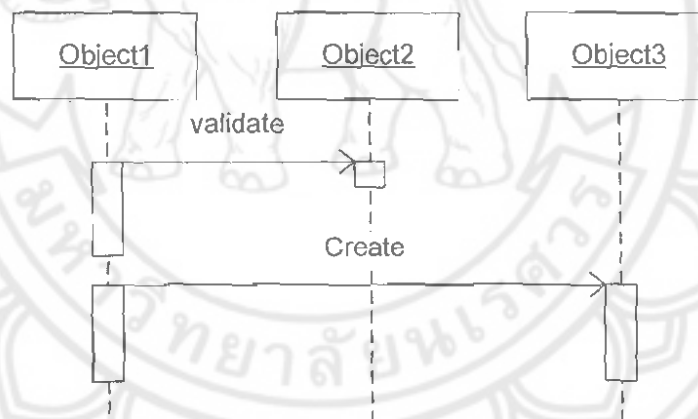
ภาพ 16 แสดงสัญลักษณ์ Shared Aggregation ที่มีความสัมพันธ์ระหว่างคลาส

3.1.1 Composite Aggregation เป็นสัญลักษณ์ของ Aggregation ที่แสดงความสัมพันธ์แบบเหนียวแน่นไม่สามารถแยกออกจากกันได้ เมื่อคลาสแม่ตายคลาสลูกก็จะตายตามไปด้วย จะประกอบเป็นเนื้อเดียวกัน เช่น โรงเรียนจะต้องมีห้องเรียนอยู่ภายในโรงเรียน เป็นต้น สัญลักษณ์ที่ใช้เป็นรูปสี่เหลี่ยมข้าวหลามตัดทึบ ดังภาพ 17 (ศิริลักษณ์ โรจนกิจอำนวย, 2545; Triyawatt, 2001)



ภาพ 17 แสดงสัญลักษณ์ Composite Aggregationที่มีความสัมพันธ์ระหว่างคลาส

3.1.2 Sequence Diagram เป็นไดอะแกรมที่แสดงการเชื่อมโยง แสดงขั้นตอนต่างๆ ในการทำงาน แสดงขั้นตอนการติดต่อสื่อสารกันของออบเจกต์ด้วยการส่งข้อความ (Message) ไดอะแกรมนี้จะช่วยให้ผู้พัฒนาระบบเข้าใจถึงการทำงานได้รวดเร็ว และสามารถควบคุมโปรแกรมตามที่ได้ออกแบบไว้ แต่ไดอะแกรมนี้ไม่สามารถแสดงความสัมพันธ์ระหว่างออบเจกต์ได้ ส่วนประกอบภายใน Sequence Diagram แสดงดังภาพ 18



ภาพ 18 แสดงองค์ประกอบของ Sequence Diagram

ข้อดีของการออกแบบและพัฒนาเชิงวัตถุ สามารถสรุปได้ ดังนี้ (สุนทริน วงศ์ศิริกุล, ม.ป.ป.; Ambler, 2002)

1. มีความเป็นอิสระต่อการทำงาน
2. จำแนกการทำงานออกเป็นกลุ่มตามคุณลักษณะของแต่ละออบเจกต์
3. ในการทำงานแต่ละออบเจกต์ สามารถทำงานร่วมกันได้ที่มีการถ่ายทอดคุณสมบัติและพฤติกรรมให้กับออบเจกต์ สร้างความสัมพันธ์ระหว่างออบเจกต์



4. เป็นเครื่องมือสื่อสารที่ง่ายต่อการติดต่อสื่อสาร สามารถช่วยในการตัดสินใจได้รวดเร็วขึ้น

5. สามารถนำกลับมาพัฒนาใหม่ได้ง่ายและเอื้อต่อการปรับเปลี่ยน

6. สามารถรองรับการใช้งานได้ในปริมาณที่มาก

7. ลดความซับซ้อนในการทำงานด้วยวิธีสืบทอดคุณสมบัติและแอนิเมชัน

ข้อเสียของการพัฒนาและออกแบบเชิงวัตถุ สามารถสรุปได้ ดังนี้

1. การเรียกใช้งานผ่านการส่งข้อมูล ทำให้ทำงานล่าช้า เพราะแต่ละออบเจกต์ต้องทำงานร่วมกัน

2. ใช้เวลาในการประมวลผลนาน หากมีความสัมพันธ์กับหลายๆ ออบเจกต์ที่ถูกใช้งานในโพรเซสที่แตกต่างกันหลายโพรเซส

3. มีความซับซ้อนในการพัฒนาและออกแบบสูง

คุณภาพการบริการเว็บเซอร์วิส (Quality of Service)

ในการให้บริการทางธุรกิจที่มีหลากหลาย เว็บเซอร์วิสเป็นอีกหนึ่งทางเลือกที่ถูกนำมาใช้ในการให้บริการข้อมูลต่างๆ และในการสร้างเว็บเซอร์วิสของแต่ละธุรกิจแตกต่างกัน เช่น ความต้องการช่องทางการส่งข้อมูล (Bandwidth), ความต้องการเวลาในการประมวลผล (Processing time) ฯลฯ ด้วยเหตุนี้เองจึงทำให้มีการจัดสร้างมาตรฐานในการบริการให้กับเว็บเซอร์วิสเพื่อให้เป็นมาตรฐานเดียวกัน เรียกว่า คุณภาพการบริการของเว็บเซอร์วิส มีดังนี้ (Lee, et al., 2003)

สมรรถภาพ (Performance)

สมรรถภาพของเว็บเซอร์วิส คือ ความรวดเร็วในการร้องขอข้อมูลจนได้ข้อมูลนั้นมาตัวแปรที่ถูกนำมาใช้ในการทดสอบสมรรถภาพมีดังนี้ (Lee, et al., 2003; Meier, et al., 2004; Sumra and Arulazi, 2004.)

1. Throughput คือ จำนวนครั้งของเว็บเซอร์วิสที่มีการร้องขอไปยังเครื่องที่ให้บริการในระยะเวลาหนึ่งหน่วยที่กำหนดไว้ (Timerperiod)

2. Response Time คือ ระยะเวลาในการตอบรับข้อมูลที่ได้ถูกร้องขอไป

3. Latency คือ จำนวนเวลาที่ใช้ในแต่ละรอบ (Round-Trip) ระหว่างการส่งการร้องขอ (Request) จนถึงการตอบรับ (Response)

4. Execution Time คือ เวลาที่หยิบขึ้นมาในการประมวลผลของเว็บเซอร์วิสตามความถี่ของการกระทำ (Sequence of Activities)

5. Transaction Time คือ ช่วงเวลาขณะที่เว็บเซอวิสกำลังทำงานในแต่ละกระบวนการทำงาน

6. Processor Time คือ เวลาที่ใช้ในการประมวลผลเป็นเปอร์เซ็นต์

7. Memory คือ หน่วยความจำที่ใช้ไปในการทำงานโดยเฉลี่ย

โดยทั่วไปแล้วเว็บเซอวิสที่ดีควรมี จำนวนครั้งที่ร้องขอ (Throughput) สูง, เวลาในการตอบรับ (Response Time) ที่รวดเร็ว, ศักยภาพ (Latency) ที่ถูกใช้ต่ำ, เวลาที่ใช้ในการประมวลผล (Execution Time) ต่ำ และเวลาในการทำงาน (Transaction Time) แต่ละขั้นตอนที่รวดเร็ว

ความน่าเชื่อถือ (Reliability)

เป็นคุณลักษณะของเว็บเซอวิสที่แสดงให้เห็นถึงระดับเริ่มต้น และการให้การดูแลในบริการ จำนวนข้อผิดพลาดและความล้มเหลวต่อเดือนหรือต่อปี ที่จะวัดได้ของเว็บเซอวิส นอกจากนี้ ยังกล่าวถึงอีกเหตุผลหนึ่งคือ ความปลอดภัยและลำดับการบริการของข้อมูลที่เริ่มส่งและรับจากผู้ร้องขอบริการและผู้ให้บริการ (Meier, et al., 2004)

การรองรับการเข้าถึง (Accessibility)

การรองรับการเข้าถึงข้อมูลของเว็บเซอวิสแสดงให้เห็นว่า เว็บเซอวิสสามารถรองรับการร้องขอของไคลเอนต์ปริมาณการเข้าถึงที่เว็บเซอวิสสามารถรองรับได้ (Meier, et al., 2004)

ความพร้อมในการบริการ (Availability)

ความพร้อมของเว็บเซอวิสที่จะเป็นตัวกลางของระบบ โดยมีความสัมพันธ์กับความน่าเชื่อถือ เวลาที่ใช้ในการแก้ไขข้อผิดพลาดที่เกิดขึ้น (Time-To-Repair) ความเหมาะสมของเวลาต่อการร้องขอในแต่ละครั้ง (Meier, et al., 2004)

ความสามารถในการควบคุม (Regulatory)

การควบคุมคุณภาพของเว็บเซอวิสในส่วนของโครงสร้างกับกฎเกณฑ์ ข้อบังคับ กับมาตรฐานที่มีร่วมกัน และระดับวัตถุประสงค์การให้บริการ เว็บเซอวิสใช้มาตรฐาน SOAP, UDDI และ WSDL ดังนั้น บริการที่จัดให้จะขึ้นอยู่กับความจำเป็นความเหมาะสมของเว็บเซอวิสจากผู้ให้บริการ (Mani and Nagarajan, 2002)

คุณภาพของเว็บเซอวิสมีองค์ประกอบหลายประการ ซึ่งบางประการมีความยุ่งยากและใช้เวลานานในการทดสอบและวัดผล โดยในการทำวิจัยครั้งนี้จะทำการทดสอบและประสิทธิภาพของเว็บเซอวิส และความสามารถในการรองรับการเข้าถึงเท่านั้น ส่วนเรื่องของความสามารถในการควบคุมจะสุ่มตรวจสอบคำตอบที่ได้ เพื่อให้แน่ใจว่าปฏิบัติตามกฎในระดับหนึ่ง ความเชื่อถือและความพร้อมในการให้บริการนี้จำเป็นต้องใช้เวลานานในการตรวจสอบ งานวิจัยนี้จึงไม่ขอตรวจสอบคุณภาพดังกล่าว

หลักสถิติที่นำมาใช้ในการวิจัย

หลักสถิติเบื้องต้นที่ถูกนำมาใช้ในการศึกษาประสิทธิภาพการให้บริการเว็บเซอร์วิสได้แก่ การหาค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน และการอธิบายลักษณะของข้อมูลโดยใช้กราฟเส้น เป็นต้น และเพื่อศึกษาเปรียบเทียบประสิทธิภาพของเว็บเซอร์วิส 2 รูปแบบว่ามีความแตกต่างกันอย่างมีนัยสำคัญทางสถิติหรือไม่ ผู้วิจัยจึงได้นำเทคนิคการทดสอบความแตกต่างระหว่างค่าเฉลี่ยของประชากรสองกลุ่มที่เป็นอิสระต่อกันที่มีชื่อเรียกว่าการทดสอบแบบที (Independent Samples T-Test) มาใช้ นอกจากนี้ เพื่อทำการศึกษาเปรียบเทียบประสิทธิภาพของเว็บเซอร์วิสทั้ง 4 รูปแบบพร้อมๆ กัน จึงได้นำเทคนิคการวิเคราะห์ความแปรปรวนแบบจำแนกทางเดียว (One Way Analysis of Variance: One Way ANOVA) มาใช้

การทดสอบความแตกต่างของค่าเฉลี่ยประชากร 2 กลุ่ม ในกรณีกลุ่มตัวอย่างทั้งสองกลุ่มเป็นอิสระต่อกัน (Independent Samples T-Test) เป็นการทดสอบความแตกต่างระหว่างค่าเฉลี่ยของตัวอย่างทั้งสองกลุ่ม เมื่อกลุ่มตัวอย่างที่สุ่มมานั้นไม่มีความสัมพันธ์กันหรือเป็นอิสระต่อกัน โดยในการศึกษานี้ผู้วิจัยได้ตั้งสมมติฐานในการทดสอบ ดังนี้

H_0 : ประสิทธิภาพของเว็บเซอร์วิสทั้ง 2 รูปแบบไม่แตกต่างกัน

H_1 : ประสิทธิภาพของเว็บเซอร์วิสทั้ง 2 รูปแบบแตกต่างกัน

โดยกำหนดระดับนัยสำคัญในการทดสอบเท่ากับ 0.025 และจะตัดสินใจยอมรับหรือปฏิเสธ H_0 โดยการพิจารณาค่า P-value หรือ Sig ใน SPSS จากผลลัพธ์ที่ได้จากโปรแกรม SPSS for Windows เวอร์ชัน 11.5 ถ้าค่า Sig มีค่ามากกว่า 0.025 จะยอมรับ H_0 (ปฏิเสธ H_1) และจะปฏิเสธ H_0 (ยอมรับ H_1) ถ้าค่า Sig มีค่าน้อยกว่าหรือเท่ากับ 0.025

การวิเคราะห์ความแปรปรวนแบบจำแนกทางเดียว (One Way ANOVA) เป็นการทดสอบความแตกต่างระหว่างค่าเฉลี่ยของประชากรตั้งแต่ 3 กลุ่มขึ้นไป โดยถือว่าหน่วยที่ได้รับปัจจัยเดียวกันจะมีลักษณะคล้ายคลึงกันมากที่สุด และหน่วยที่ได้รับปัจจัยต่างกันจะมีลักษณะที่แตกต่างกันมากที่สุด (กัลยา วาณิชย์บัญชา, 2540) โดยในการศึกษาผู้วิจัยได้ตั้งสมมติฐานว่า

H_0 : ประสิทธิภาพของเว็บเซอร์วิสทุกรูปแบบไม่แตกต่างกัน

H_1 : ประสิทธิภาพของเว็บเซอร์วิสอย่างน้อย 1 คู่มีความแตกต่างกัน

โดยกำหนดระดับนัยสำคัญในการทดสอบเท่ากับ 0.05 ในการตัดสินใจผลที่ได้จากการวิเคราะห์ความแปรปรวนนั้น จะพิจารณาค่า Sig ถ้าค่า Sig มีค่ามากกว่า 0.05 จะยอมรับ H_0 (ปฏิเสธ H_1) และจะปฏิเสธ H_0 (ยอมรับ H_1) ถ้าค่า Sig มีค่าน้อยกว่าหรือเท่ากับ 0.05 ถ้าผลสรุปที่ได้พบว่าเกิดการ ปฏิเสธ H_0 แสดงว่ามีความแตกต่างระหว่างประสิทธิภาพของเว็บเซอร์วิสอย่างน้อย 1 คู่ จะต้องมีการเปรียบเทียบต่อ เพื่อหาว่ามีคู่ใดบ้างที่แตกต่างกัน ซึ่งเทคนิคที่ใช้มีชื่อเรียกว่า

การเปรียบเทียบเชิงซ้อน (Multiple comparison) หรือทฤษฎี Post hoc test ในโปรแกรม SPSS for Windows นั้นเอง

งานวิจัยที่เกี่ยวข้อง

Wickramage and Weerawarana (2005, pp. 233–242) ได้กล่าวว่า ในงานของเว็บเซอร์วิส เครื่องลูกข่ายและเครื่องแม่ข่ายจะแลกเปลี่ยนข้อมูลโดยใช้ SOAP Message เครื่องลูกข่ายที่เรียกใช้ method บนเครื่องแม่ข่ายโดยนำพารามิเตอร์ต่างๆ และชื่อของ method ของการร้องขอเข้าไปใน SOAP Message (SOAP Serialize) เครื่องแม่ข่ายนำ SOAP Message มาแยกส่วนออก (Desterilized) และพารามิเตอร์ต่างๆ จะถูกสร้างขึ้นใหม่ ดังนั้น สามารถส่งค่าต่างเข้าไปยัง method ที่รอการร้องขอบนโฮสต์ได้ ข้อมูลถูกส่งกลับมาโดย method ที่ฝังใน SOAP Message และส่งกลับไปยังเครื่องลูกข่าย ซึ่งเครื่องลูกข่ายสามารถแยกส่วน SOAP Message และอ่านค่าออกมาเพื่อนำไปใช้ประมวลผลต่อไป เพราะฉะนั้น Round Trip Time (RTT) สำหรับ เครื่องลูกข่ายที่ร้องขอ method บนเครื่องแม่ข่ายและรับค่ากลับคืนมานั้น ขึ้นอยู่กับปัจจัยดังนี้

1. เวลาที่ใช้ไปโดยเครื่องลูกข่ายเว็บเซอร์วิสอ่านข้อมูลจากโครงสร้างข้อมูล
2. เวลาที่ใช้ไปโดยเครื่องลูกข่ายเว็บเซอร์วิสสร้าง SOAP Message โดยฝังค่าเหล่านั้น
3. เวลาที่ใช้ไปโดยเครื่องลูกข่ายเว็บเซอร์วิสส่ง SOAP Message
4. เวลาที่ใช้ไปโดย SOAP Message มาถึงเครื่องแม่ข่าย
5. เวลาที่ใช้ไปโดยเครื่องแม่ข่ายเว็บเซอร์วิสอ่าน SOAP Message
6. เวลาที่ใช้ไปโดยเครื่องแม่ข่ายเว็บเซอร์วิสวิเคราะห์ SOAP Message และดึงข้อมูลที่

ต้องการออกมา

7. เวลาที่ใช้ไปโดยเครื่องแม่ข่ายเว็บเซอร์วิสสร้างโครงสร้างข้อมูลเพื่อที่จะส่งให้กับ method ที่เหมาะสม

8. เวลาที่ใช้ไปโดย method ประมวลผลเสร็จ (และส่งค่ากลับ)
9. เวลาที่ใช้ไปโดยเครื่องแม่ข่ายเว็บเซอร์วิสดึงค่าออกจากโครงสร้างข้อมูลการส่งกลับ
10. เวลาที่ใช้ไปโดยเครื่องแม่ข่ายเว็บเซอร์วิสสร้าง SOAP Message โดยฝังค่าเหล่านั้นด้วย
11. เวลาที่ใช้ไปโดยเครื่องแม่ข่ายเว็บเซอร์วิสส่ง SOAP Message
12. เวลาที่ใช้ไปโดย SOAP Message ถึงเครื่องลูกข่าย
13. เวลาที่ใช้ไปโดยเครื่องลูกข่ายเว็บเซอร์วิสอ่าน SOAP Message
14. เวลาที่ใช้ไปโดยเครื่องลูกข่ายเว็บเซอร์วิสวิเคราะห์ SOAP Message และดึงข้อมูลที่

ต้องการออกมา

15. เวลาที่ใช้ไปโดยเครื่องลูกข่ายเว็บเซอร์วิสสร้างโครงสร้างข้อมูลเพื่อนำไปประมวลผลต่อไป

จากปัจจัยทั้ง 15 ข้อนั้น ปัจจัยที่ 4 และ 12 ขึ้นอยู่กับประสิทธิภาพของเครื่องข่าย ปัจจัยที่ 2, 6, 10 และ 14 ขึ้นอยู่กับตัววิเคราะห์ XML ใช้ในขอบข่ายงานเว็บเซอร์วิส (Web Service Framework) โดยทั่วไปแล้ว ขอบข่ายงานเว็บเซอร์วิสเป็นขอบข่ายงานที่เป็นอิสระของตัววิเคราะห์ XML เฉพาะและผู้ใช้อาจจะเลือกเฉพาะตัววิเคราะห์ XML ใช้ภายในขอบข่ายงาน ปัจจัยที่ 3, 5, 11 และ 13 มีแนวโน้มที่จะขึ้นอยู่กับ HTTP Client & Server มากกว่าขอบข่ายงานของเว็บเซอร์วิส ปัจจัยที่ 8 เป็นการทำงานโดยอิสระของขอบข่ายงานเว็บเซอร์วิส ดังนั้นเหลือปัจจัยที่ 1, 7, 9 และ 15 เป็นปัจจัยที่ทำให้ RTT สามารถขึ้นอยู่กับประสิทธิภาพของขอบข่ายงานเว็บเซอร์วิสจริงๆ (Wickramage and Weerawarana, 2005)

ในงานวิจัยนี้ได้ออกแบบตัววัดประสิทธิภาพของเว็บเซอร์วิสตามลักษณะที่ Wickramage and Weerawarana ได้นำเสนอเอาไว้ โดยจะแบ่งช่วงเวลากการวัดประสิทธิภาพออกเป็น 4 ช่วง ซึ่งจะอธิบายโดยละเอียดในบทที่ 4 ต่อไป