

ASSIGNMENT 1



“หนึ่งหน่วยกลุ่ม” (Group 1)

Selected Topics in Computer Engineering II



CONTENTS

- 1. web services using SOAP on the Internet/Cloud
- 2. web services using WSDL on the Internet/Cloud

web services using SOAP on the Internet/Cloud

SOAP(Simple Object Access Protocol)

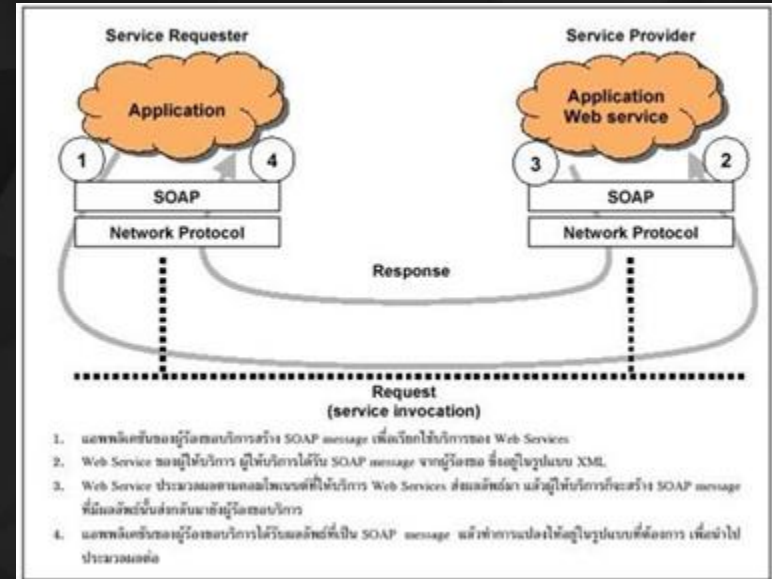
SOAP เป็นโปรโตคอลที่ผู้จัดหา Web Services เลือกใช้ที่จะส่ง message ระหว่าง Web Services SOAP เป็น Transport Protocol ที่มี XML เป็นพื้นฐานและใช้ HTTP เป็นโปรโตคอลร่วมในการส่งผ่านเครือข่าย SOAP จะระบุวิธีในการเข้ารหัสส่วนหัว (Header Encoding) ของทั้ง HTTP และไฟล์ XML ให้อย่างชัดเจนทั้งใน ส่วนของการติดต่อไปยังคอมพิวเตอร์อีกเครื่องหนึ่งและส่งผ่านข้อมูลไปให้รวมถึงระบุวิธีที่โปรแกรมซึ่งถูกเรียกนั้นจะส่งค่าคืนกลับมาด้วย



web services using SOAP on the Internet/Cloud

บริษัทไมโครซอฟท์, ไอบีเอ็ม, โลตัส, ยูสเซอร์แลนด์ (UserLand) และ ดีเวลลอปเปอร์เมนเตอร์ (DeveloperMenter) ได้ร่วมกันกำหนดมาตรฐานของ SOAP ขึ้น ซึ่งต่อมาได้มีบริษัทอีก 30 กว่าบริษัทเข้าร่วมและ จัดตั้งเป็น W3C XML Protocol Workgroup

จุดเด่นของ SOAP ก็คือเป็นโปรโตคอลที่เป็นกลาง กล่าวคือ ไม่มีใครเป็นเจ้าของและเป็นโปรโตคอล ที่ทำงานกับโปรโตคอลอื่นหลายชนิด การพัฒนาก็อนุญาตให้ทำได้อย่างอิสระตามแพลตฟอร์ม ระบบปฏิบัติการ แบบจำลองทางวัตถุ (Object model) และภาษาโปรแกรมของผู้ที่ทำการพัฒนา





web services using SOAP on the Internet/Cloud



สถาปัตยกรรมของ **SOAP**

แบ่งสถาปัตยกรรมภายในออกเป็น 3 ส่วนประกอบ ดังรูป



web services using SOAP on the Internet/Cloud

SOAP Message

SOAP message มีลักษณะการสื่อสารทางเดียวจากแอปพลิเคชันที่ส่งไปยังแอปพลิเคชันที่รับข้อมูล โดย message อาจประกอบไปด้วยคำร้องขอข้อมูลจากแอปพลิเคชันที่เป็นตัวรับเมื่อคำร้องขอได้รับการประมวลผลก็จะส่งผลลัพธ์ย้อนกลับเป็น SOAP message อีกอย่างหนึ่ง ดังนั้นจึงมี SOAP message อยู่หลายตัวสื่อสารอยู่ระหว่างแอปพลิเคชันบนรีโมตคอมพิวเตอร์

เอกสาร SOAP นั้นมีโครงสร้างในรูปแบบ XML ซึ่งเราสามารถแบ่งเป็นส่วนของเอกสารได้เป็น 3 ส่วนหลัก ดังนี้คือ

1. SOAP envelop เนื้อหาสาระ (Content) ของเอกสารทั้งหมด
2. SOAP header ส่วนเพิ่มเติมของเอกสาร SOAP ซึ่งจะมีก็ได้ หรือไม่มีก็ได้
3. SOAP body ส่วนที่ใช้ในการเรียกใช้งานเซอร์วิส และผลลัพธ์ที่ได้จากเซอร์วิส





web services using SOAP on the Internet/Cloud

ตัวอย่างเอกสาร SOAP

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <soap:Header>
    <!-- ข้อมูลในส่วนของ Header -->
    <i:local xmlns:i="http://www.i3t.or.th/ws/">
      <i:currency>Bath</i:currency>
    </i:local>
  </soap:Header>
  <soap:Body>
    <!-- ข้อมูลในส่วนของ Body -->
    <GetPrice>
      <Item>Rose</Item>
      <Quantity>100</Quantity>
    </GetPrice>
  </soap:Body>
  <soap:Fault>
```

```
<!-- ข้อมูลของ SOAP ในกรณีที่มีข้อผิดพลาดจาก SOAP Node -->
    </soap:Fault>
  </soap:Envelope>
  <soap:Fault>
    <faultcode>Client</faultcode>
    <faultstring>Invalid Request</faultstring>
  </soap:Fault>
  <soap:Fault>
    <faultcode>Client</faultcode>
    <faultstring>Invalid Request</faultstring>
  </soap:Fault>
```



web services using SOAP on the Internet/Cloud

Element	Definition
<Envelope>	<p>จะเป็น Root ของเอกสาร XML SOAP เสมอ โดยจากตัวอย่างจะใช้ Namespaces จาก "http://schemas.xmlsoap.org/soap/envelope/"</p> <p>(Schema for the SOAP/1.1 envelope)</p> <p>ดังนั้น Element ต่างๆ ในตัวอย่างนี้ที่อยู่ในเอกสาร XML ส่วนที่เป็นมาตรฐานของ SOAP เราจะมี Namespaces ที่เราตั้งเป็น soap (<soap:Envelop>, <soap:Header>, <soap:Body>, <soap:Fault>)</p> <p>Attribute ใน Envelop</p> <p>Attribute encodingStyle ในตัวอย่างมีการอ้างการ encoding จาก "http://schemas.xmlsoap.org/soap/encoding/"</p> <p>(Schema for the SOAP/1.1 encoding)</p> <p>จะใช้ในกรณีที่เรามีการใช้ Parameter หลายชนิดข้อมูลในเอกสาร (Data Type) เช่น Boolean, String, Integer เป็นต้น</p>



web services using SOAP on the Internet/Cloud

<Header>

จากตัวอย่างมีการเพิ่มเนื้อหาของเอกสาร SOAP เข้าไปในส่วนของ Header ในที่นี้คือ สกุลเงิน (Currency) ซึ่งส่วนนี้จะใช้ในส่วนของการ Application ไม่ได้เป็นส่วนหนึ่งของมาตรฐานของ SOAP แต่ผู้ใช้งานเป็นผู้กำหนดใช้เอง (User-defined Attribute ใน Header

ใน <Header> Element อาจจะมีการใส่ Attribute mustUnderstand เพื่อให้ฝั่งที่รับเอกสาร SOAP ให้ทำการประมวลผลในส่วน Header ด้วย โดยกำหนดค่าเป็น Boolean เช่น

```
<i:local xmlns:i="http://www.i3t.or.th/ws/">
```

```
    <i:country mustUnderstand="1">Thailand</i:country>
```

```
    <i:currency mustUnderstand="0">Bath</i:currency>
```

```
</i:local>
```

ค่าของ mustUnderstand ถ้าเป็น "0" แสดงว่าไม่ต้องประมวลผล

ถ้าเป็น "1" : true คือจำเป็นต้องมีการประมวลผล (โดยค่า default = "0" : fault)



web services using SOAP on the Internet/Cloud

<Body>	โดยปกติแล้วเอกสาร SOAP จะต้องมีส่วนของ Body เพราะเป็นส่วนเป็นเนื้อหาสาระจริงๆ ของ SOAP จากตัวอย่างเราต้องการสอบถามราคาของดอกกุหลาบจำนวน 100 ดอก ซึ่ง <GetPrice>, <Item> และ <Quantity> เป็น Element ที่ใช้งานใน Application ไม่ใช่มาตรฐานของ SOAP
<Fault>	<p>เป็นส่วนที่จะถูกใช้เมื่อมีข้อผิดพลาดในการประมวลผลของเอกสาร SOAP ซึ่งโดยปกติจะเห็นเฉพาะที่เป็นเอกสารตอบกลับเท่านั้น (Reply Message หรือ Answer Message) ซึ่งตัวอย่างจะเป็นเอกสารที่เป็นการร้องขอบริการ (Request Message) ซึ่งไม่น่าจะมีส่วนของ <Fault> Element แต่ที่เขียนไว้เพื่อให้เห็นภาพรวมของโครงสร้างเอกสาร SOAP ทั้งหมดก่อน ตัวอย่างข้อมูลใน <Fault> Element ที่มีการเกิดข้อผิดพลาด</p> <pre data-bbox="546 764 1812 982"><soap:Fault> <faultcode>Client</faultcode> <faultstring>Invalid Request</faultstring> </soap:Fault></pre>



web services using SOAP on the Internet/Cloud

SOAP Serialization

เมื่อ **SOAP Message** ถูกส่งข้ามเครือข่ายเราจะต้องแน่ใจว่าเมสเสจนั้นอยู่ในรูปแบบที่แอปพลิเคชันที่เป็นตัวรับสามารถแปลความหมายและเข้าใจได้ง่าย สิ่งนี้ทำให้เราต้องทำการ **Serialize** เนื้อหาใน **Header** และ **Body** ของ **SOAP Message** ให้อยู่ในรูปแบบที่แอปพลิเคชันที่เป็นตัวรับเมสเสจเข้าใจ โดยใช้เนมสเปซ **System.XML.Serialization** เพื่อทำการ **Serialize** ข้อมูลให้เป็น **XML** ซึ่งแอปพลิเคชันทุกตัวสามารถทำความเข้าใจได้ รูปแบบของการ **Serialization** ตามข้อกำหนดของ **SOAP** แบ่งออกเป็น 2 รูปแบบได้แก่

1. **Literal** รูปแบบนี้ข้อมูลต้องเป็นไปตามข้อกำหนดของ **XML Schema** หรือ **XSD** ซึ่งไม่มีกฎเกณฑ์พิเศษ
2. **encoding** รูปแบบนี้จะเป็นไปตามกฎเกณฑ์พิเศษในข้อกำหนดของ **SOAP**

รูปแบบของ **SOAP Message** ข้อกำหนดของ **SOAP** กำหนดรูปแบบของ **SOAP Message** ไว้ 2 รูปแบบได้แก่

1. **Document Style** รูปแบบนี้อิลิเมนต์ **Body** จะประกอบด้วยอิลิเมนต์ย่อยตั้งแต่หนึ่งอิลิเมนต์หรือมากกว่า เรียกว่า **parts** อย่างไรก็ตามในข้อกำหนดของ **SOAP** ไม่ได้กำหนดกฎพิเศษสำหรับอิลิเมนต์ย่อย จึงสามารถใช้ข้อมูลใดๆก็ได้ที่ต้องการส่งไปยังแอปพลิเคชันที่เป็นตัวรับข้อมูล
2. **RPC Style** รูปแบบนี้อิลิเมนต์ **Body** จะประกอบด้วยเมธอดหรือวีโมดโปรซีเจอร์ที่ต้องการเรียก รวมทั้งพารามิเตอร์ที่ต้องการผ่านไปด้วย



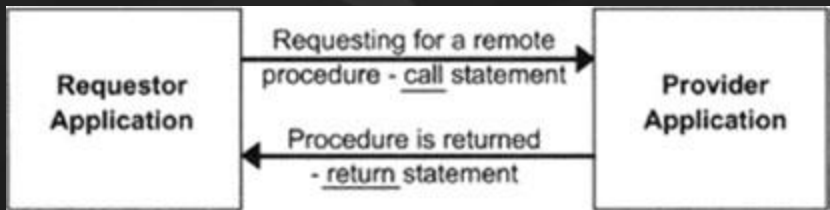
web services using SOAP on the Internet/Cloud

SOAP RPC

พิจารณาเหตุการณ์ที่เราต้องการเรียกโปรซีเจอร์ในรีโมตคอมพิวเตอร์ ในขณะที่เราเรียกเราต้องส่งผ่านพารามิเตอร์มายังโปรซีเจอร์ตานั้นด้วย กระบวนการเรียกโปรซีเจอร์จากรีโมตคอมพิวเตอร์เรียกว่า **Remote Procedure Call** หรือ **RPC**

คำร้องขอให้โปรซีเจอร์สามารถทำได้โดยใช้ **SOAP** ในรูปแบบของ **SOAP Message** ซึ่งเมสเสจประกอบด้วยคำร้องขอหรือผลลัพธ์ตอบกลับคำร้องขอนั้น การกำหนดวิธีการแลกเปลี่ยนเมสเสจเรียกว่า **RPC Covention** ซึ่งจะกำหนดกฎเกณฑ์ที่เราสามารถใช้ในการ **Serialize SOAP message**

รีโมตโปรซีเจอร์ เราต้องส่ง **request** ไป คำสั่งที่ใช้ในการส่ง **Request** รีโมตโปรซีเจอร์เรียกว่าคำสั่ง **Call** และเมื่อ **Request** ได้รับการประมวลผลก็จะส่ง **Response** กลับมา เรียกว่า **Result**





web services using SOAP on the Internet/Cloud

SOAP Encoding Rules

เมื่อเราสร้าง SOAP Message เราจะต้องปฏิบัติตามกฎระเบียบที่กำหนดไว้ในข้อกำหนดของ SOAP ซึ่งเรียกว่า SOAP Encoding โดยกฎระเบียบนี้มีลักษณะคล้ายกันกับ XML Schema ที่จะกำหนดชนิดและโครงสร้างที่เราจะใช้ในการสร้าง SOAP Message โดยกฎระเบียบนี้จะปรากฏอยู่ในเนมสเปซ encoding ที่มี URL <http://schemas.xmlsoap.org/soap/encoding/>

ชนิดของข้อมูล

ใน encoding rules จะกำหนดชนิดข้อมูลที่เราสามารถใช้ใน SOAP Message เป็น 2 ชนิดคือ

1. **simple** เป็นข้อมูลที่ตามข้อกำหนดของ XML Schema ที่เราสามารถนำมาใช้กับ SOAP Message ได้แก่ integer, float, negativeInteger, string, date, เป็นต้น โดยข้อมูลชนิดนี้กำหนดไว้ในเนมสเปซ <http://www.w3.org/1999/XMLSchema>
2. **compound** สร้างมาจากข้อมูล simple โดย SOAP encoding รองรับข้อมูล compound อยู่ 2 ชนิดคือ structs และ arrays



web services using SOAP on the Internet/Cloud

struct เป็นกลุ่มของตัวแปรที่เป็นชนิดเดียวกันหรือต่างกัน ดังตัวอย่าง

```
struct Student {  
    string ID;  
    string FirstName;  
    string LastName;  
    string Address;  
    int Age;  
};
```

การเรียกใช้ struct เราต้องสร้างอินสแตนซ์ของโครงสร้างขึ้นมาเช่น

```
Student std = {"S001", "Daniel", "Smith", "10932 Bigge Rd.", 15};
```

และสามารถ Serialize Struct Student ด้วย SOAP encoding rules ได้ดังนี้

```
<std xsi:type="n:Student">  
  <ID xsi:type="xsd:string">S001</ID>  
  <FirstName xsi:type="xsd:string">Daniel</FirstName>  
  <LastName xsi:type="xsd:string">Smith</LastName>  
  <Address xsi:type="xsd:string">10932 Bigge Rd.</Address>  
  <Age xsi:type="xsd:integer">15</Age>  
</std>
```



web services using SOAP on the Internet/Cloud

Array เป็นข้อมูล compound ที่ประกอบไปด้วยข้อมูล simple ชนิดเดียวกัน การกำหนด array ต้องใช้แอตทริบิวต์ `xsi:type` ซึ่งมีค่าเป็น `SOAP-ENC:Array` ดังตัวอย่างข้างล่าง

```
<scores xsi:type="SOAP-ENC:Array" SOAPENC:arrayType="xsd:integer[5]">
  <item>96</item>
  <item>92</item>
  <item>87</item>
  <item>56</item>
  <item>70</item>
</scores>
```



web services using SOAP on the Internet/Cloud

ข้อดีของการใช้โพรโตคอล SOAP

1. โพรโตคอล SOAP สามารถให้เราเรียกใช้คอมพิวเตอร์ หรือ เว็บเซิร์ฟเวอร์ ซ้ำเครื่อง ซ้ำแพลตฟอร์มหรือซ้ำภาษา ได้ โดยอาศัยโพรโตคอลที่มีอยู่เดิมในอินเทอร์เน็ต อย่าง HTTP
2. โครงสร้างข้อมูลของ SOAP เป็นรูปแบบข้อความที่สื่อสารกันด้วยภาษา XML ทำให้แอปพลิเคชันที่เป็นตัวรับข้อมูลสามารถแปลความหมายได้ง่าย ทำให้เข้าใจได้ในทุกแพลตฟอร์ม
3. โพรโตคอล SOAP สามารถทำงานผ่านระบบไฟล์วอลล์ได้ง่ายเนื่องจาก SOAP ทำงานอยู่กับ โพรโตคอล HTTP ซึ่งไฟล์วอลล์จะเปิดให้การสื่อสารด้วย HTTP ผ่านได้อย่างสะดวก
4. SOAP ใช้ภาษา XML ที่ยอมให้สามารถส่งผ่านข้อมูลในปริมาณที่มากและทำให้เรื่องยากกลายเป็นเรื่องง่ายด้วยการเข้ารหัสข้อมูล



web services using SOAP on the Internet/Cloud

ข้อเสียของการใช้โพรโตคอล SOAP

1. เนื่องจากลักษณะของ SOAP message เป็นเอกสาร XML ทำให้เสียเวลาในการแปลกลับมาเป็นรูปแบบ
2. ที่โปรแกรมเข้าใจในกรณีที่ SOAP ทำงานอยู่กับโพรโตคอล HTTP ซึ่งมีสมรรถนะในการรับ-ส่งข้อมูลต่ำกว่าโพรโตคอล DCOM, RMI, หรือ IIOP จึงทำให้โพรโตคอล SOAP มีอัตราการรับ-ส่งข้อมูลต่ำ
3. SOAP ไม่ยอมให้ทำการตรวจสอบความผิดพลาดในเวลาที่ทำการแปลเอกสาร สามารถตรวจสอบความผิดพลาดได้เฉพาะในเวลารันเท่านั้น
4. เอกสาร XML ที่มีการกำหนดโครงสร้างด้วย DTD และ XML Schema ใช้หน่วยความจำมาก ซึ่งเป็นเรื่องใหญ่ และใช้เวลาในการประมวลผลมาก ซึ่งอาจก่อให้เกิดทรานแซกชันในระบบเครือข่ายได้และส่งผลให้เกิด Overhead
5. องค์การที่จะทำการติดต่อสื่อสารกันด้วยเว็บเซอร์วิส ต้องทำความตกลงร่วมกันบนมาตรฐานที่กำหนดโดย XML Schema



web services using WSDL on the Internet/Cloud

WSDL (Web Services Description Language)

เป็นภาษาที่ใช้อธิบายคุณลักษณะการใช้บริการของ Web Services และวิธีการติดต่อกับ Web Services ความต้องการของนิยามนี้ เกี่ยวเนื่องกับความต้องการของ distributed system ที่จะกำหนด Interface Definition Language(IDL) โดยใช้ภาษา XML, WSDL เกิดจากการรวมแนวคิดของ NASSL (The Network Accessible Service Specification Language), WDS (Well-Defined Services) ของบริษัท ไอบีเอ็ม, SDL (The Service Description Language) และ SCL (the SOAP Contract Language) ของบริษัทไมโครซอฟท์ ปัจจุบัน WSDL เป็นภาษา ที่อยู่ในการดูแลของ W3C (World Wide Web Consortium) ซึ่งยังไม่เป็นมาตรฐานที่สมบูรณ์ เวอร์ชันที่ใช้งานอยู่ใน ปัจจุบันคือ WSDL 1.1

WSDL คือ มาตรฐานสำหรับการประกาศ process ที่จำเป็นในการเรียกใช้เซอวิซ SOAP (Simple Object Access Protocol) โครงสร้างเอกสาร WSDL

WSDL เป็นภาษาที่อยู่ในความดูแลขององค์กร W3C (World Wide Web Consortium) version ที่มีอยู่ในปัจจุบัน คือ WSDL 1.1 ใน การใช้งานจริง หากเราสร้างบริการ Web Services ก็จะมีเครื่องมือช่วยสร้างเอกสาร WSDL สำหรับ Web Services อย่างอัตโนมัติ



web services using WSDL on the Internet/Cloud

จุดภายในเอกสารที่เราควรรู้เกี่ยวกับการติดต่อและเรียกใช้บริการของ Web Services มีจุดที่ควรรู้ ดังนี้

Element	Definition
<port Type>	เป็นส่วนที่สำคัญที่สุดในWSDL element อธิบาย operations ที่ web service มีให้บริการและ messages ที่เกี่ยวข้อง เทียบได้กับ function library หรือ module หรือ class ในการเขียนโปรแกรม
<operation>	อธิบาย method ที่ให้บริการ Web Services หนึ่งจะมี method จำนวนกี่ method ก็ได้
<message>	อธิบาย data elements ของ operation แต่ละ message อาจมีมากกว่าหนึ่งส่วนเทียบได้กับ parameter ของ function ในการเขียนโปรแกรม
<types>	อธิบายชนิดข้อมูลที่ web serviceใช้ เพื่อความเป็นกลาง WSDL ใช้ XML Schema syntax ในการระบุชนิดข้อมูล
<binding>	อธิบาย format ของ message และ protocol details ในแต่ละ port
<service>	สำหรับ web server จะมี Web Services จำนวนกี่บริการก็ได้ และ ชื่อ Web Services ก็เป็นตัวจำแนกและบ่งบอกแต่ละบริการซึ่งห้ามมีชื่อซ้ำกัน



web services using WSDL on the Internet/Cloud

ตามทฤษฎีแล้ว ไฟล์เอกสาร WSDL แต่ละไฟล์ สามารถอธิบายคุณลักษณะของบริการ Web Services ได้มากกว่า 1 บริการโดยแต่ละ Web Services จะมี port สื่อสารเฉพาะตัว ซึ่งบ่งบอกไว้ในเอกสาร WSDL อยู่แล้ว

มาตรฐานอื่นๆ ของเว็บเซอร์วิส

มาตรฐาน WSDL SOAP และ UDDI เป็นเพียงมาตรฐานพื้นฐานของเว็บเซอร์วิส การพัฒนาเว็บเซอร์วิสในทางปฏิบัติจำเป็นต้องพิจารณาเรื่องอื่นเช่น ความปลอดภัย Transaction หรือ Messaging เป็นต้น ดังแสดงในรูปซึ่งแสดงตัวอย่างมาตรฐานเว็บเซอร์วิสอื่นๆ ตามฟังก์ชันของการทำงาน โดยจะมีมาตรฐานที่สำคัญ ดังต่อไปนี้

WS-Addressing: มาตรฐานที่ใช้ร่วมกับ SOAP Header ในการระบุโพรโทคอลการสื่อสารและระบบข่าวสาร (Messaging Systems)

WS-Security: มาตรฐานที่เป็นโครงสร้าง (Framework) เพื่อเชื่อมต่อกับเทคโนโลยีระบบความปลอดภัยต่างๆ

SAML: Security Assertion Markup Language เป็นมาตรฐานที่ทาง OASIS กำหนดขึ้นเพื่อสนับสนุนการทำ Single Sign On (SSO) และ Authentication

WS-BPEL: มาตรฐานสำหรับการประกอบ (orchestration) กระบวนการทางธุรกิจ (Business Process) โดยใช้คำสั่งที่เป็นภาษา XML

WSRP: Web Services for Remote Portal มาตรฐานสำหรับการเรียกใช้ Web Services จากเว็บท่า (Portal)

Web Services Interoperability

web services using WSDL on the Internet/Cloud

Access	Portal WSRP			
Management	Management WS-Management, WS-DistributedManagement (WSDM)			
Integration	Business Collaboration Languages: Choreography WS-CDL, ebXML BP			
	Business Process Languages: Orchestration WS-BPEL*, WFMC XPDL			
Quality of Services	Reliable Messaging: WS-Reliability WS-ReliableMessaging	Security WS-Security, WS-Trust SAML, XACML, XKMS XML-Signature XML-Encryption Liberty ID-FF, ID-WSF, ID-SIS WS-Federation WS-I Basic Security Profile	Transaction WS-AtomicTransaction WS-TXM (WS-CAF)	Policy WS-Policy
			Coordination WS-Coordination WS-CF (WS-CAF)	Context WS-Context (WS-CAF)
Discovery	Registry UDDI, ebXML RS		Repository ebXML RS/RIM	
Description	Service Description Language WSDL			
Messaging	Addressing & Notifications WS-Addressing, WS-Notification, WS-Topics, WS-Eventing			
	Messages SOAP, WS-MetadataExchange, XML, XML Schema, Encoding			
Transport	Protocols HTTP, IIOP, Other			

เนื่องจากเว็บเซอร์วิส คือเทคโนโลยีที่เป็นอิสระจาก platform หรือภาษาที่ใช้ในการพัฒนา ดังนั้นสาระสำคัญประการหนึ่งของการทำงานในส่วนของ **web services** ก็คือพยายามให้เกิดการทำงานร่วมกัน (Interoperability) ระหว่าง **services** ที่ถูกพัฒนาจากหน่วยงานต่างๆ ให้ได้มากที่สุด ซึ่งหน่วยงานที่เกิดจากการประสานงานของหลายฝ่ายอันที่จะพยายามสร้างมาตรฐาน ของ **Interoperability** ระหว่างเว็บเซอร์วิสก็คือ **Web Services Interoperability Organization (WS-I)** สิ่งที่ WS-I นำเสนอก็คือมาตรฐานแห่งการเชื่อมต่อระหว่างผู้ให้บริการ ซึ่งกลุ่มทำงานภายใน WS-I ได้ออกมาตรฐานที่เกี่ยวข้องในรูปแบบของ **Profile** ซึ่งเป็นการให้ข้อเสนอแนะเกี่ยวกับการใช้ข้อกำหนดทางเทคนิคอันที่จะทำให้เกิด **Interoperability** ได้เหมาะสมที่สุด

ในสถานะปัจจุบันสิ่งที่ **profile** ที่ WS-I ได้กำหนดออกมาประกอบด้วย **Basic Profile Attachments Profile Simple SOAP binding profile** และกำลังอยู่ในระหว่างการจัดทำ **Basic Security Profile** โดยเบื้องต้นแนวทางในการทำการทดสอบเว็บเซอร์วิส ของกลุ่มงาน **Web Technology GITI** จะยึดรูปแบบตาม **Profile** ที่ WS-I กำหนดเอาไว้



Examples web services using SOAP and WSDL on the Internet/Cloud

รายชื่อและข้อมูลที่สำคัญของเว็บเซอร์วิสที่น่าสนใจ

เว็บเซอร์วิสดิกชันนารี ภาษาอังกฤษที่อยู่เว็บเซอร์วิส

<http://services.aonaware.com/DictService/DictService.asmx>

ที่อยู่เอกสาร **WSDL** <http://services.aonaware.com/DictService/DictService.asmx?WSDL>

ประเภทของเว็บเซอร์วิส **SOAP**

เว็บเซอร์วิสแปลภาษา

ที่อยู่เว็บเซอร์วิส

<http://www.websvcex.net/TranslateService.asmx>

ที่อยู่เอกสาร **WSDL**

<http://www.websvcex.net/TranslateService.asmx?WSDL>

ประเภทของเว็บเซอร์วิส **SOAP**



Examples web services using SOAP and WSDL on the Internet/Cloud

หมวดหมู่ดินฟ้าอากาศและเวลาทั่วโลก
อากาศทั่วโลก (Global Weather)

ที่อยู่เว็บเซอร์วิส

<http://websvcx.com/globalweather.asmx>

ที่อยู่เอกสาร WSDL

<http://websvcx.com/globalweather.asmx?WSDL>

ประเภทของเว็บเซอร์วิส SOAP

Assignment 1

Report #2

“หนึ่งหน่วยกลุ่ม” (Group 1)

Selected Topics in Computer Engineering II



Then create web services for your group XML data

Library: SOAPpy 0.12.22

SOAP Services for Python

SOAPpy provides tools for building SOAP clients and servers.

INTRODUCTION

The goal of the SOAPpy team is to provide a full-featured SOAP library for Python that is very simple to use and that fully supports dynamic interaction between clients and servers.

INCLUDED

General SOAP Parser based on sax.xml

General SOAP Builder

SOAP Proxy for RPC client code

SOAP Server framework for RPC server code





Then create web services for your group XML data

FEATURES

- **Handles all SOAP 1.0 types**
- Handles faults
- Allows namespace specification
- Allows SOAPAction specification
- Homogeneous typed arrays
- Supports multiple schemas
- Header support (mustUnderstand and actor)
- XML attribute support
- Multi-referencing support (Parser/Builder)
- Understands SOAP-ENC:root attribute
- Good interop, passes all client tests for Frontier, SOAP::LITE, SOAPRMI
- Encodings
- SSL clients (with Python compiled with OpenSSL support)
- SSL servers (with Python compiled with OpenSSL support and M2Crypto installed)
- Encodes XML tags per SOAP 1.2 name mangling specification (Gregory Warnes)
- Automatic stateful SOAP server support (Apache v2.x) (blunck2)
- **WSDL client support**
- **WSDL server support**

Library: SOAPpy 0.12.22



Then create web services for your group XML data

Server

```
from SOAPpy import SOAPServer
import xml.etree.ElementTree as ET
import sys, time
from SOAPpy import WSDL
class webservice:
    def return_xml():
        tree = ET.parse('MovieAll_SPN.xml')
        root = tree.getroot()
        return ET.tostring(root,encoding="us-ascii", method="xml")
    def movieNameFromType(self,type):
        list_ = []
        tree = ET.parse('MovieAll_SPN.xml')
        root = tree.getroot()
        for movie in root.findall('movie'):
            for type_ in movie.find('types'):
                if type_.text == type:
                    list_.append(movie.find('name').text)
        return list_
```



Then create web services for your group XML data

```
def update_xml(self, name, type, name_actor, Director, day, month, year):  
    tree = ET.parse('MovieAll_SPN.xml')  
    root = tree.getroot()  
    new_tag = ET.Element('movie')  
    name_tag = ET.Element('name')  
    name_tag.text = name  
    types_tag = ET.Element('types')  
    for t in type:  
        type_tag = ET.Element('type')  
        type_tag.text = t  
        types_tag.append(type_tag)  
    stars_tag = ET.Element('stars')  
    for star in name_actor:  
        actor_tag = ET.Element('name_actor')  
        actor_tag.text = star  
        stars_tag.append(actor_tag)
```

1.

```
director_tag = ET.Element('director_actor')  
director_tag.text = Director  
date_tag = ET.Element('date')  
day_tag = ET.Element('day')  
month_tag = ET.Element('month')  
year_tag = ET.Element('year')  
date_tag.append(day_tag)  
date_tag.append(month_tag)  
date_tag.append(year_tag)  
new_tag.append(name_tag)  
new_tag.append(types_tag)  
new_tag.append(stars_tag)  
new_tag.append(director_tag)  
new_tag.append(date_tag)  
root.insert(0, new_tag)  
tree.write('MovieAll_SPN.xml')
```

2.

```
print "start"  
server = SOAPServer(("localhost", 8081))  
server.registerObject(webService(), "xml")  
server.serve_forever()
```

3.



Then create web services for your group XML data

Start Server

```
File Edit Shell Debug Options Window Help
Python 2.7.12 (v2.7.12:d33e0cf91556, Jun 27 2016, 15:19:22) [MSC v.1500 32 bit
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Admin\Desktop\test_soappy.py =====
start
```



Then create web services for your group XML data

Client: ใส่ข้อมูลข้อมูลหนังประเภท Adventure และ Sci-fi ให้ XML

```
import sys
from SOAPpy import SOAPProxy
from SOAPpy import WSDL
import xml.etree.ElementTree as ET
serverUrl='http://localhost:8081'
namespace='xml'
server = SOAPProxy(serverUrl, namespace)
server.config.dumpSOAPIn = 1
server.update_xml("movie name",["Adventure","Sifi"],["name1","name2"],"name3","1","oct","1234")
response = server.movieNameFromType("Adventure")
print response
```



Then create web services for your group XML data

Result

```
*****
<<class 'SOAPpy.Types.typedArrayType'> Result at 39387440>: ['movie name', 'movie name',
'movie name', 'The Last Airbender', 'The Sorcerer's Apprentice', 'Captain America: Civil
War', 'Suicide Squad', 'Justice League', 'Rogue One: A Star Wars Story', 'Star Trek Beyon
d', 'Batman v Superman: Dawn of Justice', 'The Avenger', 'The Avenger 2 : Age of Altron',
'Ironman', 'Ironman 2', 'Ironman 3', 'The Incredible Hulk', 'Thor', 'Thor The Dark World'
, 'Ant Man', 'Captain America: The Winter Soldier', 'In the Heart of the Sea', 'The Reven
ant', 'Interstellar', 'The Martian', 'Back to the Future', 'Back to the Future II', 'Back
to the Future III', 'Batman Begins']
>>>
```



Then create web services for your group XML data

SOAP → in

```
*** Incoming SOAP *****
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema"
>
<SOAP-ENV:Body>
<movieNameFromTypeResponse SOAP-ENC:root="1">
<Result SOAP-ENC:arrayType="xsd:string[29]" xsi:type="SOAP-ENC:Array">
<item xsi:type="xsd:string">movie name</item>
<item xsi:type="xsd:string">movie name</item>
<item xsi:type="xsd:string">movie name</item>
<item xsi:type="xsd:string">Back to the Future III</item>
<item xsi:type="xsd:string">Batman Begins</item>
</Result>
</movieNameFromTypeResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



Then create web services for your group XML data

SOAP → out

```
*** Outgoing SOAP *****
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema"
>
<SOAP-ENV:Body>
<ns1:movieNameFromeType xmlns:ns1="xml" SOAP-ENC:root="1">
<v1 xsi:type="xsd:string">Adventure</v1>
</ns1:movieNameFromeType>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
*****
```



Then create web services for your group XML data

Client: อัปเดตข้อมูล XML

```
import sys
from SOAPpy import SOAPProxy
from SOAPpy import WSDL
import xml.etree.ElementTree as ET
serverUrl='http://localhost:8081'
namespace='xml'
server = SOAPProxy(serverUrl, namespace)
server.update_xml("movie name",["Adventure","Sifi"],["name1","name2"],"name3","1","oct","1234")
response = server.movieNameFromType("Adventure")
print response
```



Then create web services for your group XML data

Result

```
===== RESTART: C:\Users\Admin\Desktop\test_soappy_update.py =====  
<<class 'SOAPpy.Types.typedArrayType'> Result at 39922288>: ['movie name', 'movie  
e name', 'movie name', 'movie name', 'The Last Airbender', 'The Sorcerer's Appre  
ntice', 'Captain America: Civil War', 'Suicide Squad', 'Justice League', 'Rogue  
One: A Star Wars Story', 'Star Trek Beyond', 'Batman v Superman: Dawn of Justice  
, 'The Avenger', 'The Avenger 2 : Age of Altron', 'Ironman', 'Ironman 2', 'Iron  
man 3', 'The Incredible Hulk', 'Thor', 'Thor The Dark World', 'Ant Man', 'Captai  
n America: The Winter Soldier', 'In the Heart of the Sea', 'The Revenant', 'Inte  
rstellar', 'The Martian', 'Back to the Future', 'Back to the Future II', 'Back t  
o the Future III', 'Batman Begins']
```



Then create web services for your group XML data

SOAP → in

```
*** Incoming SOAP ****
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
>
<SOAP-ENV:Body>
<update_x005F_xmlResponse SOAP-ENC:root="1">
<Result xsi:null="1"/>
</update_x005F_xmlResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
*****
```



Then create web services for your group XML data

SOAP → out

```
*** Outgoing SOAP *****
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema"
>
<SOAP-ENV:Body>
<ns1:update_xml xmlns:ns1="xml" SOAP-ENC:root="1">
<v1 xsi:type="xsd:string">movie name</v1>
<v2 SOAP-ENC:arrayType="xsd:string[2]" xsi:type="SOAP-ENC:Array">
<item>Adventure</item>
<item>Sifi</item>
</v2>
<v3 SOAP-ENC:arrayType="xsd:string[2]" xsi:type="SOAP-ENC:Array">
<item>name1</item>
<item>name2</item>
</v3>
<v4 xsi:type="xsd:string">name3</v4>
<v5 xsi:type="xsd:string">1</v5>
<v6 xsi:type="xsd:string">oct</v6>
<v7 xsi:type="xsd:string">1234</v7>
</ns1:update_xml>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
*****
```

● References



web services using SOAP
on the Internet/Cloud

<https://www.gotoknow.org/posts/161670>

<http://www.rightsoftcorp.com/>

<http://inforzone.kktech.ac.th/>

<http://www.mindphp.com/>

Members

- 1. นาย อภิวัฒน์ วงศ์โพธิ์ *รหัสนักศึกษา* 52-1116-530-2
- 2. นางสาว พัลลภา เขมรังสฤษฎ์ *รหัสนักศึกษา* 56-010126-2008-1
- 3. นางสาว อัญธิกา หนองบัว *รหัสนักศึกษา* 56-010126-3028-1
- 4. นาย ธนดล เตชะวัชรกุล *รหัสนักศึกษา* 56-010126-3009-4
- 5. นาย ชีรวัฒน์ ผ่องสกุล *รหัสนักศึกษา* 56-010126-3015-9
- 6. นาย ภูมิพล ไชยเชิดเกียรติ *รหัสนักศึกษา* 56-010116-2131-8

Question & Answer



THANKYOU

