บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

1. ทฤษฎีพื้นฐาน

1.1 Web Service [1]

1.1.1 ตัวบริการเว็บ (Web Service) เป็นเทคโนโลยีที่ทำให้โปรแกรมประยุกต์ต่าง ๆ ที่พัฒนาขึ้นโดยมีพื้นฐานทางด้านภาษา และสถาปัตยกรรมที่แตกต่างกัน สามารถสื่อสารแลกเปลี่ยนข้อมูลกันได้โดยใช้ XML [2] เป็นภาษากลางในการสื่อสารแลกเปลี่ยนข้อมูล และทำงานร่วมกันอยู่บนมาตรฐานอื่นอีกเช่น SOAP [3], WSDL [4] และ UDDI [5] ซึ่งจะกล่าวถึงในหัวข้อถัดไป และได้สรุปความหมายของตัวบริการเว็บเป็นข้อ ๆ ดังนี้

(1) เป็นเทคโนโลยีที่สามารถใช้งานร่วมกันได้ทุกแพลตฟอร์ม

(2) โปรแกรมประยุกต์ต่าง ๆสามารถติดต่อสื่อสารกันโดยใช้ XML เป็นสื่อกลาง

(3) ถูกเรียกใช้งานจาก Application อื่น ๆในรูปแบบ RPC (Remote Procedure Call)

(4) การให้บริการจะมีเอกสารที่อธิบายคุณสมบัติของบริการกำกับไว้ (WSDL : Web Services Description Language)

(5) เป็นมาตรฐานเปิดที่ทำให้เราสามารถ "ประกาศ" (Publish) ว่ามี "บริการ" ใดบ้าง และ "Web Application" อื่น สามารถเข้าไปสืบค้น และ เรียกใช้บริการจาก services นั้นได้ (UDDI : Universal Description, Discovery and Integration)

(6) ใช้มาตรฐาน SOAP ในการสื่อสาร ซึ่งสามารถทำงานร่วมกับโพรโทคอลอื่น เช่น HTTP, SMTP

1.1.2 **ประโยชน์ของตัวบริการเว็บ**

(1) การดำเนินธุรกิจการค้าและบริการระหว่างองค์กรสามารถเป็นไปได้ แบบอัตโนมัติ โดยการแลกเปลี่ยนข้อมูลผ่านทางตัวบริการเว็บ

(2) พันธมิตรทางการค้าสามารถเกิดขึ้นได้ตลอดเวลา โดยการค้นหาจาก UDDI

(3) ตัวบริการเว็บใช้ XML สำหรับอธิบายข้อมูลโดย XML เป็นข้อมูลในรูปของ text file ที่คนอ่านกันเข้าใจและไม่ขึ้นกับ platform หรือระบบปฏิบัติการ และด้วยความข้อมูลที่เป็น text file ธรรมดาทำให้มีความยืดหยุ่นในการใช้งาน รวมทั้งการเขียนโปรแกรมที่เกี่ยวข้องก็จะทำได้ง่าย

(4) ตัวบริการเว็บช่วยให้การ Integrate หรืออย่างน้อยก็การแลกเปลี่ยนข้อมูลทำได้ง่ายขึ้น เนื่องจากการเขียนโปรแกรมเพื่อทำงานกับตัวบริการเว็บ สามารถเขียนด้วยภาษาอะไรก็ได้ และไม่ขึ้นกับ Hardware และระบบปฏิบัติการ

(5) ตัวบริการเว็บสามารถใช้เป็นส่วนหนึ่งในการดำเนินธุรกิจตามเฟรมเวิร์คของ ebXML

(6) ง่ายต่อการนำไปใช้งานเนื่องจากในปัจจุบันมีเครื่องมือมากมายที่ใช้ช่วยเหลือในการพัฒนาตัวบริการเว็บ

(7) ลดต้นทุนในการพัฒนาระบบบางอย่างที่ไม่จำเป็นโดยขอบริการจากตัวบริการเว็บของพันธมิตรทางการค้า

1.1.3 เครื่องมือที่ใช้ในการพัฒนาตัวบริการเว็บ

ตัวบริการเว็บเป็นมาตรฐานที่ง่ายต่อการพัฒนาเนื่องจากอยู่ในรูปแบบของข้อมูลตัวอักษร มีชุดเครื่องมือที่สนับสนุนการสร้าง Web Services หลายตัว เช่น

(1) ชุดเครื่องมือที่สนับสนุนโดย Microsoft ตามแพลตฟอร์มของ Microsoft .NET Framework เช่น Microsoft SOAP Toolkit 3.0 [6]

(2) ชุดเครื่องมือที่สนับสนุนโดย Sun Microsystem ตามแพลตฟอร์มของ Sun ONE (Sun Open Net Environment) [7]

(3) ชุดเครื่องมือที่สนับสนุนโดย IBM เช่น Web Services Toolkit [8], WebSphere Studio Application Developer [9]

(4) เครื่องมืออื่น ๆที่สนับสนุน SOAP, XML ทั้งที่เป็น Commercial Product และ Open Source

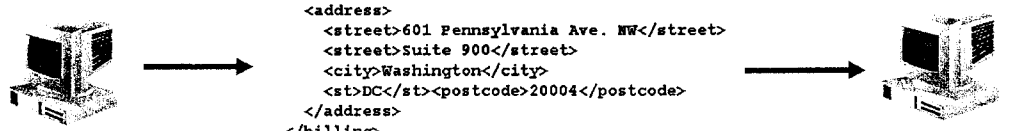
1.2 มาตรฐานที่เกี่ยวข้องกับตัวบริการเว็บ

ตัวบริการเว็บ (Web Service) เป็นเทคโนโลยีที่สามารถทำให้ซอฟต์แวร์แลกเปลี่ยนข้อมูลร่วมกันได้ ทำให้เกิดผลลัพธ์ที่สามารถทำงานร่วมกันระหว่างแพลตฟอร์มที่ต่างกัน เพื่อเป็นประโยชน์ต่อผู้ใช้ มาตรฐานเทคโนโลยีหลักที่เกี่ยวข้องกับตัวบริการเว็บ คือ XML SOAP WSDL และ UDDI ซึ่งแต่ละมาตรฐาน สามารถอธิบายได้ดังนี้

1.2.1 XML (Extensible Markup Language)

เป็นภาษามาตรฐานที่ถูกออกแบบมาเพื่อให้เป็นภาษากลางสำหรับแลกเปลี่ยนข้อมูลระหว่างกัน มีลักษณะข้อมูลเป็น text-based โดยเป็นภาษาที่ไม่ติดอยู่กับแพลตฟอร์มหรือโปรแกรมเฉพาะใด ๆ (Platform and Application Independent) ดังนั้นระบบที่อยู่ต่างแพลตฟอร์มกัน ใช้ฐานข้อมูลต่างกัน หรือมีสถาปัตยกรรมที่ต่างกันก็สามารถแลกเปลี่ยนข้อมูลกันได้ โดยผู้ใช้เองสามารถกำหนดแท็ก (Tag) ที่ใช้สื่อความหมายของเนื้อหา กำหนดโครงสร้างของเอกสาร และยังสามารถนำ xml นั้นไปประมวลผลต่อในแอพพลิเคชั่นอื่นได้

XML จะอาศัยโพรโทคอลที่ชื่อว่า SOAP (Simple Object Access Protocol) ซึ่งเปรียบเสมือนกติกา หรือข้อตกลงในการสื่อสารบนเครือข่ายอินเทอร์เน็ต




```
<?xml version="1.0" ?>
<?xml:stylesheet?>
<purchase-order>
  <header>
    <po-number>1234</po-number>
    <date>1999-02-08</date><time>14:05</time>
  </header>
  <billing>
    <company>XMLSolutions</company>
    <address>
      <street>601 Pennsylvania Ave. NW</street>
      <street>Suite 900</street>
      <city>Washington</city>
      <st>DC</st><postcode>20004</postcode>
    </address>
  </billing>
  <order items="1">
    <item>
      <reference>097251</reference>
      <description>Widgets</description>
      <quantity>4</quantity>
      <unit-price>11.99</unit-price>
      <price>47.46</price>
    </item>
    <tax type="sales">
      <tax-unit>VA</tax-unit>
      <calculation>0.045</calculation>
      <amount>2.16</amount>
    </tax>
    ...
  </order>
</purchase-order>
```

ภาพที่ 1 ตัวอย่าง XML Purchase Order

1.2.2 SOAP (Simple Object Access Protocol)

เนื่องจากจุดประสงค์หลักของการใช้งานตัวบริการเว็บคือ ต้องการให้โปรแกรมประยุกต์ติดต่อสื่อสารกันได้ผ่านทางระบบเครือข่าย ซึ่งเทคโนโลยีที่มีอยู่ในปัจจุบันที่ใช้มีการสื่อสารระหว่างออปเจ็กต์ในระยะไกล (Remote Procedure Calls : RPC) เช่น DCOM, EJB หรือ CORBA ไม่ได้ถูกออกแบบมาใช้สำหรับโพรโทคอล HTTP

เทคนิค RPC ของเทคโนโลยีที่กล่าวข้างต้นนั้นมีปัญหาในด้านการนำมาใช้งานในแง่ของความเข้ากันได้ของการเรียกใช้งานข้ามเทคโนโลยี เนื่องจากเป็นเทคโนโลยีเฉพาะของแต่ละค่าย (ยกเว้น CORBA) ทำให้ผู้พัฒนาระบบต้องพัฒนาโปรแกรมที่มีความซับซ้อน และยังมีปัญหาในส่วนของ Firewalls และ Proxy เซิร์ฟเวอร์ เนื่องจากโดยปกติเซิร์ฟเวอร์จะปิดการสื่อสารที่ไม่ใช่โพรโทคอล HTTP เพื่อความปลอดภัยของระบบที่มีการติดต่อสื่อสารกับภายนอก

ดังนั้นทางเลือกของการสื่อสารที่จะนำมาใช้ในการทำตัวบริการเว็บ คือทำงานอยู่บนโพรโทคอล HTTP และเป็นมาตรฐานเปิดที่ทำให้สามารถติดต่อสื่อสารกับเครื่องคอมพิวเตอร์ที่มีความแตกต่างกันทั้งระบบปฏิบัติการ เทคโนโลยี รวมไปถึงภาษาที่ใช้ในการพัฒนาด้วย

SOAP เป็นกลไกการแลกเปลี่ยนข้อมูลระหว่างกันแบบกระจายจากศูนย์กลาง โดยใช้โครงสร้างของ XML เป็นพื้นฐาน สามารถนำไปใช้กับระบบได้หลากหลายตั้งแต่ระบบข้อความ (Messaging System) จนถึง RPC

SOAP ประกอบด้วย 3 ส่วน คือ

(1) SOAP Envelop สำหรับใช้ในการระบุสิ่งที่อยู่ในเอกสาร การจัดการเอกสาร และบอกความจำเป็นในการใช้งาน

(2) SOAP Encoding Rule สำหรับกำหนดกลไกที่ใช้ในการแลกเปลี่ยนข้อมูล

(3) SOAP RPC Representation สำหรับนิยามรูปแบบ RPC และ response

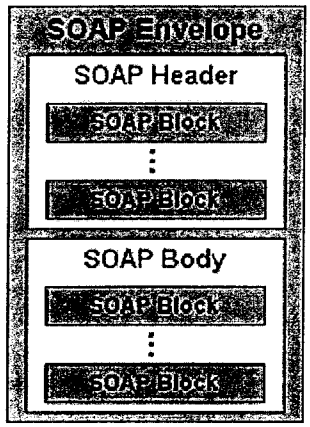
โครงสร้างของ SOAP

เอกสาร SOAP นั้นมีโครงสร้างในรูปแบบ XML ซึ่งเราสามารถแบ่งเป็นส่วนของเอกสารได้เป็น 3 ส่วนหลักดังนี้คือ

(1) SOAP envelop เนื้อหาสาระ (Content) ของเอกสารทั้งหมด

(2) SOAP header ส่วนเพิ่มเติมของเอกสาร SOAP ซึ่งจะมีก็ได้ หรือไม่มีก็ได้

(3) SOAP body ส่วนที่ใช้ในการเรียกใช้งานเซอร์วิส และผลลัพธ์ที่ได้



ภาพที่ 2 โครงสร้างของเอกสาร SOAP

1.2.3 WSDL (Web Services Description Language)

คิดค้นโดยบริษัท IBM และ Microsoft เป็นภาษาที่ใช้อธิบายคุณลักษณะการให้บริการของตัวบริการเว็บ และวิธีการติดต่อขอรับบริการจากตัวบริการเว็บ เช่น ชื่อตัวบริการเว็บ ชื่อเมทธอดของ COM Component ที่เปิดให้บริการ พารามิเตอร์ที่ส่งไปยังเมทธอดชนิดข้อมูลของพารามิเตอร์ เป็นต้น โดยรายละเอียดเหล่านี้จะเป็นไปตามไวยากรณ์ของภาษา XML (Extensible Markup Language)

SOAP UDDI และ WSDL เป็นแกนหลักสำคัญของตัวบริการเว็บ สิ่งสำคัญในการเรียกใช้งานตัวบริการเว็บ คือต้องรู้ที่อยู่จาก UDDI รู้คุณลักษณะของตัวบริการเว็บนั้น ๆจากเมทธอดของตัวบริการเว็บและพารามิเตอร์ต่าง ๆที่จำเป็นจากเอกสาร WSDL และร้องขอบริการ รอผลลัพธ์จากผู้ให้บริการด้วยโพรโทคอล SOAP

ความสัมพันธ์ระหว่าง 3 ส่วนนี้ สามารถแสดงได้ดังนี้

(1) SOAP เป็นโพรโทคอลสื่อสารที่อาศัยไวยากรณ์ของภาษา XML และทำงานกับโพรโทคอลอื่น ๆได้หลายชนิด เนื่องจากใช้ไวยากรณ์ XML จึงทำให้ทำงานได้ทุกแพลตฟอร์ม ดังนั้นจึงสามารถเรียกใช้คอมโพเนนต์ข้ามแพลตฟอร์มได้

(2) WSDL เป็นเอกสารที่ใช้อธิบายคุณลักษณะของตัวบริการเว็บ โดยใช้ไวยากรณ์ของภาษา XML

(3) UDDI เปรียบเสมือนฐานข้อมูลที่เก็บรายละเอียดของตัวบริการเว็บไว้ เพื่อให้ผู้ใช้บริการค้นหาบริการ ในกรณีของผู้ให้บริการต้องนำข้อมูลเกี่ยวกับตัวบริการเว็บของตนไปเก็บไว้ใน UDDI

โครงสร้างของ WDSL

เอกสาร WSDL นั้นแบ่งออกเป็น 2 ส่วนใหญ่ ๆ คือ

(1) Abstract Definitions Group

ทำหน้าที่กำหนด message SOAP ที่ไม่ขึ้นกับแพลตฟอร์มหรือรูปแบบภาษา กล่าวคือ ไม่บรรจุส่วนประกอบที่มีเฉพาะในเครื่องบางเครื่อง หรือภาษาบางภาษา โดยที่ส่วนประกอบของ abstract นั้น จะประกอบไปด้วย type message และ port type

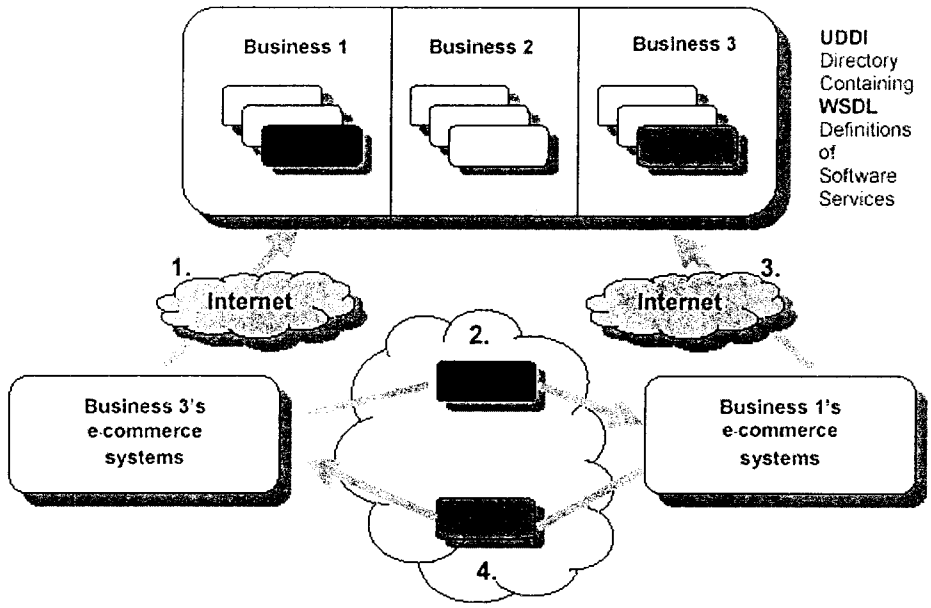
(2) Concrete Descripions Group

ทำหน้าที่ตรงกันข้ามกับส่วนของ abstract Definitions คือ เป็นข้อมูลที่ขึ้นอยู่กับเว็บไซต์ เครื่องและภาษา โดยประกอบด้วย binding และ service

1.2.4 UDDI (Universal Description, Discovery and Integration)

เป็นมาตรฐานในการค้นหาบริการต่าง ๆของ Web Services ทำหน้าที่เป็นตัวกลางเพื่อให้ Provider มาลงทะเบียนไว้ โดยจะมีไฟล์ WSDL สำหรับบอกรายละเอียดของการบริการ เพื่อให้ Requestor สามารถค้นหาและเรียกใช้การบริการของ Web service นั้น ๆได้ โดยดูรายละเอียดของการบริการและวิธีการเรียกใช้จากไฟล์ WSDL ซึ่งเปรียบได้กับสมุดโทรศัพท์นั่นเอง

UDDI เป็นมาตรฐานที่จัดตั้งขึ้นโดย IBM, Microsoft และ Ariba ปัจจุบันมีบริษัทที่ร่วมกันกำหนดมาตรฐานของ UDDI มากกว่า 70 บริษัท ซึ่งมาตรฐานของ UDDI ถูกกำหนดให้เป็นมาตรฐานสำหรับ B2B interoperability



ภาพที่ 3 ภาพจำลองการทำงานของตัวบริการเว็บ ที่เกิดจากการทำงานร่วมกันของ SOAP, UDDI และ WSDL [1]

ภาพจำลองการทำงานร่วมกันของ SOAP, UDDI และ WSDL จะแสดงได้ดังภาพที่ 3 ดังรายละเอียดต่อไปนี้

(1) Business 3 ค้นหาการให้บริการของ Business 1 ใน UDDI ซึ่งรายละเอียดจะอยู่ใน WSDL ผ่านอินเทอร์เฟสของ Business 1

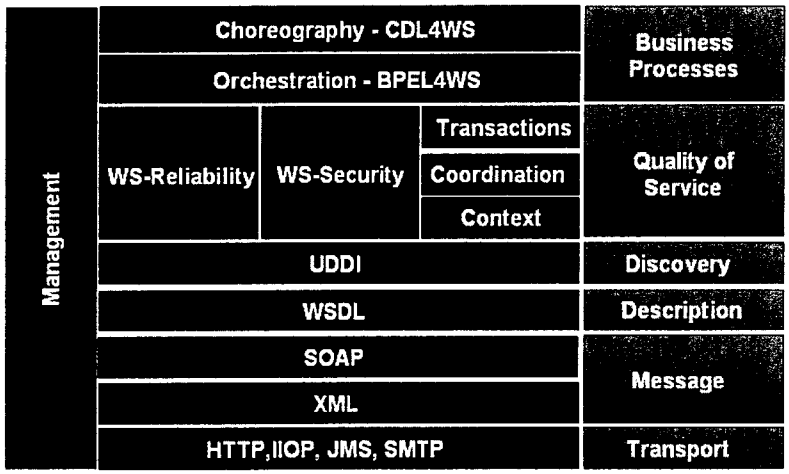
(2) ติดต่อผ่านอินเทอร์เฟสของแอพพลิเคชั่นที่ให้บริการ (ด้วย SOAP message) ผ่านทางอินเทอร์เน็ต

(3) และ 4. Business 1 รวมหรือ Integration กับ Business 3 ด้วยวิธีเดียวกัน

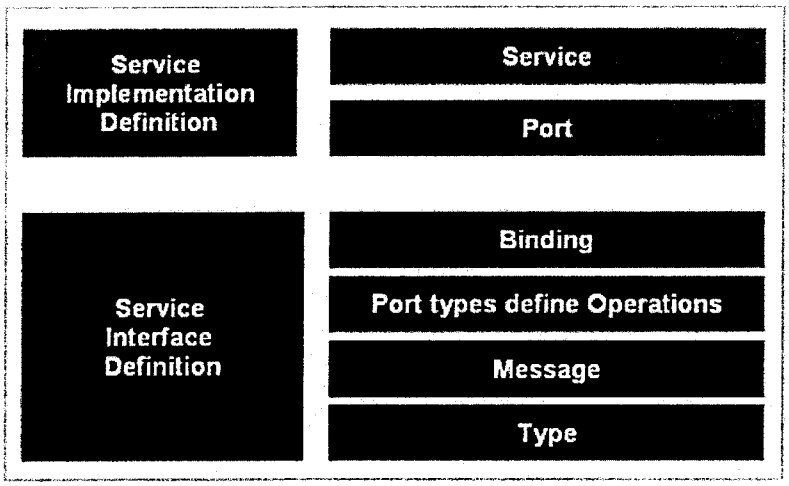
1.2.5 BPEL4WS (Business Process Execution Language for Web Services) [10]

BPEL4WS คือมาตรฐานที่ใช้กำหนดถึงการทำงานร่วมกันของตัวบริการเว็บ โดยจะกำหนดถึงวิธีการ และลำดับขั้นตอนของการดำเนินการร่วมกัน เกิดขึ้นจากความพยายามที่จะสร้างมาตรฐานร่วมกันระหว่าง มาตรฐาน WSFL [11] ของ IBM และมาตรฐาน XLANG [12] ของ Microsoft เพื่อให้เกิดมาตรฐานที่ดีที่สุดโดยหยิบยกเอาข้อดีของทั้งสองมาตรฐานมารวมกัน

BPEL4WS เป็น Top-level ของ WSDL ตามระดับชั้นมาตรฐานดังภาพที่ 4 ใช้รูปแบบไวยากรณ์ของภาษา XML ในการที่จะกำหนดถึงลำดับการทำงานในแต่ละ process ของตัวบริการเว็บ จะดูได้จากไฟล์ WSDL ของแต่ละตัวบริการเว็บ ที่ได้ประกาศไว้บน UDDI ว่าบริการนี้จะต้องส่งค่าอะไรไปบ้าง มาสร้างเป็นกระบวนการทางธุรกิจ (Business Process) ซึ่งจะได้เป็นนามสกุล .bpel (ดังภาพที่ 5) ส่วนเครื่องมือที่ใช้ในการพัฒนาอย่างเช่น WebSphere Studio Application Developer ของ IBM หรือ Oracle JDeveloper ของ Oracle



ภาพที่ 4 Standards Building Blocks of BPEL [10]



ภาพที่ 5 BPEL Depends on WSDL and WSDL Extensions [10]

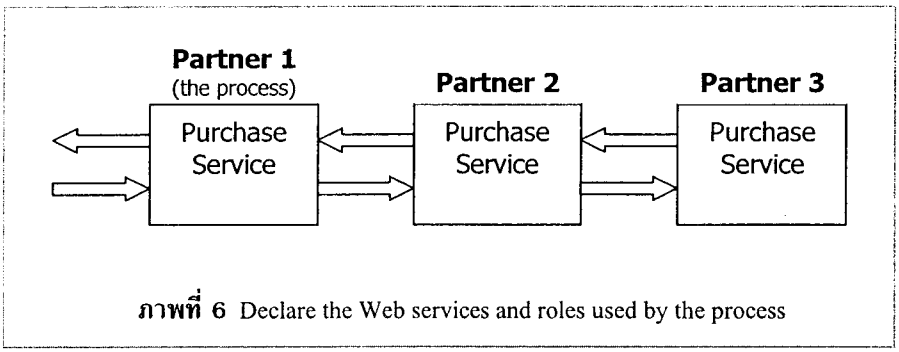
(1) ตัวอย่างโครงสร้างของไฟล์ BPEL (BPEL Scenario Structure)

```
<process>
  <!– Definition and roles of process participants -->
        <partnerLinks> ... </partnerLinks>
  <!- Data/state used within the process -->
  <variables> ... </variables>
  <!- Properties that enable conversations -->
  <correlationSets> ... </correlationSets>
  <!- Exception handling -->
  <faultHandlers> ... </faultHandlers>
        <!- Error recovery – undoing actions  -->
  <compensationHandlers> ... </compensationHandlers>
  <!- Concurrent events with process itself  -->
  <eventHandlers> ... </eventHandlers>
  <!- Business process flow  -->
  (activities)*
</process>
```

(2) กลุ่ม Element Activities ของไฟล์ BPEL แบ่งออกเป็น 2 กลุ่ม คือ

Basic Activities	**Structured Activities**
• <invoke>	• <sequence>
• <receive>	• <switch>
• <assign>	• <pick>
• <reply>	• <flow>
• <throw>	• <link>
• <terminate>	• <while>
• <wait>	• <scope>

(3) Partners Element จะระบุถึง service ของผู้ร่วมทำธุรกิจทั้งหมดว่ามี service ใดบ้าง และมี Attributes กำหนดถึง Service Link Type ว่าคืออะไรเพื่อโยงไปถึง WSDL และบอกถึงบทบาทหน้าที่ว่าทำอะไร ดังตัวอย่างนี้



ภาพที่ 6 Declare the Web services and roles used by the process

- ตัวอย่างไฟล์ BPEL :

```
<partnerLinks>
 <partnerLink name="customer" serviceLinkType="lns:purchasePLT"
   myRole="purchaseService"/>
 <partnerLink name="inventoryChecker" serviceLinkType="lns:inventoryPLT"
   myRole="inventoryRequestor" partnerRole="inventoryService"/>
 <partnerLink name="creditChecker" serviceLinkType="lns:creditPLT"
 myRole="creditRequestor" partnerRole="creditService"/>
</partnerLinks>
```

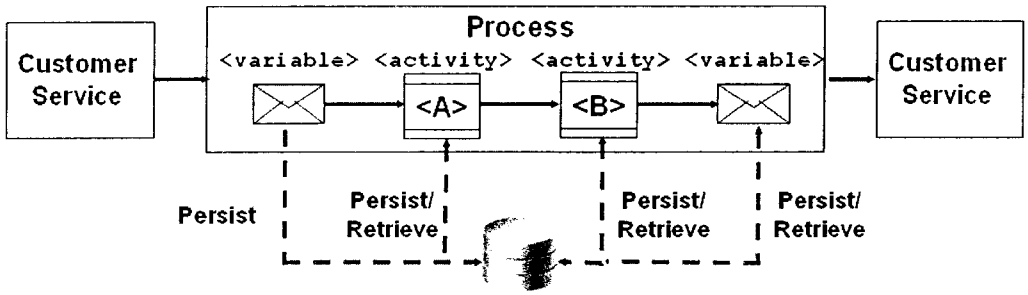
- Purchase Process WSDL:

```
<plt:partnerLinkType name="purchasePLT">
 <plt:role name="purchaseService">
  <plt:portType name="tns:purchasePT"/>
 </plt:role>
</plt:partnerLinkType>
```

- Purchase Process PortType:

```
<portType name="purchasePT">
 <operation name="sendPurchase">
 </operation>
</portType>
```

(4) Variables Element จะกำหนดถึง messages ที่ใช้รับส่งถึงกันระหว่างตัวบริการเว็บ



ภาพที่ 7 Messages sent and received from partners

- ตัวอย่างไฟล์ BPEL :

```
<variables>
 <variable name="PO" messageType="lns:POMessage"/>
 <variable name="Invoice" messageType="lns:InvMessage"/>
 <variable name="POFault" messageType="lns:orderFaultType"/>
</variables>
```

- Purchase Process WSDL:

```
<message name="POMessage">
  <part name="customerInfo" type="sns:customerInfo"/>
  <part name="purchaseOrder" type="sns:purchaseOrder"/>
</message>
<message name="InvMessage">
  <part name="IVC" type="sns:Invoice"/>
</message>
<message name="orderFaultType">
 <part name="problemInfo" type="xsd:string"/>
</message>
```

(5) การจัดการโยกย้ายข้อมูลระหว่าง variables สามารถใช้ <assign> และ <copy> ซึ่ง element <copy> สนับสนุนการสืบค้นข้อมูลแบบ XPath ด้วย ตัวอย่างเช่น

```
<assign>
 <copy>
  <from variable="PO" part="customerInfo"/>
  <to variable="creditRequest" part="customerInfo"/>
 </copy>
</assign>
```

(6) Activities

Basic Activities เช่น

- invoke ส่ง message เพื่อเรียกใช้บริการจาก partner ใด แบ่งได้เป็น 2 ประเภท คือ Synchronous และ Asynchronous
- Receive รอรับ messages จาก partner
- Reply ตอบกลับ message
- Assign จัดการ copy ข้อมูลจากที่หนึ่งไปอีกที่หนึ่ง
- throw บอกถึงเมื่อมีข้อผิดพลาดเกิดขึ้น

- Wait หยุดรอการประมวลผล
- Empty กำหนดเป็นค่าว่าง
- Terminate สิ้นสุดการประมวลผลของ activities นั้น ๆ

Structured Activities เช่น

- Sequence จัดการกับ Activities ต่าง ๆ ให้ประมวลผลตามลำดับ
- Flow จัดการกับ Activities ต่าง ๆ ให้ประมวลผลไปพร้อม ๆกัน
- While กำหนดการวนซ้ำ
- Switch การควบคุม logic ของ sequence ย่อย

(7) ตัวอย่างของ BPEL Code ต่อไปนี้ ประกอบด้วย 3 บริการ โดย Process แรกคือ ServiceA เรียกใช้แบบ Synchronous ส่วน ServiceB และ ServiceC เรียกใช้แบบ Asynchronous ประมวลผลแบบขนาน (flow) โดยรับ Output ของ ServiceA มาเป็น input แล้วทำการประมวลผล สุดท้ายจึงส่งผลลัพธ์กลับมาที่ Client

```
<process name="test">
    <partnerLinks>
        <partnerLink name="client"/>
        <partnerLink name="serviceA"/>
        <partnerLink name="serviceB"/>
        <partnerLink name="serviceC"/>
    </partnerLinks>
    <variables>
        <variable name="processInput"/>
        <variable name="AInput"/>
        <variable name="AOutput"/>
        <variable name="BCInput"/>
        <variable name="BOutput"/>
        <variable name="COutput"/>
        <variable name="processOutput"/>
        <variable name="AError"/>
    </variables>
    <sequence>
        <receive name="receiveInput" variable="input"/>
        <assign>
            <copy>
                <from variable="processInput"/>
                <to variable="AInput"/>
            </copy>
        </assign>
        <scope>
            <faultHandlers>
                <catch faultName="faultA" faultVariable="AError"/>
            </faultHandlers>
            <sequence>
                <invoke name="invokeA" partnerLink="serviceA" inputVariable="AInput" outputVariable="AOutput"/>
            </sequence>
        </scope>
        <assign>
            <copy>
                <from variable="AOutput"/>
                <to variable="BCInput"/>
            </copy>
        </assign>
        <flow>
            <sequence>
                <invoke name="invokeB" partnerLink="serviceB" inputVariable="BCInput"/>
                <receive name="receive_invokeB" partnerLink="serviceB" variable="BOutput"/>
            </sequence>
            <sequence>
                <invoke name="invokeC" partnerLink="serviceC" inputVariable="BCInput"/>
                <receive name="receive_invokeC" partnerLink="serviceC" variable="COutput"/>
            </sequence>
        </flow>
        <switch>
            <case>
                <!-- assign value to processOutput -->
            </case>
        </switch>
        <invoke name="reply" partnerLink="client" inputVariable="processOutput"/>
    </sequence>
</process>
```

ภาพที่ 8 ตัวอย่างของไฟล์ BPEL

2. งานวิจัยที่เกี่ยวข้อง

2.1 Web-Services Modeling for E-Marketplace [13]

2.1.1 บทนำ

ในปัจจุบันการทำธุรกิจร่วมกันระหว่างผู้ซื้อและผู้ขาย หรือระหว่างผู้ขายด้วยกัน ไม่ว่าจะเป็นด้านการผลิต การจัดหา การขาย การนำส่ง และการจัดซื้อในสายการผลิต ต้องเสียเวลาในการกำหนดข้อตกลงซึ่งต้องทำร่วมกันเพื่อให้เป็นแนวทางหรือแบบแผนเดียวกัน

เว็บเซอร์วิสทำให้เกิด e-marketplace ซึ่งเป็นพื้นฐานริเริ่มของการซื้อขายสินค้าหรือบริการระหว่างผู้ซื้อและผู้ขาย หรือระหว่างผู้ขายด้วยกันเอง และในปัจจุบันมีแนวโน้มที่เพิ่มมากขึ้นเรื่อย ๆที่จะนำเทคโนโลยีของเว็บเซอร์วิสมาใช้ในทำธุรกิจการค้าอิเล็กทรอนิกส์ (E-Commerce) แต่ก็ยังมีข้อบกพร่องในการจัดการ e-marketplace ที่ดี ซึ่งเป็นปัญหาที่ต้องเปลี่ยนแปลงและพัฒนาให้ดีขึ้น ซึ่งในงานวิจัยนี้ได้สร้างแบบจำลองเว็บเซอร์วิส เพื่อเป็นแนวทางในการดำเนินการอย่างเป็นระบบในการซื้อขายสินค้า และการเข้าร่วมทำธุรกิจระหว่างผู้ขายเอง และในแบบจำลองนี้ยังแสดงให้เห็นขั้นตอนที่ลูกค้าซื้อบริการท่องเที่ยวผ่านทางตัวแทนท่องเที่ยว (Travel Agency) และกิจกรรมในการทำธุรกิจระหว่างผู้ขายเอง ตั้งแต่การจองโรงแรม การจองเที่ยวบิน การจองรถเช่า และการชำระเงินกับธนาคาร ซึ่งทำให้เกิดการเชื่อมโยงเครือข่ายธุรกิจกัน

แบบจำลองของเว็บเซอร์วิสที่ถูกคิดค้นออกแบบขึ้นมานี้ อยู่บนพื้นฐานของความเป็นจริงของการดำเนินการทางธุรกิจ ทำงานโดยอาศัยมาตรฐานเทคโนโลยีของเว็บเซอร์วิส ดังเช่น XML, SOAP, WSDL และ UDDI

งานวิจัยนี้ได้อธิบายถึงกิจกรรมของสิ่งที่เกี่ยวข้องกับระบบ พร้อมแสดงตัวอย่างประกอบ และได้นำเสนอถึงขั้นตอนการจัดการของเว็บเซอร์วิสทั้งภายในองค์กรและระหว่างองค์กร การแนะนำถึงการบริการแบบกระจายที่เป็นโครงสร้างพื้นฐานที่เหมาะสมสำหรับสนับสนุนการดำเนินการ และการอภิปรายถึงงานที่เกี่ยวข้องและสรุปผล

2.1.2 Service oriented computing context

งานวิจัยนี้มีลักษณะการทำงานแบบกระจายและใช้กับ e-business เป็นแบบจำลองที่มีเทคโนโลยีสนับสนุน มีระเบียบแบบแผน มีหลักการและทฤษฎีทางวิศวกรรม และมีสิ่งที่เกี่ยวข้องกับการจัดการองค์ความรู้ทั้งภายในและระหว่างองค์กร

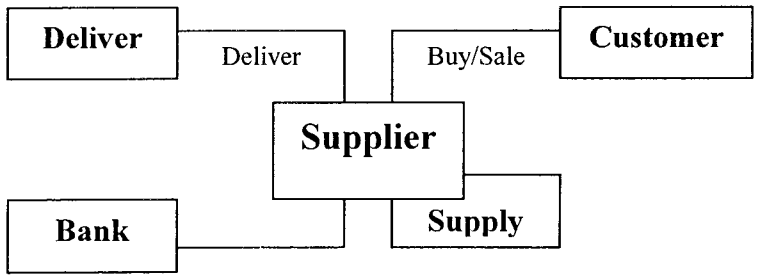
ในการจัดการกับปัญหาในงานนี้มีแบบจำลองที่เป็นระเบียบแบบแผน และโดยเฉพาะอย่างยิ่งใช้เป็นแบบแผนในการดำเนินการของเว็บเซอร์วิสทั้งภายในและระหว่างองค์กรก่อให้เกิดความสัมพันธ์ในธุรกิจระหว่างผู้ขายด้วยกันเอง ความแตกต่างของหน้าที่ของแต่ละส่วนซึ่งสามารถอธิบายได้ดังนี้ (ดูภาพที่ 9 ประกอบ)

Customer (ลูกค้าหรือผู้ซื้อ) – ทำการซื้อสินค้าหรือบริการจากบ้าน โดยผ่านทางอินเทอร์เน็ต และใช้บัตรเครดิตในการชำระค่าสินค้าหรือบริการ

Supplier (ผู้ขาย) – ทำการจัดเตรียมสินค้าหรือบริการแก่ลูกค้า รวมถึงการเชื่อมต่อกับธนาคารเพื่อชำระเงิน และติดต่อทำธุรกิจกับผู้ขายหรือผู้ผลิตรายอื่น

Bank (ธนาคาร) – รับชำระเงินโดยมีระบบ SET (Secure Electronic Transaction) สำหรับป้องกันความปลอดภัยในการจ่ายเงินของลูกค้าที่ชำระเงินผ่านทางบัตรเครดิต

Delivery (การนำส่ง) – เมื่อมีสินค้าที่ต้องการนำส่ง ทำการส่งสินค้าหรือผลิตภัณฑ์



ภาพที่ 9 องค์ประกอบที่เกี่ยวข้องกับผู้ขาย [13]

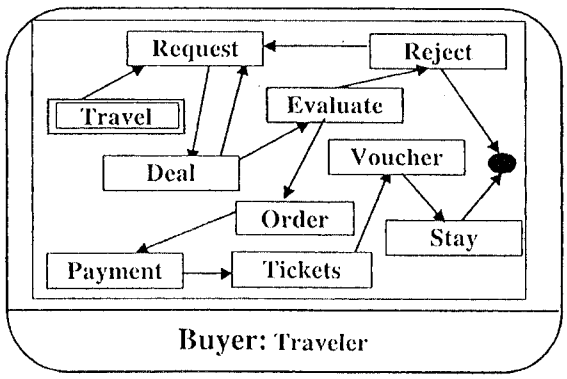
กระบวนการทำงานของแบบจำลองเว็บเซอร์วิส โดยที่ตัวแทนท่องเที่ยว (Travel Agency) จะทำหน้าที่เป็นสื่อกลางในการให้บริการลูกค้าในการท่องเที่ยว โดยลูกค้าสามารถเข้าใช้บริการได้โดยผ่านทางเว็บไซต์ โดยมีองค์ประกอบดังนี้คือ (1) ลูกค้าสั่งซื้อเป็นลักษณะของการท่องเที่ยวแบบเหมาจ่าย (Package Tour) ซึ่งในแต่ละ package ประกอบด้วย 3 ส่วนคือ ตั๋วเดินทาง สิ่งอำนวยความสะดวก และรถเช่า (2) บริษัทที่ทำธุรกิจเกี่ยวกับเที่ยวบิน (3) ธุรกิจรถเช่าสำหรับใช้ในการเดินทางเมื่อมาถึง (4) ธุรกิจโรงแรมที่พัก (5) ธนาคาร (6) ธุรกิจการขนส่ง ในกรณีที่นำส่งตั๋วเดินทาง (7) ตัวแทนการท่องเที่ยว หรือ Travel Agency เพื่อเป็นตัวกลางในการติดต่อกับลูกค้า และประสานงานทางธุรกิจกับบริษัทสายการบิน โรงแรม และบริษัทที่ให้บริการรถเช่า

การพาณิชย์อิเล็กทรอนิกส์ (E-Commerce) เป็นแนวทางหนึ่งที่องค์กรธุรกิจส่วนใหญ่หันมาใช้เพื่อเพิ่มกำไรให้กับธุรกิจ บางธุรกิจตั้งขึ้นใหม่เพื่อทำการค้าอิเล็กทรอนิกส์เพียงอย่างเดียว การทำการพาณิชย์อิเล็กทรอนิกส์ที่เห็นกันในปัจจุบัน มีลักษณะเป็น B2C (Business to Customer) คือการขายสินค้าจาก ผู้ขายที่เป็นองค์กรธุรกิจไปยังผู้ซื้อซึ่งเป็นบุคคลทั่วไป และ B2B (Business to Business) ซึ่งเป็นการติดต่อทำธุรกิจกันระหว่างทางธุรกิจ ผ่านโลกอิเล็กทรอนิกส์ ทำให้การดำเนินธุรกิจเป็นไปได้อย่างมีประสิทธิภาพมากขึ้น ซึ่งประโยชน์ อย่างแรกคือเป็นการดำเนินการทางธุรกิจระหว่างองค์กรที่เป็นไปอย่างอัตโนมัติ และอย่างที่สองลูกค้าสามารถทำการซื้อสินค้าหรือบริการได้โดยง่ายผ่านทางอินเทอร์เน็ต

2.1.3 การจัดการเกี่ยวกับการดำเนินงานทั้งภายในและระหว่างองค์กร

ในการจัดการกระบวนการทำงานมีสาระสำคัญอยู่ที่กิจกรรมของ e-marketplace ทั้งภายในและระหว่างองค์กรเอง แบบจำลองของพันธมิตรทางการค้านี้ได้แสดงถึงกิจกรรมความสัมพันธ์ที่เกิดขึ้น มีด้วยกัน 2 ส่วน ดังนี้

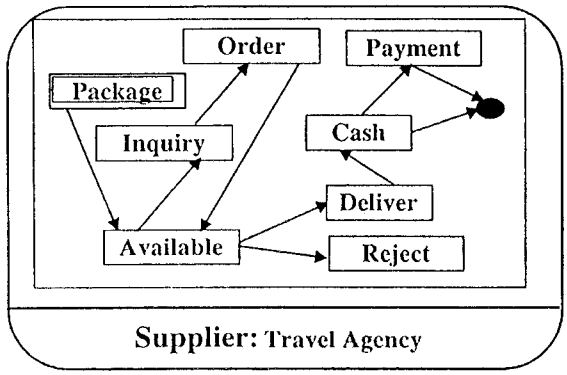
(1) กระบวนการทางธุรกรรมอิเล็กทรอนิกส์ภายในองค์กร



ภาพที่ 10 กระบวนการทางธุรกรรมอิเล็กทรอนิกส์สำหรับการจัดซื้อจัดหา [13]

กระบวนการทำงานของลูกค้าหรือผู้ซื้อจะเป็นไปตามขั้นตอนดังภาพที่ 10 นี้ โดยเริ่มจากลูกค้าต้องการที่จะท่องเที่ยวสอบถามรายการการท่องเที่ยวกับตัวแทนของบริษัททท่องเที่ยว จากนั้นตัวแทนของบริษัททท่องเที่ยวก็จะนำเสนอรายการท่องเที่ยวให้กับลูกค้า เมื่อลูกค้าสนใจก็จะตอบกลับมา จากนั้นตัวแทนของบริษัททท่องเที่ยวจะสอบถามไปยังบริษัทผู้ให้บริการต่าง ๆเช่น การจองโรงแรมที่พัก การจองเที่ยวบิน การจองรถเช่า พร้อมกับสอบถามถึงราคา แล้วจึงประเมินราคาตอบกลับลูกค้าซึ่งมีลักษณะเป็นการท่องเที่ยวแบบเหมาจ่าย (Package Tour) เมื่อลูกค้ายอมรับข้อเสนอแล้วก็ทำการชำระเงิน จากนั้นบริษัทตัวแทนก็จะส่งตั๋วเดินทางและใบเสร็จรับเงินกลับมาให้

ทางด้านของบริษัทตัวแทนท่องเที่ยวเมื่อได้รับการซื้อบริการจากลูกค้าหรือผู้ซื้อแล้ว ก็จะไปดำเนินการจัดเตรียมหาบริการให้กับลูกค้าที่ซื้อบริการกับบริษัทต่าง ๆ เช่น โรงแรม บริษัทการบิน และบริษัทขนส่ง เป็นลักษณะการทำธุรกิจ B2B



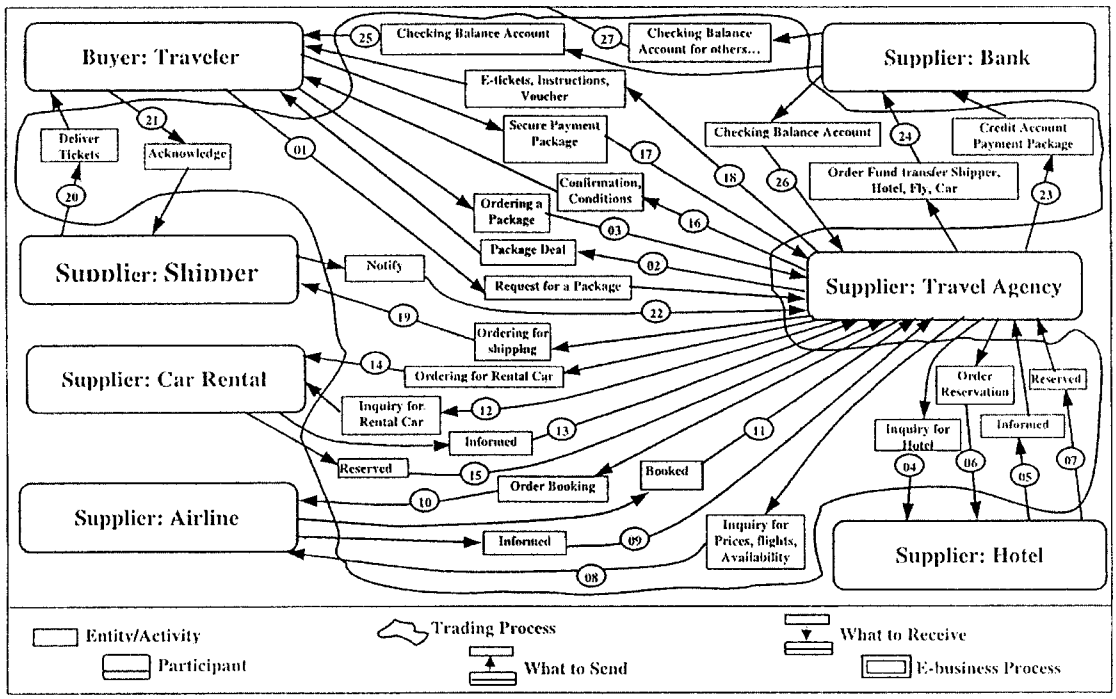
ภาพที่ 11 กระบวนการทางธุรกรรมอิเล็กทรอนิกส์สำหรับการขาย [13]

ขั้นตอนของการขาย package ท่องเที่ยวของผู้ขายหรือบริษัทตัวแทนท่องเที่ยว ซึ่งถูกกำหนดให้เป็นไปตามภาพที่ 11 คือเมื่อได้รับการร้องขอบริการจากลูกค้าแล้ว ทางบริษัทตัวแทนท่องเที่ยวต้องดำเนินการตามที่สั่งให้สมบูรณ์ ไม่เช่นนั้นลูกค้าอาจจะยกเลิกโปรแกรมท่องเที่ยวนี้ได้ และนี่คือส่วนที่สำคัญ ถ้าดำเนินการสมบูรณ์ ก็จะนำส่งสิ่งต่าง ๆ ซึ่งประกอบไปด้วยรายละเอียด เอกสารที่แสดงค่าใช้จ่าย และคำแนะนำที่จำเป็นทุก ๆอย่างสำหรับการเดินทาง ไปทางสื่ออิเล็กทรอนิกส์หรือทางจดหมาย และเมื่อลูกค้าชำระเงินแล้วบริษัทตัวแทนท่องเที่ยวก็จะจ่ายเงินไปยังบริษัทต่าง ๆ เพื่อจ่ายค่าที่พัก ค่าเที่ยวบิน และค่าเช่ารถ

(2) กระบวนการระหว่างองค์กรพันธมิตรทางการค้า

แบบจำลองของพันธมิตรทางการค้าเป็นการกำหนดและยอมรับข้อตกลงในการทำงานร่วมกันระหว่างองค์กร เป็นรูปแบบของกระบวนการทางธุรกิจที่มีความซับซ้อน ในแต่ละองค์กรได้มีความสัมพันธ์ซึ่งจะขึ้นอยู่กับหน้าที่และฟังก์ชั่นของแต่ละองค์กรในการทำงานร่วมกันบนตลาดอิเล็กทรอนิกส์ (e-marketplace) โดยอาศัยเว็บเซอร์วิสที่องค์กรได้จัดหามาทำให้เกิดการทำงานที่มีประสิทธิภาพ

จากตัวอย่างตามภาพที่ 12 เป็นลักษณะการดำเนินธุรกิจแบบ B2B โดยมีบริษัทตัวแทนท่องเที่ยวทำหน้าที่ให้บริการลูกค้าที่ต้องการท่องเที่ยว มีบริษัทการบินสำหรับจองเที่ยวบิน มีบริษัทที่ทำธุรกิจโรงแรมที่พัก มีบริษัทที่ให้บริการรถเช่า มีบริษัทขนส่งสินค้า และธนาคาร



ภาพที่ 12 Model of trading alliance, including e-business processes [13]

จากภาพที่ 12 สามารถอธิบายถึงขั้นตอนตามหมายเลขที่กำกับในแบบจำลองได้ดังนี้

① ผู้ซื้อหรือลูกค้าขอรายการท่องเที่ยวจาก Travel Agency

② Travel Agency แจ้งให้ทราบถึงรายการท่องเที่ยว

③ ส่ง order ของ package ที่ต้องการไปยัง TA

④ - ⑮ Travel Agency สอบถามถึงค่าใช้จ่ายและจองบริการกับ Supplier อื่น ๆ ทั้งโรงแรมที่พัก เที่ยวบิน และบริษัทรถเช่า

⑯ สอบถามการยืนยันจากลูกค้า โดยส่งรายละเอียดเงื่อนไขไปให้

⑰ ลูกค้าชำระเงินค่า Package

⑱ Travel Agency ออกใบรับรองการจ่ายเงินโดยใช้ระบบ E-tickets

⑲ จากนั้นส่ง order ไปยังบริษัทขนส่ง

⑳ บริษัทขนส่ง นำส่งตั๋วเดินทาง ใบเสร็จรับเงิน และเอกสารคำแนะนำที่จำเป็นทุก ๆ อย่างสำหรับการเดินทาง

㉑ ลูกค้าแจ้งว่าได้รับแล้ว

㉒ บริษัทขนส่งแจ้งกับ Travel Agency ว่าได้ส่งสินค้าให้ลูกค้าเรียบร้อยแล้ว

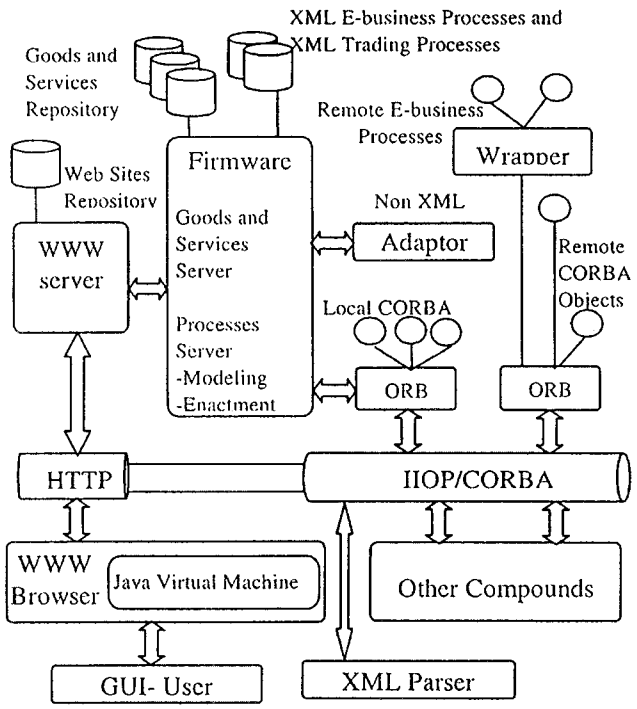
㉓ - ㉔ Travel Agency นำเงินเข้าบัญชีธนาคาร เพื่อโอนเงินให้กับ Supplier อื่น ๆ

㉕ - ㉗ ธนาคารเช็คยอดเงินของลูกค้าที่ทำธุรกรรม

2.1.4 โครงสร้างพื้นฐานและสถาปัตยกรรมที่ใช้ในการสื่อสารที่ส่งผ่านบริการต่าง ๆของระบบ

ระบบงานข้างต้นนี้จะมีความพยายามที่จะกระจายการทำงานของโปรแกรมประยุกต์บนอินเทอร์เน็ตซึ่งยังทำได้ไม่มีประสิทธิภาพมากนัก เนื่องจากแต่ละเทคโนโลยีที่ใช้ยังยึดติดกับมาตรฐานของตนเองมากเกินไป ทำให้ระบบที่ต่างกันไม่สามารถติดต่อกันได้ จึงได้มีการพัฒนามาตรฐาน Web Services ขึ้นมาเพื่อแก้ปัญหาโดยมีแนวความคิดหลักก็คือ การแลกเปลี่ยนข้อมูลซึ่งกันและกันระหว่างโปรแกรมประยุกต์ที่แตกต่างกัน

หนึ่งในคุณลักษณะของเว็บเซอร์วิสก็คือการใช้ภาษา XML ซึ่งถูกออกแบบมาเพื่อรองรับการแลกเปลี่ยนข้อมูลระหว่างกันท่ามกลางมาตรฐานที่แตกต่าง เช่น CORBA หรือ DCOM ทำให้โปรแกรมประยุกต์ทำงานร่วมกันได้อย่างมีประสิทธิภาพ ดังภาพที่ 13



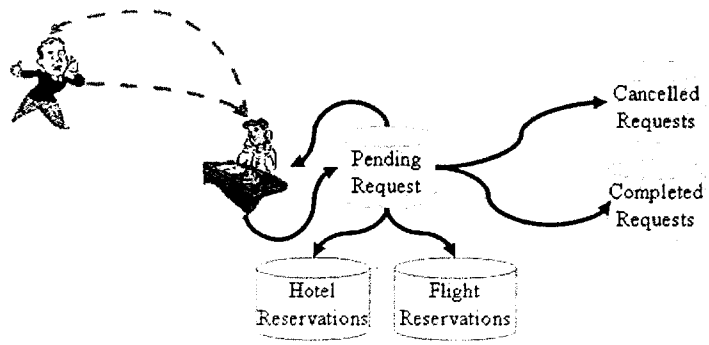
ภาพที่ 13 โครงสร้างพื้นฐานขององค์ประกอบสำหรับ e-marketplace

2.1.5 สรุปผล

มีงานวิจัยที่เกี่ยวข้องการงานวิจัยนี้หลายงาน เช่น A conceptual Architecture for a Combined Negotiation Support System [14] และ Business to Business E-Commerce [15] เพราะว่าเป็นการทำงานในลักษณะของการประมวลผลแบบกระจาย แต่ก็มีปัญหาที่ว่าไม่สนับสนุนกับงานในอนาคตที่ไม่สามารถคาดเดาสถานการณ์ได้ และไม่สามารถทำงานร่วมกันกับพันธมิตรทางการค้าอื่นผ่านทางเว็บได้ดีนัก ในขณะที่งานวิจัยนี้มีแบบจำลองแสดงถึงขั้นตอนการทำงานร่วมกันระหว่าง Suppliers อย่างชัดเจน อย่างเช่นแบบจำลองของการจัดซื้อ จัดหาสินค้า ทำให้เกิดการประสานงานกันอย่างเป็นระบบ แต่อาจยังไม่ยืดหยุ่นพอ

2.2 Web Service Composition–Current Solutions and Open Problems [16]

ทีมนักวิจัย 2 ท่านจากห้องปฏิบัติงานวิจัยของ IBM ได้ทำการศึกษาวิจัยนี้เพื่อเป็นการเปรียบเทียบและอภิบายถึงวิธีการแก้ปัญหาของการสร้างองค์ประกอบของเว็บเซอร์วิสระหว่างภาษาที่ใช้ควบคุมการไหลเวียนของข้อมูล คือ BPEL4WS [10] และ DAML-S [17] โดยได้สร้างเค้าเรื่องตัวอย่างเพื่อเป็นการประกอบการวิจัยเกี่ยวกับเรื่องจองบริการท่องเที่ยว (Package Tour) ผ่านทางตัวแทนท่องเที่ยว (Travel Agency) ดังภาพที่ 14



ภาพที่ 14 เค้าเรื่องตัวอย่างแสดงการทำงานของ Travel Agency

จากภาพที่ 14 อธิบายได้ว่าเมื่อลูกค้าต้องการที่จะเดินทางไปท่องเที่ยว จะทำการติดต่อผ่านทางตัวแทนท่องเที่ยว จากนั้นตัวแทนท่องเที่ยวซึ่งทำหน้าที่รับคำร้องขอทั้งหมดเพื่อทำการติดต่อไปยังผู้ให้บริการจองตั๋วเครื่องบิน และจองโรงแรม ตามที่ลูกค้าต้องการและจะต้องแจ้งให้ลูกค้าทราบเพื่อทำการยืนยัน ถ้าการจองล้มเหลวหรือลูกค้าต้องการยกเลิก ตัวแทนท่องเที่ยวจะติดต่อทุกผู้ให้บริการเพื่อยกเลิกทั้งหมด ลองจินตนาการดูว่าถ้าต้องติดต่อกับผู้ให้บริการอื่น ๆ จำนวนมาก ทั้งในส่วนของการจองตั๋วเครื่องบิน และการบริการที่พัก แน่นอนว่าโปรแกรมประยุกต์ของผู้ให้บริการแต่ละรายนั้นทำงานแยกกัน หรือมีระบบเป็นของตัวเอง นั้นก็ทำให้เห็นภาพที่ชัดเจนขึ้นว่า การใช้ระบบเดิม หรือระบบปิดไม่สามารถดำเนินการเพื่อสนองความต้องการที่แท้จริงของลูกค้าต่อไปได้

ดังนั้นการให้บริการรูปแบบใหม่จะต้องมีการผสานความร่วมมือระหว่างกัน โดยแต่ละส่วนต้องมีการสนองตอบต่อกันในแนวทางที่ถูกต้องหรือแนวทางเดียวกัน ในสถานการณ์ใหม่นี้ ตัวแทนการท่องเที่ยวยังคงต้องสร้างหรือวางแผนการท่องเที่ยว ซึ่งในขณะนี้มีการทำงานที่อิสระต่อกัน ซึ่งผลการติดต่อจะถูกส่งกลับมายังตัวแทนการท่องเที่ยว เพื่อตอบผลการดำเนินการว่าสำเร็จหรือล้มเหลว เมื่อการขั้นตอนทั้งหมดเสร็จสมบูรณ์ การยืนยันจะเป็นการนำเอาเอกสารทั้งหมดออกมาเพื่อให้เกิดการยืนยันทุกบริการพร้อมกันเกี่ยวกับแผนการเดินทางที่เสร็จสิ้นสมบูรณ์ แต่ถ้าบางบริการล้มเหลว ขั้นตอนการยกเลิกบริการอื่น ๆจะเกิดขึ้นเพื่อทบทวนทัวร์ดังกล่าวอีกครั้ง และการที่จะเห็นสิ่งที่เกิดขึ้นในความเป็นจริงที่การวางแผนดังกล่าวไม่ใช่เรื่องง่าย

งานวิจัยนี้ได้แสดงถึงวิธีการนำ 2 วิธีการที่ใช้ในการแก้ปัญหาในทำงานร่วมกันอย่างเป็นลำดับขั้นตอนของเว็บเซอร์วิสของแต่ละผู้ให้บริการ ดังนี้

(1) Industry Solution : WSDL + BPELWS

ส่วนที่ 1 เป็นการสร้างไฟล์ WSDL (Web Services Description Language) โดย WSDL เป็นภาษา XML ประเภทหนึ่งที่มาอธิบายวิธีการเรียกใช้เว็บเซอร์วิสซึ่งเปรียบเสมือนการอ่านคู่มือการใช้งานโปรแกรม แต่

มีข้อแตกต่างกันตรงที่ไม่เฉพาะมนุษย์เท่านั้นที่สามารถเข้าใจคู่มือ โปรแกรมที่สามารถอ่านเอกสารภาษา XML เข้าใจ ก็สามารถที่จะเข้าใจเอกสาร WSDL ได้เช่นกัน ซึ่งจากคุณสมบัตินี้ช่วยทำให้การเรียกใช้เว็บเซอร์วิสเป็นไปได้อย่างอัตโนมัติ

ในงานวิจัยนี้จะแสดงให้เห็นถึงโค้ดของ WSDL ที่ใช้ในเรียกใช้บริการเว็บเซอร์วิสจากผู้ให้บริการต่าง ๆ ผ่านทางตัวแทนท่องเที่ยว ดังนี้

```
<definitions targetNamespace="http://..." xmlns="http://schemas.xmlsoap.org/wsdl/">
    <message name = "OrderEvent"></message>
    <message name = "TripRquest"></message>
    <message name = "FlightRequest"></message>
    <message name = "HotelRequest"></message>
    <message name = "BookingFailure"></message>

    <portType name ="pt1">
        <operation name ="CToCI">
        <input message ="TripRequest"/>
        </operation>
    </portType>
    <portType name ="pt2">
        <operation name ="CIToHS">
        <output message ="HotelRequest"/>
        </operation>
    </portType>
    <portType name ="pt3">
        <operation name ="CIToFS">
        <output message ="FlightRequest"/>
        </operation>
    </portType>
    ...
    <portType name ="pt9">
        <operation name ="RIToFS">
        <output message ="BookingFailure"/>
        </operation>
    </portType>
</definitions>
```

ส่วนที่ 2 เป็นการสร้างไฟล์ BPEL เพื่อสร้าง Business Process ในธุรกิจการท่องเที่ยว

```
<process name="TripHandling">
    <partners>
      <partner name="Customer" myRole="TripHandlingAgent"
      serviceLinkType="ExternalServiceLink"
        partnerRole="CustomerAgent"/>
      <partner name="FlightService" myRole="TripHandlingAgent"
      serviceLinkType="InternalServiceLink"
        partnerRole="FlightServiceAgent"/>
      <partner name="HotelService" myRole="tripHandlingAgent"
      serviceLinkType="InternalServiceLink"
        partnerRole="HotelServiceAgent"/>
    </partners>
    <containers>
      <container name="OrderEvent" messageType="OrderEventType"/>
      <container name="TripRequest" messageType="TripRequestType"/>
      <container name="FlightRequest" messageType="FlightRequestType"/>
      <container name="HotelRequest" messageType="HotelRequestType"/>
      <container name="BookingFailure" messageType="BookingFailureType"/>
    </containers>
    <sequence>
      <receive partner="Customer" portType="pt1" operation="CToCI" container="OrderEvent"/>
      <flow>
        <invoke partner="HotelService" portType="pt2" operation="CIToHS"
            inputContainer="HotelRequest"/>
        <invoke partner="FlightService" portType="pt3" operation="CIToFS"
```

```
            inputContainer="FlightRequest"/>
        </flow>
      </sequence>
      <flow>
        <receive partner="HotelService" portType="pt4" operation="HSToEVAL1"
          container="HotelRequest"/>
        <receive partner="FlightService" portType="pt5" operation="FSToEVAL1"
          container="FlightRequest"/>
      </flow>
      <switch>
        <case condition="condition1">
          <invoke partner="Customer" portType="pt6" operation="ConIToC"
            inputContainer="TripRequest"/>
        </case>
        <otherwise>
          <flow>
            <invoke partner="Customer" portType="pt7" operation="RIToC"
              inputContainer="BookingFailure"/>
            <switch>
              <case condition="condition2">
                <invoke partner="HotelService" portType="pt8" operation="EVAL2ToHS"
                  inputContainer="BookingFailure"/>
              </case>
              <otherwise>
                <invoke partner="FlightService" portType="pt9" operation="EVAL2ToFS"
                  inputContainer="BookingFailure"/>
              </otherwise>
            </switch>
          </flow>
        </otherwise>
      </switch>
</process>
```

(2) Semantic Web Solution : RDF/DAML-s + Golog/Planning

ส่วนที่ 1 เป็นการเว็บเซอร์วิสในรูปแบบของ Semantic Web ซึ่งวัตถุประสงค์หลักของ Semantic Web คือการเพิ่มส่วนที่เรียกว่าเมตะดาต้า (Metadata) เข้าไปเป็นส่วนหนึ่งของข้อมูลออนไลน์ เพื่อเครื่องคอมพิวเตอร์จะสามารถอ่านข้อมูลเหล่านั้นและเข้าใจได้ว่าข้อมูลนั้นคืออะไรหรือเป็นส่วนไหนของชุดข้อมูล โดยแสดงออกเป็น Resource Description Format (RDF) และสามารถนำข้อมูลเหล่านั้นไปประมวลผลได้อย่างอัตโนมัติ

ในตัวอย่างนี้ จะเพิ่มในส่วนของ RDF เข้าไปตามรูปแบบมาตรฐาน DAML-S ซึ่งตัวอย่างนี้จะคล้ายกับวิธีจองตั๋วเครื่องบินกับบริษัท Bravo Air ที่ผู้ใช้คุ้นเคย ดังนี้

```
<profile:output>
    <profile:ParameterDescription rdf:ID="FlightItinerary">
    <profile:parameterName>
            FlightItinerary
    </profile:parameterName>
    <profile:restrictedTo rdf:resource=".../concepts.daml#FlightItinerary"/>
    <profile:refersTo rdf:resource="#roundTrip_In"/>
    </profile:ParameterDescription>
</profile:output>

<grounding:wsdlOutputMessageParts rdf:parseType="daml:collection">
    <grounding:WsdlMessageMap>
        <grounding:damlsParameter rdf:resource=".../FlightItinerary"/>
    <grounding:wsdlMessagePart>
            <xsd:uriReference rdf:value=".../availFlightItinerary"/>
        </grounding:wsdlMessagePart>
    </grounding:WsdlMessageMap>
</grounding:wsdlOutputMessageParts>
```

ส่วนที่ 2 เป็นโค้ดตัวอย่างของการประมวลผลซึ่งเป็นไปตามมาตรฐาน DAML-S

```
<daml:Class rdf:ID="BravoAir_Process">
 <daml:subClassOf rdf:resource=".../Process.daml#CompositeProcess"/>
  <daml:subClassOf>
   <daml:Restriction>
    <daml:onProperty rdf:resource=".../Process.daml#composedOf"/>
     <daml:toClass>
      <daml:Class>
       <daml:intersectionOf rdf:parseType="daml:collection">
        <daml:Class rdf:about="process:Sequence"/>
         <daml:Restriction>
          <daml:onProperty rdf:resource=".../Process.daml#components"/>
           <daml:toClass>
            <daml:Class>
  <process:listOfInstancesOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#GetDesiredFlightDetails"/>
    <daml:Class rdf:about="#SelectAvailableFlight"/>
    <daml:Class rdf:about="#BookFlight"/>
  </process:listOfInstancesOf>
                </daml:Class>
</daml:Class>
```

สรุปผลและการศึกษาวิจัยต่อ

จากการศึกษาเปรียบเทียบใน 2 วิธีที่ใช้แก้ปัญหาคือ the industrial approach และ the Semantic Web approach มีความสามารถในการทำ Business Process เทียบเท่ากัน แต่ใน Semantic Web นั้นมีโครงสร้างภาษาที่ซับซ้อนกว่า นอกจากนี้ผู้วิจัยพบว่าทั้ง 2 วิธีนี้ยังมีประเด็นปัญหาย่อยอื่น ๆ ที่เกี่ยวข้องและสัมพันธ์กับเรื่องของการวางแผนและนำเอาเทคโนโลยีทางด้าน AI มาใช้ ดังนั้นในการที่จะทำการศึกษาต่อ ควรที่จะมีการนำเอาเทคโนโลยีของ AI เข้ามาใช้ในการออกแบบการศึกษาวิจัยด้วย เพราะเป็นส่วนที่มีความสัมพันธ์กับลักษณะของปัญหาที่ศึกษาในครั้งนี้

2.3 A Multi-Agent Based Tourism Kiosk on Internet [18]

Chris Yeung, Pang-Fei Tung และ Jerome Yen ได้ศึกษาวิจัยระบบของบริษัทบริการการท่องเที่ยวที่อาศัยตัวแทนการท่องเที่ยวหลายตัว (Multi-Agent) ในประเทศฮ่องกง ซึ่งระบบนี้ทำให้ผู้ใช้บริการได้ดึงเอาสารสนเทศที่เป็นปัจจุบันที่สุดเกี่ยวกับฮ่องกงออกมาใช้ผ่านทางเว็บบราวเซอร์ที่สนับสนุนภาษาจาวา ระบบประกอบด้วยชุดตัวแทนซอฟต์แวร์ซึ่งจัดการกับสารสนเทศประเภทต่าง ๆ เช่น โรงแรม ศูนย์การค้า โรงภาพยนตร์ ฯลฯ โดยมีภาษา KQML เป็นภาษาที่ใช้ในการติดต่อสื่อสารและแลกเปลี่ยนสารสนเทศระหว่างตัวแทนบริการ ผู้ใช้ระบบสามารถส่งคำค้นไปยังระบบได้ แล้วระบบจะส่งต่อไปยังตัวแทนบริการท่องเที่ยวต่อไปเรื่อย ๆ จนได้ข้อมูลผลลัพธ์และดึงสารสนเทศมาให้ผู้ใช้ต่อไป

ในการติดต่อสื่อสารและแลกเปลี่ยนสารสนเทศระหว่างตัวแทนบริการท่องเที่ยวในแต่ละแห่งนั้น พบว่ายังมีข้อจำกัดอยู่เช่น ในกรณีที่ตัวแทนบริการในแต่ละแห่งนั้นมีความแตกต่างกันทางด้านเทคโนโลยีหรือภาษาในการพัฒนาตัวแทนบริการ จะทำให้เกิดปัญหาในการแลกเปลี่ยนสารสนเทศระหว่างตัวแทนบริการเองได้ ซึ่งต่างจากงานวิทยานิพนธ์นี้ ซึ่งมีการนำเอาเทคโนโลยีตัวบริการเว็บ ทำให้ลดข้อจำกัดในเรื่องการแลกเปลี่ยนข้อมูลระหว่างองค์กร ถึงแม้ว่าตัวแทนบริการจะพัฒนาจากเทคโนโลยีหรือภาษาใดก็ตาม ก็ไม่มีปัญหาในเรื่องของการติดต่อสื่อสารและและเปลี่ยนข้อมูลร่วมกันระหว่างตัวแทนบริการได้

2.4 Web Service Composition Languages: Old Wine in New Bottles? [19]

Van der Aalst, Marlon Dumas และ Arthur ter Hofstede ได้ทำการศึกษาวิจัยภาษาต่าง ๆ ที่ใช้สร้างองค์ประกอบตัวบริการเว็บ ทั้งนี้เมื่อเร็ว ๆ นี้ เกิดมีภาษาหลายภาษาสำหรับใช้สร้างองค์ประกอบตัวบริการเว็บ อย่างเช่น ภาษา BPEL4WS (Business Execution Language for Web Services) และภาษา WSCI (Web Service Choreography Interface) โดยเป้าหมายของภาษาเหล่านี้คือ เพื่อเชื่อมตัวบริการเว็บต่าง ๆ เข้าด้วยกัน ภาษาต่าง ๆ เหล่านี้ได้ยืมแนวคิดจากระบบการจัดการการไหลเวียนของงานแล้วนำแนวคิดเหล่านี้เข้าไว้ในสิ่งที่เรียกว่า Web Services Stack จนถึงปัจจุบันนี้ได้ใช้ความพยายามน้อยมากหรือไม่ได้ใช้เลยที่จะประเมินความสามารถและข้อจำกัดต่าง ๆ ของภาษาเหล่านี้อย่างเป็นระบบ เช่น ภาษา BPEL4WS กล่าวกันว่าจะรวมมาตรฐานอื่น ๆ ที่ดีที่สุดสำหรับการสร้างองค์ประกอบตัวบริการเว็บ โดยดึงเอาข้อดีจากภาษา WSFL (Web Services Flow Language) ของบริษัทไอบีเอ็ม และภาษา XLANG ของบริษัทไมโครซอฟต์ ทำให้ได้การผสมผสานกันเป็นรูปแบบกระบวนการที่เป็นโครงสร้างชั้นของแบบจำลอง อย่างไรก็ตาม ด้านต่าง ๆ เช่น ความลึกซึ้งเชิงความหมาย ความเพียงพอ มุมมองในแง่คิดของคนทั่วไป และการบอกลักษณะอย่างเป็นทางการของ BPEL4WS เช่นความสามารถการเข้าถึง ยังไม่ได้ทำการศึกษาอย่างเป็นระบบ ถึงแม้ว่า BPEL4WS จะไม่ใช่ข้อเสนอที่เลวอะไรนัก แต่ก็น่าที่งที่ได้รับความสนใจอย่างมากในปัจจุบัน ในขณะที่ประเด็นปัญหาที่เป็นพื้นฐานมากกว่าหลายประเด็น เช่น ความลึกซึ้งเชิงความหมาย ไม่ได้รับความสนใจเท่าที่ควร ดังนั้นงานวิจัยนี้จึงทำการวิเคราะห์และประเมินภาษาที่เรียกว่าเป็นมาตรฐานสำหรับการสร้างองค์ประกอบตัวบริการเว็บ ซึ่งก็พบว่าภาษา BPEL4WS มีฟังก์ชั่นการทำงานที่มากกว่าภาษาที่ใช้ในการสร้างองค์ประกอบตัวบริการเว็บอื่น ๆ และเป็นภาษาที่ดีที่สุดสำหรับใช้ในการสร้างองค์ประกอบตัวบริการเว็บในปัจจุบัน