บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 บทนำ

ในอดีตนั้น การแสดงผลบนอินเตอร์เน็ต (Internet) จะอยู่บนมาตรฐาน Hypertext Markup Language (HTML) ซึ่งทำให้ไม่สามารถประมวลผล หรือโต้ตอบกับผู้ใช้งานได้ ซึ่งเป็นข้อจำกัดที่สำคัญในการทำธุรกรรมบนอินเตอร์เน็ตหรือการค้าอิเล็กทรอนิกส์ (E-Commerce) ทำให้มีการพัฒนารูปแบบของการแสดงผลบนอินเตอร์เน็ตจาก HTML ธรรมดา ไปเป็น SHTML และ XHTML จนกระทั่งเป็นเว็บแอพพลิเคชั่น (Web Application) และมีการพัฒนาต่อมาจนเป็นรูปแบบของเว็บเซอร์วิสในปัจจุบัน

2.1.1 Web Application คืออะไร?

เว็บแอพพลิเคชั่น คือโปรแกรมที่อยู่ในเว็บเซิร์ฟเวอร์ (Web Server) ที่ทำหน้าที่ให้บริการเมื่อมีการร้องขอการใช้บริการจากทางไคลเอนท์ (Client) ผ่านทางโปรโตคอลสำหรับสื่อสารผ่านเว็บ เช่น HTTP ซึ่งการแสดงผลนั้น จะแสดงผลของผลลัพธ์ที่ได้จากการร้องขอในรูปแบบ HTML ผ่านทางเว็บบราวเซอร์ (Web Browser)

2.1.2 Web Service คืออะไร?

จากคำจำกัดความใน Web Services Glossary, W3C Working Group Note 11 February 2004 ระบุไว้ว่า

> *"A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-process able format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards."*

หรือกล่าวคือ เว็บเซอร์วิส (Web Service) คือระบบซอฟต์แวร์ที่ถูกออกแบบมาเพื่อใช้ในการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ผ่านระบบเครือข่าย โดยอธิบายรูปแบบการเชื่อมต่อได้ด้วยรูปแบบที่เครื่องคอมพิวเตอร์สามารถเข้าใจตรงกัน (ปัจจุบันใช้ WSDL) หากระบบอื่นต้องการติดต่อกับเว็บเซอร์วิสจะต้องกำหนดรายละเอียดด้วย SOAP-message ซึ่งส่งผ่าน HTTP โดยมี XML เป็นตัวกลางระหว่างมาตรฐานอื่นๆ

2.1.3 ความแตกต่างระหว่าง Web Application กับ Web Service

ในอดีต การใช้งานเว็บแอพพลิเคชั่นได้รับการยอมรับว่ามีความสามารถให้บริการได้เป็นอย่างดี และยังสามารถตอบสนองต่อแนวคิดการทำระบบการประมวลผลแบบกระจาย (Distributed Processing) ได้ในระดับหนึ่ง กล่าวคือสามารถแบ่งการประมวลผลไว้ในฝั่งของเซิร์ฟเวอร์ และใช้ฐานข้อมูล (Database) ควบคู่กันไปกับการทำเว็บแอพพลิเคชั่นแล้วผู้ใช้บริการ (Client) ร้องขอบริการไปยังผู้ให้บริการ (Server) ได้ ซึ่งเป็นไปตามความต้องการในการทำธุรกรรมในยุคใหม่ หรือการค้าอิเล็กทรอนิกส์ แต่ก็เกิดปัญหาที่ตามมาคือเรื่องของการจ่ายเงิน หรือที่เรียกว่า การจ่ายเงินอิเล็กทรอนิกส์ (E-Payment) ซึ่งเว็บแอพพลิเคชั่นที่ทำการค้าอิเล็กทรอนิกส์ต้องใช้บริการจากธนาคารออนไลน์ในการจัดเก็บเงินกับลูกค้า เพราะด้วยเทคโนโลยีนี้ การใช้บริการเก็บเงินจากธนาคารออนไลน์จำเป็นที่ผู้ค้าจะต้องไปทำการตกลงกับธนาคาร และเขียนโปรแกรมให้ตรงตามมาตรฐานที่ธนาคารผู้ให้บริการเพย์เม้นต์เกตเวย์ (Payment-Gateway) กำหนดไว้

ด้วยปัญหาดังกล่าว ทำให้เกิดความยุ่งยากในการค้นหา ติดต่อ และตกลงในการขอใช้บริการเก็บเงินจากธนาคารออนไลน์แนวคิดของเว็บเซอร์วิสจึงถูกมองว่าเป็นทางออกของปัญหานี้ ด้วยความโดดเด่นของเทคโนโลยีเว็บเซอร์วิสในการทำให้เว็บกับเว็บ สามารถติดต่อสื่อสารกันได้ผ่านทางเอกสาร XML ที่ทั้งคนและคอมพิวเตอร์ต่างก็เข้าใจตรงกัน และคอมพิวเตอร์ยังสามารถนำข้อมูลนั้นไปประมวลผลต่อไป และเอกสาร XML นี้เอง ทำให้เว็บสามารถส่งข้อมูลที่จำเป็นไปให้อีกเว็บหนึ่งทำงานบางอย่างได้ ทำให้เป็นการง่ายต่อการพัฒนาโปรแกรมติดต่อสื่อสารหรือขอใช้บริการเก็บเงินจากธนาคารออนไลน์แต่สำหรับเว็บแอพพลิเคชั่นที่ใช้การส่งข้อมูลผ่านทางเอกสาร HTML นั้น ทำให้ข้อมูลไม่สามารถนำไปประมวลผลต่อได้ การเขียนโปรแกรมจึงยุ่งยากกว่า ตามที่กล่าวไว้ข้างต้น

จะเห็นได้ว่าเว็บแอพพลิเคชั่นและเว็บเซอร์วิสนั้น ต่างก็ใช้ HTTP โปรโตคอลและอินเตอร์เน็ตเป็นช่องทางในการสื่อสารเหมือนกัน แต่มีวัตถุประสงค์ที่ต่างกัน เว็บแอพพลิเคชั่นใช้เพื่อการแลกเปลี่ยนไฟล์ HTML ระหว่างเว็บเซิร์ฟเวอร์กับไคลเอนท์ ส่วนเว็บเซอร์วิสเป็นการแลกเปลี่ยนบริการ (ซอฟแวร์คอมโพเน้นต์) ระหว่างระบบสารสนเทศ ผ่านเว็บเซิร์ฟเวอร์

ในด้านความสามารถในการใช้งาน โดยส่วนใหญ่จะใช้เว็บแอพพลิเคชั่นในการติดต่อกับผู้ใช้ ผ่านทางเว็บบราวเซอร์ เพื่อนำเสนอข้อมูลและการทำธุรกรรมต่างๆ ส่วนเว็บเซอร์วิสจะทำหน้าที่ในการติดต่อกับเว็บเซิร์ฟเวอร์เพื่อแลกเปลี่ยนข้อมูล และการทำงานหรือใช้บริการข้ามระบบกัน โดยใช้เว็บแอพพลิเคชั่นหรือแอพพลิเคชั่นอินเตอร์เฟส ในการติดต่อกับผู้ใช้ นอกจากนี้เว็บ

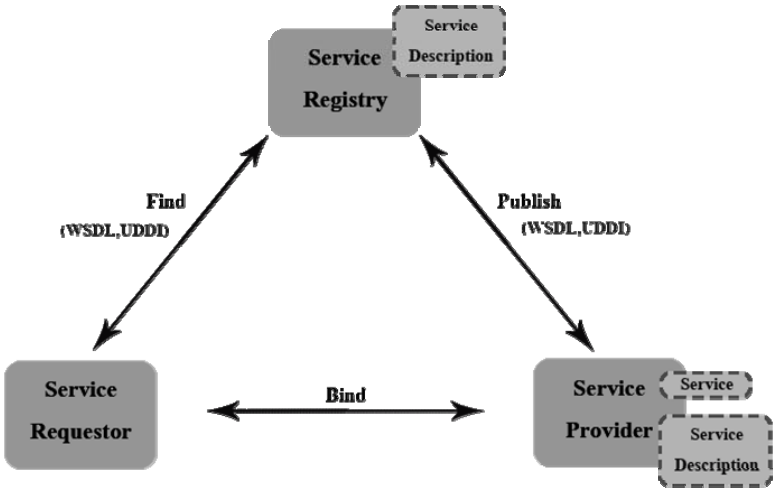
เซอร์วิสยังสามารถทำงานกับระบบต่างๆได้มากกว่า 1 ระบบ ในขณะที่เว็บแอพพลิเคชั่นไม่สามารถทำได้โดยตรง ซึ่งสามารถสรุปข้อเปรียบเทียบได้ดังตารางที่ 2.1

ตารางที่ 2.1: ข้อแตกต่างระหว่าง Web Application และ Web Service

หัวข้อ	Web Application	Web Service
การเชื่อมต่อ	มนุษย์กับโปรแกรม	โปรแกรมกับโปรแกรม
ภาษาที่ใช้	HTML	XML
รายชื่อการให้บริการ	ค้นหาผ่านเซิร์ทเอนจิ้น	ค้นหาผ่าน UDDI
ขอบเขตการใช้งาน	ธุรกิจกับลูกค้า	ธุรกิจกับธุรกิจ
โปรโตคอล	HTTP	SOAP+HTTP

และจากแนวคิดของเว็บเซอร์วิสก็คือการที่เว็บสามารถให้บริการจากการร้องขอจากต่างเซิร์ฟเวอร์ได้ ด้วยเหตุนี้เอง ทำให้เทคโนโลยีเว็บเซอร์วิสเอื้อต่อแนวคิดระบบการประมวลผลแบบกระจายมากกว่าเว็บแอพพลิเคชั่น เมื่อประกอบกับการที่เว็บเซอร์วิสมี Universal Description, Discovery and Integration (UDDI) ซึ่งเป็นมาตรฐานในการอธิบายและค้นหาเว็บเซอร์วิส ทำให้เว็บเซอร์วิสสามารถค้นหาบริการต่างๆที่ต้องการได้จากทั่วทุกมุมโลก อย่างไรก็ตาม ในการนำไปใช้งานจริงนั้น จำเป็นที่จะต้องประยุกต์ใช้งานเว็บเซอร์วิสร่วมกับเว็บแอพพลิเคชั่น โดยเว็บเซอร์วิสจะช่วยให้การเข้าถึงข้อมูลสารสนเทศจากแอพพลิเคชั่นที่ต่างกันนั้นเป็นไปได้โดยง่าย โดยการพัฒนาแอพพลิเคชั่นนั้นๆพัฒนาบนภาษาต่างๆกันและทำงานบนเครื่องคอมพิวเตอร์ในแพลทฟอร์ม (platform) ที่ต่างกัน เช่นแอพพลิเคชั่นหนึ่งอาจเขียนด้วยภาษาจาวาและทำงานอยู่บนโซลาริส แอพพลิเคชั่นเซิร์ฟเวอร์หรืออาจจะเขียนด้วย C++ และทำงานอยู่บน Microsoft Windows XP หรือแม้แต่เขียนด้วย Perl และทำงานอยู่บนเครื่อง Linux RedHat ก็เป็นไปได้ ซึ่งมาตรฐานของเว็บเซอร์วิสทำให้อินเตอร์เฟสของแอพพลิเคชั่นเหล่านี้ถูกอธิบายโดย WSDL และทำให้อยู่ในมาตรฐานของ UDDI จึงสามารถติดต่อสื่อสารถึงกันได้โดยเอกสาร XML ผ่านทางอินเตอร์เฟสของ SOAP แต่การติดต่อกับลูกค้านั้น เว็บเซอร์วิสก็ยังต้องใช้งานร่วมกับเว็บแอพพลิเคชั่นโดยส่งผ่านข้อมูลทางอินเตอร์เน็ต ซึ่งนับเป็นวิธีที่มีประสิทธิภาพในการติดต่อสื่อสารกับลูกค้าหรือหุ้นส่วน ถึงแม้ว่าจะต้องคำนึงถึงระบบรักษาความปลอดภัยและการจัดรายการของข้อมูลอยู่ก็ตาม แต่เว็บเซอร์วิสได้ใช้มาตรฐานทั่วไปของอินเตอร์เน็ต ดังนั้นเรื่องดังกล่าวจึงนับเป็นเรื่องของการติดต่อสื่อสารผ่านระบบอิเล็กทรอนิกส์ที่จำเป็นต้องมีการระมัดระวังตามปกติอยู่แล้ว และจากการพัฒนาอย่างต่อเนื่องนี้ ในอนาคตเป็นไปได้ว่าการใช้งานเว็บแอพพลิเคชั่นนั้น อาจเป็นเพียงแค่การรวมเอาบริการของแต่ละเว็บเซอร์วิสที่มีให้บริการมารวมกัน และแสดงผลเท่านั้นก็เป็นได้

2.2 Web Service Model



รูปที่ 2.1 : Web Service Model

โครงสร้างของบริการเว็บเซอร์วิสจะมีลักษณะดังรูป โดยมีรายละเอียดต่างๆดังนี้

- **เซอร์วิสรีเควสเตอร์ (Service Requestor)** : คือ ผู้ใช้บริการระบบ ที่ต้องการเรียกใช้บริการเว็บเซอร์วิสจากเซอร์วิสโพรไวเดอร์ซึ่งสามารถค้นหาบริการที่ต้องการได้จากเซอร์วิสรีจิสตรี้หรือสามารถติดต่อผ่านทางเซอร์วิสโพรไวเดอร์ก็ได้
- **เซอร์วิสรีจิสตรี้ (Service Registry) :** ทำหน้าที่เป็นตัวกลาง ให้เซอร์วิสโพรไวเดอร์มาลงทะเบียน โดยใช้ WSDLไฟล์ บอกรายละเอียดขององค์กร และบริการที่มีให้บริการ ซึ่งภายในระบบนั้น อาจจะใช้งานเซอร์วิสรีจิสตรี้หรือไม่ใช้ก็ได้
- **เซอร์วิสโพรไวเดอร์ (Service Provider) :** คือ ผู้ให้บริการระบบ ทำหน้าที่ในการเปิดบริการ เพื่อรองรับการร้องขอใช้บริการจากเซอร์วิสรีเควสเตอร์ที่เรียกเข้ามา

2.3 Web Service Technology

การทำงานของเว็บเซอร์วิสจะประกอบด้วยมาตรฐานหลัก 4 อย่างดังนี้

2.3.1 XML

XML หรือ Extensible Markup Language เป็นภาษามาตรฐานที่ทุกระบบสนับสนุน ทำให้ข้อมูลที่มีโครงสร้างของภาษา XML จะถูกนำไปประมวลผลต่ออย่างอัตโนมัติได้อย่างง่ายดาย ภาษา XML จึงถูกนำมาใช้เป็นภาษากลางในการแลกเปลี่ยนข้อมูลของเทคโนโลยีเว็บเซอร์วิส

2.3.1.1 XML คืออะไร

XML เป็นส่วนหนึ่งของ Standard Generalized Markup Language (SGML) ซึ่งเป็นข้อกำหนดในการสร้างหรือจัดทำเอกสารในรูปแบบอิเล็กทรอนิกส์ ที่ทำให้ได้ข้อมูลที่เป็นเมตะดาต้า (Metadata) หรือก็คือข้อมูลที่สามารถนำไปแปลงเพื่อใช้งานในแอพพลิเคชั่นที่ต่างกันได้ โดย W3C (World Wide Web Consortium) ได้ให้นิยามของ XML ว่า XML คือ รูปแบบข้อมูลที่มีโครงสร้างที่เอื้อต่อการแลกเปลี่ยนข้อมูลในเครือข่ายอินเตอร์เน็ต โดย XML มีลักษณะเป็น ไฟล์ข้อความที่ต้องศึกษาวิธีการอ่านข้อมูลมาใช้ก่อน เพราะมีไวยากรณ์บังคับเพื่อแสดงความสัมพันธ์ของข้อมูลที่มีโครงสร้างแบบเมตะดาต้ากำกับอยู่

2.3.1.2 ความแตกต่างระหว่าง XML กับ HTML

ทั้ง XML และ HTML เป็นภาษาแบบมาร์กอัฟ (Markup Language) คือข้อมูลทั้งใน XML และ HTML เป็น Tag กำกับหัวและท้ายของข้อมูลเหมือนกัน แต่ในด้านของจุดประสงค์การทำงานนั้น XML ออกแบบมาเพื่อใช้ในการส่งข้อมูล บอกว่าข้อมูลที่ต้องการส่งคืออะไร แต่ HTML ออกแบบมาเพื่อใช้ในการแสดงข้อมูล บอกว่าข้อมูลต้องแสดงผลอย่างไร อีกสิ่งหนึ่งที่ต่างกันใน XML และ HTML ก็คือแท็กของ XML นั้นผู้ใช้ต้องทำการกำหนดเอง แต่ HTML นั้นมีรูปแบบของแท็กในการแสดงผลอยู่แล้ว ผู้ใช้ไม่สามารถสร้างรูปแบบของแท็กขึ้นเองได้

2.3.1.3 ความสามารถของ XML

เมื่อนำXMLไปใช้ควบคู่กับแอพพลิเคชั่นอื่นๆ ทำให้สามารถแยกส่วนที่เป็นแอพพลิเคชั่นกับข้อมูลออกมาเก็บไว้คนละไฟล์ได้ เพื่อช่วยในการทำงานจะได้สามารถสนใจ ณ จุดๆเดียวเท่านั้น เช่น การใช้ XML ควบคู่กับ HTML เวลาใช้งาน ในส่วนของ HTML ก็จะได้สนใจเฉพาะส่วนที่เป็นเลเอ้าท์ของข้อมูล และการแสดงผลเท่านั้น โดยสามารถแน่ใจได้ว่าการเปลี่ยนแปลงต่างๆใน HTML จะไม่กระทบกับส่วนที่เป็นข้อมูล เพราะส่วนนั้นอยู่ในส่วนของ XML ซึ่งเป็นไฟล์ที่เก็บแยกจากกันเป็นต้น

ในการใช้งานจริงนั้น ระบบคอมพิวเตอร์และฐานข้อมูลที่เก็บข้อมูล แต่ละระบบนั้นมักมีรูปแบบของข้อมูลที่ไม่เหมือนกัน ทำให้เกิดความยุ่งยากในการแลกเปลี่ยนข้อมูลข้ามระบบ การใช้ XML จึงเป็นอีกทางออกหนึ่งที่ใช้เป็นสื่อกลางระหว่างแต่ละระบบ เพราะ XML สามารถทำให้ข้อมูลที่เก็บอยู่จากแอพพลิเคชั่นหนึ่งสามารถนำไปใช้กับแอพพลิเคชั่นอื่นๆที่หลากหลายได้อีกด้วย เช่น ต้องการเอาผลที่ได้จากแอพพลิเคชั่นหนึ่งไปใช้ยังอีกแอพพลิเคชั่นหนึ่งโดยที่ทั้งคู่มีรูปแบบชองไฟล์ที่แตกต่างกัน เป็นต้น หรือสามารถใช้ XML มาทำการแชร์ข้อมูลระหว่าง ระบบหรือแอพพลิเคชั่นได้อีกด้วย

2.3.1.4 การใช้งานของ XML

XML บอกรายละเอียดของข้อมูลโดยการใช้ Document Type Definite (DTD) หรือ XML Schema ก็ได้ ซึ่งเป็นการออกแบบข้อมูลแบบ Self-Descriptive หรือก็คือแท็กของ XML นั้นไม่ได้มีการกำหนดตายตัว ซึ่งผู้ใช้ต้องทำการกำหนดค่าเอง ซึ่งอาจทำให้เนื้อหามีชนิดของข้อมูลที่ต่างกันไป ดังนั้นจึงควรใช้ XML Schema มากกว่า DTD เพราะ DTD ไม่ได้ถูกออกแบบมาเพื่อตรวจสอบชนิดของข้อมูลหรือการตรวจสอบขอบเขตค่าของข้อมูล ในทางกลับกัน XML Schema ถูกเขียนขึ้นใน XML ซึ่งทำหน้าที่ในการสร้างข้อมูลที่ DTD นำเสนอ, กำหนดชนิดของข้อมูลและกำหนดช่วงของค่าแอดทริบิวต์ (Attribute) และอิลิเม้นต์ (Element) ได้

การเขียน XML file มีรูปแบบการเขียนที่บังคับตายตัว แต่ง่ายต่อการเขียนและการทำความเข้าใจ โดยส่วนแรกต้องเริ่มจากการประกาศว่า XML ที่ใช้เป็นเวอร์ชั่นใด ใช้การเข้ารหัสตัวอักษรแบบใด เช่น <?xml version= “1.0” encoding= “ISO-8859-1” ?> เป็นต้น ส่วนต่อมาจะเป็นส่วนที่บอกอิลิเม้นต์และแอดทริบิวต์ของข้อมูลที่ต้องการส่งมีโดยรูปแบบที่กำหนดจะเป็น<element> attribute </element> เช่น <date> 12/12/05 </date> หมายความว่า อิลิเม้นต์ คือ Date ที่มีแอดทริบิวต์หรือมีค่าเป็น 12/12/05 เป็นต้น โดยElement ของ XML นั้นคำนึงถึงตัวอักษรเล็กใหญ่ (Case Sensitive) ด้วย

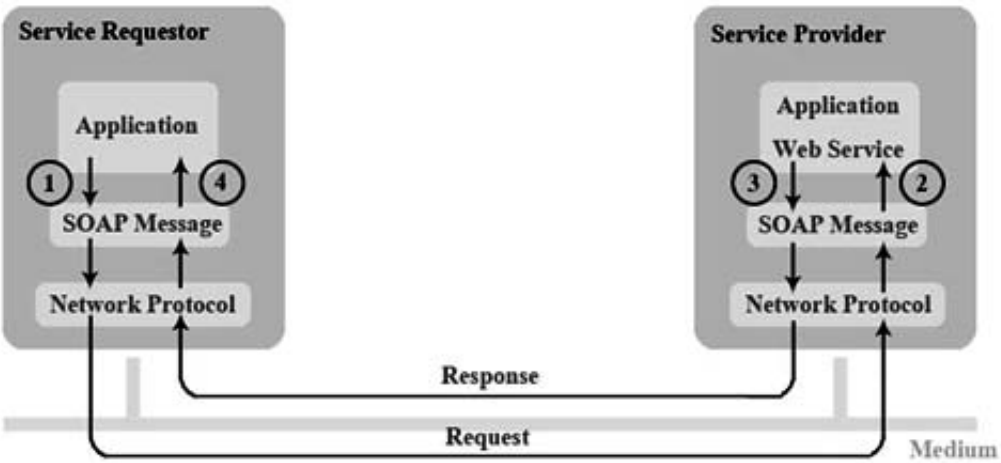
2.3.2 SOAP

SOAP หรือ Simple Object Access โปรโตคอล เป็นมาตรฐานของเทคโนโลยีวัตถุซอฟต์แวร์แบบกระจาย (Distributed Object) แบบหนึ่ง โดยทำหน้าที่ส่งข้อมูลผ่านเครือข่ายอินเตอร์เน็ตในรูปแบบของ XML ทำให้เรียกใช้งานโปรแกรมผ่านทางเครือข่ายอินเตอร์เน็ตได้

SOAP กลายเป็นสิ่งที่มีความสำคัญสำหรับเว็บเซอร์วิสอย่างรวดเร็ว และกลายเป็นโปรโตคอลที่ผู้จัดหาเว็บเซอร์วิสเลือกที่จะใช้ส่งข้อความระหว่างเว็บเซอร์วิสทั้งนี้เนื่องจาก SOAP เป็นโปรโตคอลในการส่งข้อมูล (Transport Protocol) ที่มี XML เป็นพื้นฐานและใช้ HTTP เป็นโปรโตคอลร่วมในการส่งผ่านเครือข่าย โดยจะทำหน้าที่ระบุวิธีในการเข้ารหัสส่วนหัว (Head Encoding) ของทั้งไฟล์ที่เป็น HTTP และ XML ไว้อย่างชัดเจน ทั้งในส่วนของการติดต่อไปยังคอมพิวเตอร์อีกเครื่องหนึ่งและส่งผ่านข้อมูลไปให้ รวมไปถึงระบุวิธีที่โปรแกรมซึ่งถูกเรียกนั้นจะส่งค่าคืนกลับมาด้วย

SOAP มีการแลกเปลี่ยนข้อมูลในสภาวะแวดล้อมแบบกระจายศูนย์ (Decentralized, Distributed environment) โดยได้มีการกำหนดโปรโตคอลที่ใช้ในการส่งข้อความ (Messaging

Protocol) ระหว่างผู้ขอบริการ และผู้ให้บริการ เช่นผู้ขอบริการสามารถติดต่อแลกเปลี่ยนข้อมูลกับผู้ให้บริการโดยใช้ Remote Method Invocation (RMI) ตามวิธีการของโปรแกรมเชิงวัตถุ โดยมีบริษัท Microsoft, IBM, Lotus, UserLand และ DeveloperMenter ได้ร่วมกันกำหนดมาตรฐานของ SOAP และต่อมาได้มีบริษัทชั้นนำอีกกว่า 30 บริษัทเข้าร่วมและจัดตั้งเป็น W3C XML Protocol Workgroup ขึ้น และ SOAP ก็ได้ถูกกำหนดให้เป็นโปรโตคอลสำหรับสถาปัตยกรรมการให้บริการเชิงวัตถุ (Services-Oriented Architecture Protocol) เรียบร้อยแล้ว โดยที่มีจุดเด่นก็คือเป็นโปรโตคอลที่เป็นกลาง กล่าวคือไม่มีใครเป็นเจ้าของ และเป็นโปรโตคอลที่ทำงานกับโปรโตคอลอื่นหลายชนิด การพัฒนาก็สามารถทำได้อย่างอิสระตามแพลทฟอร์มของระบบปฏิบัติการ,โมเดลเชิงวัตถุ (Object model) และภาษาโปรแกรมของผู้ที่ทำการพัฒนา



รูปที่ 2.2 การใช้ SOAP ในการส่ง XML Message

จากรูปที่ 2.2 เป็นขั้นตอนในการส่ง SOAP Message ระหว่าง เซอร์วิสรีเควสเตอร์กับเซอร์วิสโพรไวเดอร์ดังนี้

หมายเลข 1 : แอพพลิเคชั่นของผู้ร้องขอบริการสร้าง SOAP Message เพื่อเรียกใช้บริการของเว็บเซอร์วิสผ่านเน็ตเวิร์คโปรโตคอล

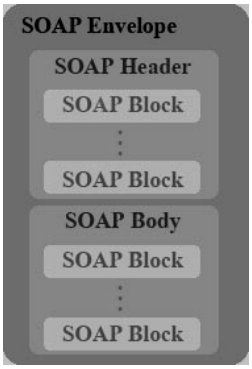
หมายเลข 2 : เว็บเซอร์วิสของผู้ให้บริการได้รับ SOAP Message จากเซอร์วิสรีเควสเตอร์ในรูปแบบของ XML

หมายเลข 3 : เว็บเซอร์วิสประมวลผลตามองค์ประกอบที่ให้บริการเว็บเซอร์วิสส่งผลลัพธ์มาแล้วผู้ให้บริการก็จะสร้าง SOAP Message ที่มีผลลัพธ์นั้นส่งกลับมายังผู้ร้องขอบริการ

หมายเลข 4 : แอพพลิเคชั่นของผู้ร้องขอบริการได้รับผลลัพธ์เป็น SOAP Message แล้วแปลงให้อยู่ในรูปแบบที่ต้องการ เพื่อนำไปประมวลผลต่อ

เอกสาร SOAP นั้นมีโครงสร้างในรูปแบบ XML ซึ่งเราสามารถแบ่งเป็นส่วนของเอกสารได้เป็น 3 ส่วนหลักดังนี้คือ

1. SOAP Envelop คือเนื้อหา (content) ของเอกสารทั้งหมด
2. SOAP Header คือส่วนเพิ่มของเอกสาร SOAP ซึ่งจะมีหรือไม่มีก็ได้
3. SOAP Body คือส่วนที่ใช้ในการเรียกใช้งานเซอร์วิสและผลลัพธ์ที่ได้จากเว็บเซอร์วิส



รูปที่ 2.3 โครงสร้างของเอกสาร SOAP

2.3.3 WSDL

WSDL หรือ Web Service Description Language เป็นภาษามาตรฐานที่ใช้สำหรับอธิบายการใช้งานระบบที่เปิดให้บริการ ซึ่งเขียนขึ้นตามมาตรฐาน XML ดังนั้น WSDL จึงเป็นเสมือนคู่มือให้กับระบบในการเรียนรู้วิธีการเรียกใช้งานเว็บเซอร์วิสซึ่งเกี่ยวข้องกับความต้องการของ ระบบกระจายศูนย์ (Distributed System)

WSDL เกิดจากการรวมแนวคิดของ NASSL (The Network Accessible Service Specification Language), WDS (Well-Defined Services) ของบริษัท IBM, SDL (The Service Description Language) และ SCL (The SOAP Contract Language) ของบริษัท Microsoft ปัจจุบัน WSDL เป็นภาษาที่อยู่ภายใต้การดูแลของ W3C (World Wide Web Consortium) ซึ่งยังไม่เป็นมาตรฐานที่สมบูรณ์ เวอร์ชั่นที่ใช้งานอยู่ในปัจจุบันคือ WSDL 1.1 อาจกล่าวได้ว่า WSDL คือมาตรฐานสำหรับการประกาศโพรเซสที่จำเป็นในการเรียกใช้บริการจาก SOAP

ในการใช้งาน WSDL จริง หากเราสร้างบริการเว็บเซอร์วิสก็จะมีเครื่องมือช่วยสร้างเอกสาร WSDL สำหรับเว็บเซอร์วิสอย่างอัตโนมัติ อิลิเม้นต์ภายในเอกสารที่เราควรรู้เกี่ยวกับการติดต่อและเรียกใช้บริการของเว็บเซอร์วิสมีจุดที่ควรรู้ดังตารางที่ 2.2

ตารางที่ 2.2 : ความหมายของ Element ที่น่าสนใจใน WSDL Document

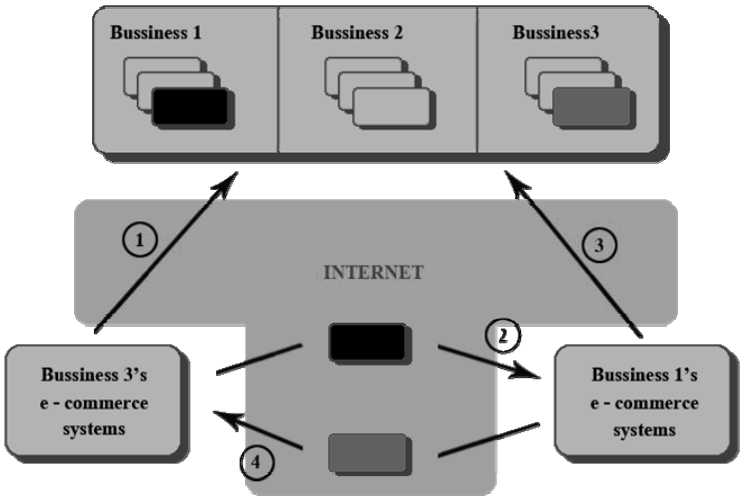
Element	Definition
<portType>	เป็นส่วนที่สำคัญที่สุดใน WSDL อิลิเม้นต์ อธิบาย การจัดการต่างๆที่เว็บเซอร์วิสมีให้บริการและ แมสเซจที่เกี่ยวข้อง เทียบได้กับฟังก์ชั่นไลบรารี่ หรือโมดูล หรือคลาส ในการเขียนโปรแกรม
<operation>	อธิบาย ที่ให้บริการเว็บเซอร์วิสโดยจะมีเมทธอด จำนวนกี่เมทธอดก็ได้
<message>	อธิบายดาต้าอิลิเม้นต์ของการจัดการโดยในแต่ละแมสเซจอาจจะมีมากกว่าหนึ่งส่วน เทียบได้กับพารามิเตอร์ของฟังก์ชั่นในการเขียนโปรแกรม
<types>	อธิบายชนิดข้อมูลที่เว็บเซอร์วิสใช้ เพื่อความเป็นกลาง WSDL จะใช้ซินแท็กของ XML Schema ในการระบุชนิดข้อมูล
<binding>	อธิบายรูปแบบของแมสเซจ และรายละเอียดของโปรโตคอล ในแต่ละพอร์ต
<service>	สำหรับเว็บเซิร์ฟเวอร์จะมีบริการเว็บเซอร์วิสกี่บริการก็ได้ และชื่อของเว็บเซอร์วิสจะใช้เป็นตัวจำแนกและบ่งบอกแต่ละบริการ ซึ่งห้ามใช้ชื่อซ้ำกัน

ตามทฤษฎีแล้วไฟล์เอกสารของ WSDL แต่ละไฟล์สามารถอธิบายคุณลักษณะของบริการเว็บเซอร์วิสได้มากกว่า 1 บริการ โดยแต่ละเว็บเซอร์วิสจะมีพอร์ตสำหรับสื่อสารเฉพาะตัว ซึ่งได้บ่งบอกไว้ในเอกสารของ WSDLอยู่แล้ว

2.3.4 UDDI

UDDI ย่อมาจาก Universal Description, Discovery, and Integration เป็นระบบมาตรฐานในการอธิบายและค้นหาเว็บเซอร์วิสโดยเป็นตัวกลางให้เซอร์วิสโพรไวเดอร์มาลงทะเบียนโดยใช้ไฟล์ WSDL บอกรายละเอียดของบริษัทและบริการที่มีให้ ทำให้เซอร์วิสรีเควสเตอร์สามารถค้นหาและทราบได้ว่าบริษัทมีผลิตภัณฑ์และบริการใดบ้าง สามารถติดต่อขอดำเนินธุรกิจกับบริษัทได้โดยอัตโนมัติผ่านทางเว็บเซอร์วิส

UDDI เป็นมาตรฐานที่ให้ชุดพื้นฐาน APIs (Application Programming Interface) ของ SOAP ที่สามารถนำมาใช้ในการพัฒนาเป็นตัวแทนของผู้ให้บริการ (Service broker) UDDI ใช้สำหรับค้นหาบริการที่ต้องการและเมื่อได้มาแล้ว UDDI ยังจัดหาข้อตกลงในวิธีการที่จะใช้งานเปรียบได้กับสมุดหน้าเหลือง เป็นมาตรฐานที่จัดตั้งขึ้นโดยบริษัทไอบีเอ็ม บริษัทไมโครซอฟต์ และบริษัทอารีบา (Ariba) ปัจจุบันมีบริษัทที่ร่วม กันกำหนดมาตรฐานของ UDDI มากกว่า 70 บริษัท ซึ่งมาตรฐานของ UDDI ถูกกำหนดให้เป็นมาตรฐานสำหรับ B2B interoperability



รูปที่ 2.4 การใช้งาน SOAP, WSDL, UDDI ร่วมกันใน Web Service

จากรูปที่ 2.4 เป็นตัวอย่างการทำงานของเว็บเซอร์วิสที่ใช้ SOAP, WSDL, UDDI ร่วมกัน ของพนักงานไอที ของ Business 3 ดังนี้

หมายเลข 1: ค้นหาข้อกำหนดการให้บริการใน WSDL ผ่านอินเตอร์เฟสของ Business 1

หมายเลข 2: ติดต่อผ่านอินเตอร์เฟสของแอพพลิเคชั่นที่ให้บริการ (ด้วย SOAP) ผ่านทางอินเตอร์เน็ต

หมายเลข 3,4: Business 1 รวมหรือบูรณาการกับ Business 3 ด้วยวิธีเดียวกัน

จากมาตรฐานทั้ง 4 อย่างข้างต้น สามารถสรุปขั้นตอนการทำงานของเว็บเซอร์วิสได้ดังนี้

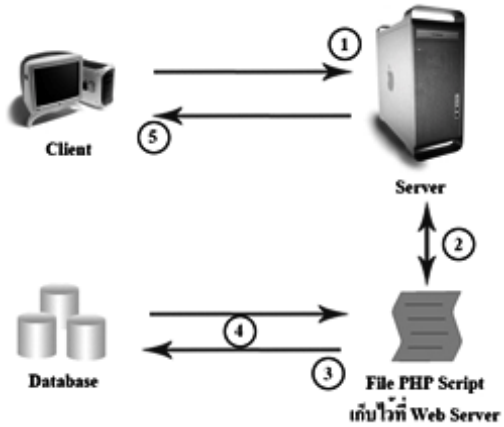
- เซอร์วิสโพรไวเดอร์จัดทำระบบหรือบริการเว็บเซอร์วิสขึ้นมา
- เซอร์วิสโพรไวเดอร์ทำการลงทะเบียนบริการเว็บเซอร์วิสกับหน่วยงานที่ให้บริการระบบ UDDI หรือเซอร์วิสรีจิสตรี้
- นำไฟล์ WSDL ไปไว้ในระบบ UDDI ที่ได้ลงทะเบียนไว้
- เซอร์วิสรีเควสเตอร์ทำการค้นหาระบบหรือบริการที่ต้องการจากระบบ UDDI
- เมื่อเซอร์วิสรีเควสเตอร์พบระบบหรือบริการที่ต้องการ ก็จะนำไฟล์ WSDL ไปเรียนรู้วิธีการเรียกใช้ระบบนั้นๆผ่านระบบของตน

เซอร์วิสรีเควสเตอร์ทำการติดต่อและเรียกใช้ระบบหรือบริการจากเซอร์วิสโพรไวเดอร์ได้โดยตรงผ่าน SOAP ในระบบของตน

2.4 PHP

PHP ได้ถูกคิดค้นขึ้นในปีค.ศ.1994 โดย Rasmus Lerdorf ซึ่งเริ่มจากการที่เขาเขียนเพิร์ลสคริปต์ เพื่อใช้เป็น CGI สำหรับการบันทึกข้อมูลผู้ที่เข้าเยี่ยมชมโฮมเพจส่วนตัวของเขา แต่เนื่องจาก Lerdorf เห็นว่าการเขียน CGI ด้วยเพิร์ลนั้นออกจะยุ่งยากเกินไป จึงได้ตัดสินใจเขียนโปรแกรมขึ้นใหม่ด้วยภาษา C ที่สามารถแยกส่วนที่เป็นภาษา HTML ออกจากส่วนที่เป็นภาษา C เพื่อแยกประมวลผลแล้วทำการสร้างโค้ด HTML ขึ้นใหม่ได้ โดยตั้งชื่อโปรแกรมนี้ว่า Personal Homepage Tools (PHP-Tools) และได้เริ่มแจกจ่ายโค้ด ออกไปในลักษณะฟรีแวร์ และต่อมาจึงได้เริ่มเปิดให้ผู้สนใจเข้าร่วมปรับปรุงและพัฒนาจนได้เป็น PHP/FI ที่เริ่มเป็นที่นิยมมากยิ่งขึ้นจนกระทั่ง Zeev Suraski และ Andi Gutmans ได้ร่วมกันเขียนโค้ดขึ้นใหม่โดยได้มีการปรับปรุงให้ดีขึ้นเป็นอย่างมากในหลายๆด้าน ทั้งด้านประสิทธิภาพ การสนับสนุนการโปรแกรมเชิงวัตถุ และในด้านอื่นๆอีกหลากหลายประการ จนเกิดเป็น PHP3 ซึ่งเป็นเวอร์ชั่นที่ได้รับความนิยมอย่างมาก แต่เมื่อมีผู้ใช้เป็นจำนวนมาก จึงมีการนำ PHP ไปใช้ในงานที่ซับซ้อนยิ่งขึ้น ด้วยเหตุนี้ Zeev Suraski และ Andi Gutmans ผู้พัฒนา PHP3 จึงได้ตัดสินใจเขียนโค้ดขึ้นใหม่ทั้งหมด และได้ตั้งชื่อว่า เซนเอนจิ้น (Zend Engine) ซึ่งเป็นหัวใจของ PHP4 ส่วน PHP 5 นั้นเป็นเวอร์ชั่นที่จัดได้ว่าเป็นการพลิกโฉมโปรแกรมเชิงวัตถุด้วย PHP เนื่องจากมีการเปลี่ยนแปลงไปสู่การโปรแกรมเชิงวัตถุที่สมบูรณ์แบบมากยิ่งขึ้น โดยนับตั้งแต่ปี 1994 ซึ่ง Rasmus Lerdorf ได้ให้กำหนด PHP จนถึงปัจจุบันนี้ มีการสำรวจพบว่า PHPโมดูลได้ถูกในไปใช้แล้วกว่า 16 ล้านโดเมนและยังมีแนวโน้มที่สดใส เนื่องจากสามารถใช้งานได้กับเว็บเซิร์ฟเวอร์ที่หลากหลาย ในหลากหลายแพลตฟอร์ม

PHP เป็นภาษาสคริปต์แบบเซิร์ฟเวอร์ไซด์สคริปต์ (Server-Side Script) ซึ่งหมายถึงการประมวลผลจะเกิดขึ้นบนเครื่องเซิร์ฟเวอร์ แล้วจึงสร้างผลลัพธ์เป็นภาษา HTML ส่งให้กับเครื่องไคลเอนท์เพื่อแสดงผล ซึ่งลดภาระการส่งถ่ายข้อมูลจำนวนมากเพื่อมาประมวลผลบนเครื่องไคลเอนท์ การเขียนสามารถทำได้โดยเขียนโค้ด PHP แทรกลงไปในโค้ด HTML หรือเขียนเป็นโค้ด PHP อย่างเดียวก็ได้ และทำการบันทึกไฟล์นั้นให้เป็นไฟล์ที่มีนามสกุล .php,.php3,.phtml หรือตามที่กำหนดไว้ในการติดตั้งเว็บเซิร์ฟเวอร์



รูปที่ 2.5 ขั้นตอนการทำงานของ PHP Script

จากรูปที่ 2.5 จะเห็นการทำงานเป็นขั้นตอนต่างๆดังนี้

หมายเลข 1: ฝั่งไคลเอนท์ทำการร้องขอหรือเรียกใช้งานไฟล์PHP ที่อยู่ในเครื่องเซิร์ฟเวอร์

หมายเลข 2: ฝั่งเซิร์ฟเวอร์จะทำการค้นหาไฟล์ PHP แล้วทำการประมวลผลไฟล์ PHP ตามที่ไคลเอนท์ได้ทำการร้องขอมา

หมายเลข 3: ทำการประมวลผลไฟล์ PHP

หมายเลข 4: ติดต่อกับฐานข้อมูล และนำข้อมูลในฐานข้อมูลมาใช้ร่วมกับการประมวลผล

หมายเลข 5: ส่งผลลัพธ์จากการประมวลผลไปให้เครื่องไคลเอนท์

สำหรับการใช้งาน PHP เพื่อรองรับการบริการเว็บเซอร์วิสสามารถเลือกใช้ได้ 3 รูปแบบ คือ

2.4.1 PHP-SOAP

SOAP ตัวแรกที่มีการใช้งานร่วมกับ PHP ปัจจุบันได้มีการเขียน PHP-SOAP ใหม่ด้วยภาษา C และมีติดมากับแพ็คเกจของ PHP 5 เป็นโมดูลเสริม (.dll, .so) การทำงานมีลักษณะเป็นคอมไพเลอร์ทำให้สามารถประมวลผลได้อย่างรวดเร็วรายละเอียดของ PHP-SOAP สามารถค้นหาได้ที่ http://phpsoaptoolkit.sourceforge.net/phpsoap/

2.4.2 PEAR

SOAP ที่มีผู้ใช้งานมากที่สุดในปัจจุบัน ในเวอร์ชั่นล่าสุดนั้นยังไม่รองรับ PHP 5 ข้อเสียคือไม่รองรับการยืนยันตัวตนแบบไดเจส (Digest Authentication) การทำงานมีลักษณะเป็นอินเตอร์พรีเตอร์ (Interpreter)รายละเอียดของ PEAR สามารถค้นหาได้ที่ http://pear.php.net/

2.4.3 NuSOAP

SOAP ตัวล่าสุดที่มีการใช้งานร่วมกับ PHP ในการพัฒนาครั้งแรกใช้ชื่อว่า SOAPx4 โดยมีจุดเด่นที่สามารถสร้างเอกสาร WSDL ได้อย่างอัตโนมัติ , สร้างคลาสตัวแทน (Proxy Class) ได้อย่างอัตโนมัติ และรองรับการใช้งานของ HTTP Proxy , SSL รายละเอียดของ NuSOAP สามารถค้นหาได้ที่ http://dietrich.ganx4.com/nusoap/index.php

2.5 CSS (Cascading Style Sheets)

CSS ถูกสร้างขึ้นเพื่อแก้ปัญหาการแสดงผลที่ต่างกันในเว็บบราวเซอร์แต่ละตัว เช่น อินเตอร์เน็ตเอ็กซ์โพลเรอร์ (Internet Explorer) และเน็ตสเคปเนวิเกเตอร์ (Netscape Navigator) จะแสดงผลจากสไตล์ที่ต่างกันทั้งๆที่เป็นสไตล์ตัวเดียวกัน ตัว CSS นั้นเป็นสิ่งที่สร้างขึ้นโดย World Wide Web Consortium (W3C) ที่เปรียบเสมือนเป็นมาตรฐานกลาง ในการแสดงผลจาก สไตล์ซึ่งทั้ง อินเตอร์เน็ตเอ็กซ์โพลเรอร์และเน็ตสเคปเนวิเกเตอร์ ตั้งแต่เวอร์ชั่น 4.0 ขึ้นไปหรือ HTML 4.0 จะมีการรับรองการใช้งานของ CSS ทั้งหมด

2.5.1 CSS คืออะไร

ปกติแล้ว สไตล์ คือส่วนที่คอยกำกับว่าอิลิเมนต์ต่างๆของ HTML นั้นจะต้องแสดงผลออกมาอย่างไร ซึ่งปกติแล้วสไตล์นั้นจะถูกเก็บอยู่ในสไตล์ชีท (Style Sheets) ต่อมาใน HTML เวอร์ชั่น 4.0 ได้มีการนำเอาสไตล์ชีทมาเก็บแยกคนละส่วนกับเอกสาร HTML เรียกว่า สไตล์ชีทภายนอก (External Style Sheets) เพื่อใช้สำหรับการทำงานที่มีเอกสาร HTMLหลายๆไฟล์ แล้วต้องการให้มีสไตล์ที่เหมือนกัน เช่น ไม่ต้องมาทำการใส่แท็กยาวๆในแต่ละเซลล์ของทุกๆตาราง เป็นต้น ซึ่งนอกจากนี้ประโยชน์อีกข้อหนึ่งของ การทำสไตล์ชีทภายนอก คือ เพื่อให้ง่ายต่อการแก้ไขสไตล์ต่างๆที่ไม่ต้องมาทำการแก้ไขทีละเอกสารHTMLแต่จะไปแก้ไขที่สไตล์ชีทภายนอกเพียงที่เดียว ซึ่งสไตล์ชีทภายนอกจะถูกเก็บไว้ในส่วนที่เรียกว่า CSS นั่นเอง

2.5.2 การใช้งาน CSS

การแทรก CSS เพื่อใช้งานมี 3 แบบด้วยกันคือ

2.5.2.1 แบบ Internal Style Sheets

ควรใช้ในกรณีที่มีเพียงไฟล์เดียวที่ใช้สไตล์นี้ หรือ มีเอกสารเดียวที่เป็นสไตล์จำเพาะนั้นเอง โดยในส่วนของไฟล์ HTML นั้นในส่วนของแท็กที่เป็น <html> ส่วนที่เป็น <head> จะมีแทรกส่วน

ที่เป็น CSS ด้วย <style type= “text/css”> ตามด้วยส่วนที่เป็นตัวกำหนดสไตล์จากนั้น ปิดแท็กด้วย </style> แล้วทำการสร้างแท็กที่เป็นของ HTML เพื่อแสดงต่อตามปกติ

2.5.2.2 แบบ External Style Sheets

เหมาะกับกรณีที่มีเพียง 1 สไตล์ แล้วต้องการใช้กับ HTML หลายๆไฟล์ โดยแต่ละหน้าจะมีการแทรกแท็กในส่วนของ <head> ในHTML ที่ต้องการด้วย<link rel= “stylesheet” type= “text/css” href= “filename.css”/> แล้วบราวเซอร์ก็จะไปอ่านสไตล์มาจาก Filename.css ซึ่งในไฟล์นี้จะเก็บสวนที่เป็นตัวกำหนดสไตล์ที่ต้องการให้ HTML ที่อ้างอิงนั้นๆใช้ ซึ่งสามารถเขียนด้วยโปรแกรมชนิดอีดิทเตอร์ใดๆก็ได้ เช่น Notepad, Edit Plus เป็นต้น

2.5.2.3 แบบ Multiple Style Sheets

เหมาะสำหรับใช้ในกรณีที่ต้องการให้มีการเลือกส่วนที่จะเปลี่ยนในแต่ละไฟล์เหมือนกันในหลายๆหน้าแต่สไตล์ ในการเปลี่ยนต่างกัน เช่น ในไฟล์หนึ่งให้ h3 มีสีที่เป็นสีขาวอยู่ตรงกลางหน้า แต่อีกไฟล์หนึ่งให้ h3 มีสีฟ้าและอยู่ตรงกลางหน้าเช่นกัน เป็นต้น โดยวิธีการทำงานคือ มีการใช้ทั้งส่วนที่เป็นสไตล์ชีทภายนอกที่กำหนดสไตล์พื้นฐานให้แต่ละไฟล์เหมือนกัน และส่วนที่เป็นสไตล์ชีทภายในที่จะกำหนดส่วนที่ต้องการให้ต่างกัน โดยบราวเซอร์จะไปอ่านส่วนที่เป็นภายนอกก่อน และเมื่อมาเจอส่วนที่เป็นภายในที่เหมือนกันจะแสดงส่วนที่เป็นภายในแทน

ส่วนที่เป็นตัวกำหนดสไตล์นั้น ประกอบด้วย 3 อย่างสำหรับการกำหนดสไตล์แต่ละส่วนจะมีรูปแบบคือ **Style: Selector {property: value}**

2.5.2.3.1 ส่วนของ Selector คือ ส่วนที่เป็นอิลิเม้นต์ปกติของ HTML หรือ แท็กของ HTML ที่ต้องการการกำหนดรูปแบบ เช่น ต้องการกำหนดสีของ h1 ทั้งหมด Selector ก็จะเป็น h1 เป็นต้น

2.5.2.3.2 ส่วนของ Property คือ แอททริบิวต์ที่ต้องการเปลี่ยนแปลงค่า เช่น สีที่ต้องการกำหนดให้ Selector นั้นๆ Property ก็จะเป็น color เป็นต้น

2.5.2.3.3 ส่วนของ Value คือ ค่าของแอททริบิวต์ที่เลือกไว้กับ Property เช่น ต้องการสีดำ จะมีค่าเป็น #000000 เป็นต้น

2.6 JavaScript

จาวาสคริปต์ถูกสร้างขึ้นเพื่อเพิ่มความสามารถให้แก่ HTML ให้สามารถทำการโต้ตอบกับผู้ใช้ได้หรือเรียกว่า อินเตอร์แอคทีฟ (Interactive) นั่นเองจาวาสคริปต์เป็น ภาษาสคริปต์หรือ ภาษา

ในการโปรแกรมมิ่งที่มีขนาดเล็ก ซึ่งแต่ละบรรทัดของจาวาสคริปต์นั้นจะประกอบไปด้วยโค้ดที่ใช้ในการเอ๊กซ์ซีคิวต์ (Execute) ซึ่งสามารถเรียกได้ว่าเป็นภาษาที่แปลด้วยการอินเตอร์พรีต คือสคริปต์สามารถทำเอ๊กซ์ซีคิวต์ได้เลยโดยไม่ต้องทำการคอมไพล์ซึ่งปกติแล้วจะเป็นถูกเก็บรวมไว้ (Embedded) ร่วมกับ HTML และที่สำคัญคือจาวาสคริปต์ไม่มีลิขสิทธิ์ใครก็สามารถใช้ได้

2.6.1 ความสามารถของ JavaScript

จาวาสคริปต์เป็นเสมือนเครื่องมือในการโปรแกรมมิ่งให้กับผู้พัฒนาเว็บเพจเพราะเป็นภาษาที่มีรูปแบบ (Syntax) ที่ง่าย และฝังอยู่ใน HTML ได้เลย, สามารถทำการสร้าง Dynamic Text หรือข้อความที่สามารถเปลี่ยนแปลงตลอดเวลาได้ให้กับเว็บเพจ สามารถทำการโต้ตอบเมื่อเกิดเหตุการณ์ขึ้นกับเว็บเพจได้ทันทีเหมือนกับเว็บเพจสามารถตอบสนองการทำงานได้ทันทีที่ทำการ คลิ๊กบน HTML อิลิเม้นต์ได้เลย, สามารถทำการอ่าน และเขียน หรือเปลี่ยนแปลง ตัว HTML อิลิเม้นต์ได้, สามารถตรวจสอบข้อมูลก่อนส่งไปยังเซิร์ฟเวอร์ซึ่งช่วยให้การทำงานของเซิร์ฟเวอร์ลดลงและประหยัดเวลามากขึ้นด้วย, สามารถตรวจสอบบราวเซอร์ ที่ใช้อยู่เป็นบราวเซอร์ของค่ายไหนเพื่อจะได้ทำการโหลดรูปแบบการแสดงผลได้ตรงตามบราวเซอร์นั้นๆ นอกจากนี้จาวาสคริปต์ยังสามารถใช้สร้างคุกกี้ หรือข้อมูลส่วนเล็กๆที่เก็บอยู่ใน เครื่องของผู้ใช้ได้อีกด้วย