บทที่ 2
ทฤษฎีและวรรณกรรมปริทัศน์

ในบทนี้นำเสนอทฤษฎีที่ใช้ในงานวิทยานิพนธ์ ซึ่งมีหัวข้อดังนี้ เอกซ์เอ็มแอล เว็บเซอร์วิส เพียร์-ทู-เพียร์ และการบูรณาการข้อมูล รวมทั้งการศึกษาและค้นคว้างานวิจัยที่เกี่ยวข้องในหัวข้อของ การบูรณาการข้อมูลในงานชีวสารสนเทศศาสตร์ ที่มีความสัมพันธ์กับวิทยานิพนธ์นี้

1. เอกซ์เอ็มแอล (eXtensible Markup Language: XML)

กานดา สายแก้ว (2553) นิยามความหมายของเอกซ์เอ็มแอลคือ ภาษามาร์กอัป ซึ่งสามารถใช้งานโดยทั่วไปด้วยคุณสมบัติที่ผู้ใช้สามารถกำหนดความหมายของข้อมูลด้วยตนเอง ซึ่งทำให้เอกซ์เอ็มแอลเป็นภาษาพื้นฐานให้กับภาษาอื่นๆ ที่มีการใช้เฉพาะเจาะจงในแต่ละวัตถุประสงค์ที่แตกต่างกันออกไป

การนำเอกซ์เอ็มแอลมาใช้งาน สามารถทำได้โดยการใช้แท็กเป็นตัวกำกับและการตั้งชื่อแท็กที่สื่อถึงความหมายของข้อมูล ดังแสดงในภาพที่ 2.1

```
<?xml version="1.0"?>
<nation id="th">
    <name>Thailand</name>
    <location>Southeast Asia</location>
</nation>
```

ภาพที่ 2.1 ตัวอย่างเอกสารเอกซ์เอ็มแอล

การตรวจสอบความถูกต้องของเอกสาร (Validity) เป็นการตรวจสอบเอกสารเอกซ์เอ็มแอลเปรียบเทียบกับเอกสารที่อธิบายโครงสร้างนั้นๆ และคาดหวังไว้ว่าจะได้รับข้อมูลจากเอกสารเอกซ์เอ็มแอลในรูปแบบตามที่กำหนดไว้ในเอกสารที่อธิบายโครงสร้าง เอกสารเอกซ์เอ็มแอลสามารถจะมีเอกสารที่อธิบายโครงสร้างซึ่งอาจจะเขียนอยู่ในรูปภาษาดีทีดี (Document Type Definition: DTD) หรือเอกซ์เอ็มแอลสกีม่า (XML Schema)

เอกสารเอกซ์เอ็มแอลที่ถูกต้องตามกฎไวยากรณ์ของเอกซ์เอ็มแอลที่องค์กร W3C ระบุไว้นั้นจะเรียกเอกสารเอกซ์เอ็มแอลนั้นว่าเป็นเอกสารเอกซ์เอ็มแอลที่ถูกต้องตามกฎไวยากรณ์ (Well-formed XML) ซึ่งเอกสารเอกซ์เอ็มแอลที่ถูกต้องตามกฎไวยากรณ์นี้ ไม่จำเป็นต้องมีเอกสารเอกซ์เอ็มแอลสกีม่า หรือเอกสารดีทีดีกำกับ แต่เอกสารที่ทั้งถูกต้องตามกฎไวยากรณ์ของเอกซ์เอ็มแอลและโครงสร้างของเอกสารเอกซ์เอ็มแอลที่ได้อธิบายไว้ในเอกสารดีทีดีหรือเอกสารเอกซ์เอ็มแอลสกีม่า นั่นจะเรียกว่าเป็นเอกสารเอกซ์เอ็มแอลที่ถูกต้องตามกฎโครงสร้าง (Valid) ซึ่งเอกสารที่ถูกต้องตามกฎโครงสร้างนี้ จะต้องเป็นเอกสารที่ถูกต้องตามกฎไวยากรณ์ด้วยเช่นกัน

ในการแลกเปลี่ยนข้อมูลระหว่างองค์กรนั้นจำเป็นต้องมีการอธิบายโครงสร้างเอกสารเอกซ์เอ็มแอล เพื่อกำหนดรูปแบบของเอกสารเอกซ์เอ็มแอลที่จะส่งถึงกัน ซึ่งการอธิบายโครงสร้างเอกสารเอกซ์เอ็มแอลสามารถอธิบายได้ 2 วิธี อย่างใดอย่างหนึ่งคือ การอธิบายโครงสร้างเอกสารเอกซ์เอ็มแอลโดยใช้ภาษาดีทีดี (DTD) และการอธิบายโครงสร้างเอกสารเอกซ์เอ็มแอลโดยใช้เอกซ์เอ็มแอลสกีม่า

1.1 การอธิบายโครงสร้างเอกสารเอกซ์เอ็มแอลโดยใช้ภาษาดีทีดี

กานดา สายแก้ว (2553) ได้นิยามภาษาดีทีดีว่า เป็นภาษาหนึ่งที่สามารถนำไปใช้ในการอธิบายโครงสร้างเอกสารเอกซ์เอ็มแอล เพื่อกำหนดว่าจะมีอิลิเมนต์และแอตทริบิวต์อะไรบ้างที่จะปรากฏในเอกสารเอกซ์เอ็มแอล นอกจากนี้เอกสารที่อธิบายโครงสร้างเอกสารเอกซ์เอ็มแอล จะระบุอีกว่าแต่ละอิลิเมนต์จะมีเนื้อหาอะไรบ้างอยู่ภายในอิลิเมนต์

ยกตัวอย่างเอกสารเอกซ์เอ็มแอลในภาพที่ 2.1 สามารถอธิบายโครงสร้างโดยใช้ภาษาดีทีดีได้ดังภาพที่ 2.2 โดยส่วนที่เป็นตัวหนาในภาพเป็นส่วนประกาศประเภทของเอกสารที่มีรูทอิลิเมนต์ชื่อ "nation" และส่วนที่เหลือเป็นส่วนดีทีดีที่อธิบายโครงสร้างของเอกสารเอกซ์เอ็มแอล

```
<!DOCTYPE nation [
    <!ELEMENT nation (name, location)>
    <!ATTLIST nation id ID #REQUIRED >
    <!ELEMENT name (#PCDATA)>
    <!ELEMENT location (#PCDATA)>
]>
```

ภาพที่ 2.2 ตัวอย่างของดีทีดีจากเอกสารเอกซ์เอ็มแอลในภาพที่ 2.1

1.2 การอธิบายโครงสร้างเอกสารเอกซ์เอ็มแอลโดยใช้เอกซ์เอ็มแอลสกีม่า

กานดา สายแก้ว (2553) ได้นิยามความหมายของเอกซ์เอ็มแอลสกีม่าว่า เป็นเอกซ์เอ็มแอลประเภทหนึ่งที่ใช้ในการระบุโครงสร้างของเอกสารเอกซ์เอ็มแอล การแลกเปลี่ยนเอกสารระหว่างบุคคล หรือระหว่างองค์กร มักจะมีการใช้เอกซ์เอ็มแอลสกีม่า เพื่อกำหนดโครงสร้างของเอกสารเอกซ์เอ็มแอลที่ใช้ในการแลกเปลี่ยนซึ่งกันและกัน นอกจากนี้เอกซ์เอ็มแอลสกีม่าได้ถูกนำไปใช้ในการระบุโครงสร้างของข้อมูลเอกซ์เอ็มแอลซึ่งถูกส่งระหว่างผู้ให้บริการเว็บเซอร์วิสและผู้ขอรับบริการเว็บเซอร์วิส

ยกตัวอย่างเอกสารเอกซ์เอ็มแอลในภาพที่ 2.1 สามารถอธิบายโครงสร้างโดยใช้เอกซ์เอ็มแอลสกีม่าได้ดังภาพที่ 2.3

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="nation">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="name" type="xsd:string"/>
                <xsd:element name="location"
                             type="xsd:string"/>
            </xsd:sequence>
            <xsd:attribute name="id" type="xsd:ID"/>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

ภาพที่ 2.3 ตัวอย่างของเอกซ์เอ็มแอลสกีม่าจากเอกสารเอกซ์เอ็มแอลในภาพที่ 2.1

จากภาพที่ 2.3 เอกสารเอกซ์เอ็มแอลสกีม่าจะมีลักษณะเหมือนกับเอกสารเอกซ์เอ็มแอลทุกประการ เพียงแต่เอกสารเอกซ์เอ็มแอลสกีม่าจะมีรูทอิลิเมนต์ชื่อว่า `schema` ชื่อย่อว่า `xsd` โดยที่ `xsd` เป็นชื่อย่อของเนมสเปซ (namespace) `http://www.w3.org/2001/XMLSchema` ซึ่งชื่อย่อ `xsd` จะเป็นชื่ออะไรก็ได้ (โดยทั่วไปนิยมใช้ `xsd` หรือ `xs` แทนคำเต็มว่าเอกซ์เอ็มแอลสกีม่า) และมีอิลิเมนต์ชื่อ (`xsd:element`) `nation` ที่เป็นแบบชนิดซับซ้อน (`xsd:complexType`) และมีอิลิเมนต์ภายใต้

อิลิเมนต์ nation เป็นลำดับดังนี้คือ name และ location ซึ่งมีชนิดข้อมูลเป็นข้อความ (xsd:string) และภายในอิลิเมนต์ nation ยังมีแอตทริบิวต์ชื่อ id ที่มีชนิดข้อมูลเป็นไอดี (xsd:ID) อีกด้วย

ข้อเปรียบเทียบในการกำหนดโครงสร้างเอกสารเอกซ์เอ็มแอลด้วยภาษาดีทีดีและเอกซ์เอ็มแอลสกีม่า สามารถสรุปได้ดังตารางที่ 2.1

ตารางที่ 2.1 ข้อเปรียบเทียบระหว่างดีทีดีและเอกซ์เอ็มแอลสกีม่า

ข้อที่เหมือนกัน	ข้อที่แตกต่าง
- ทั้งดีทีดีและเอกซ์เอ็มแอลสกีม่า สามารถกำหนดโครงสร้างของเอกสารเอกซ์เอ็มแอล - สามารถกำหนดอิลิเมนต์และแอตทริบิวต์ซึ่งปรากฏในเอกสารเอกซ์เอ็มแอล - สามารถกำหนดความสัมพันธ์ระหว่างอิลิเมนต์ - สามารถกำหนดค่าดีฟอลต์ (Default) และค่าคงที่ของแอตทริบิวต์ได้	- เอกซ์เอ็มแอลสกีม่าเป็นเอกซ์เอ็มแอลประเภทหนึ่ง แต่ดีทีดีไม่ได้อยู่ในรูปแบบเอกซ์เอ็มแอล ดังนั้นจึงทำให้ผู้ที่ใช้เอกซ์เอ็มแอลสกีม่าไม่จำเป็นต้องเรียนรู้ไวยากรณ์ของภาษาใหม่ อีกทั้งสามารถใช้โปรแกรมที่อ่านเอกสารเอกซ์เอ็มแอลนำมาอ่านและประมวลผลเอกสารเอกซ์เอ็มแอลสกีม่าได้เลย - เอกซ์เอ็มแอลสกีม่าอนุญาตให้ผู้ใช้สามารถใช้ข้อมูลได้หลายรูปแบบตามเหมาะสม เช่น เลขจำนวนเต็ม (integer) และเลขจำนวนจริง (double) และผู้ใช้สามารถกำหนดชนิดข้อมูลขึ้นมาเอง ในขณะที่ดีทีดีสามารถใช้ชนิดข้อความได้เพียงข้อความอักขระที่ถูกนำไปประมวลผล (Parsed Character Data) และข้อความอักขระ (Character Data) - เอกซ์เอ็มแอลสกีม่าอนุญาตให้ผู้ใช้กำหนดหรือจัดกลุ่มอิลิเมนต์และแอตทริบิวต์ที่ใช้ด้วยกัน ทำให้ง่ายต่อการนำไปใช้ รวมถึงการกำหนดจำนวนครั้งต่ำสุดและสูงสุดของอิลิเมนต์ที่ปรากฏได้ แต่ดีทีดีไม่สามารถทำได้ การใช้ดีทีดีทำได้เพียงการกำหนดอิลิเมนต์ให้ปรากฏหนึ่งครั้ง ไม่ปรากฏ หรือปรากฏมากกว่าหนึ่งครั้ง เท่านั้น - เอกซ์เอ็มแอลสกีม่า มีส่วนสนับสนุนการใช้เนมสเปซของกลุ่มอิลิเมนต์ต่าง ๆ ที่ครอบคลุมและสามารถนำไปใช้ได้ง่ายดีทีดี ซึ่งทำให้ผู้ใช้เอกซ์เอ็มแอลสกีม่าสามารถนำกลุ่มของอิลิเมนต์ที่กำหนดโดยองค์กรหรือกลุ่มคนที่แตกต่างกันมาใช้ได้โดยง่าย

2. เว็บเซอร์วิส (Web Services)

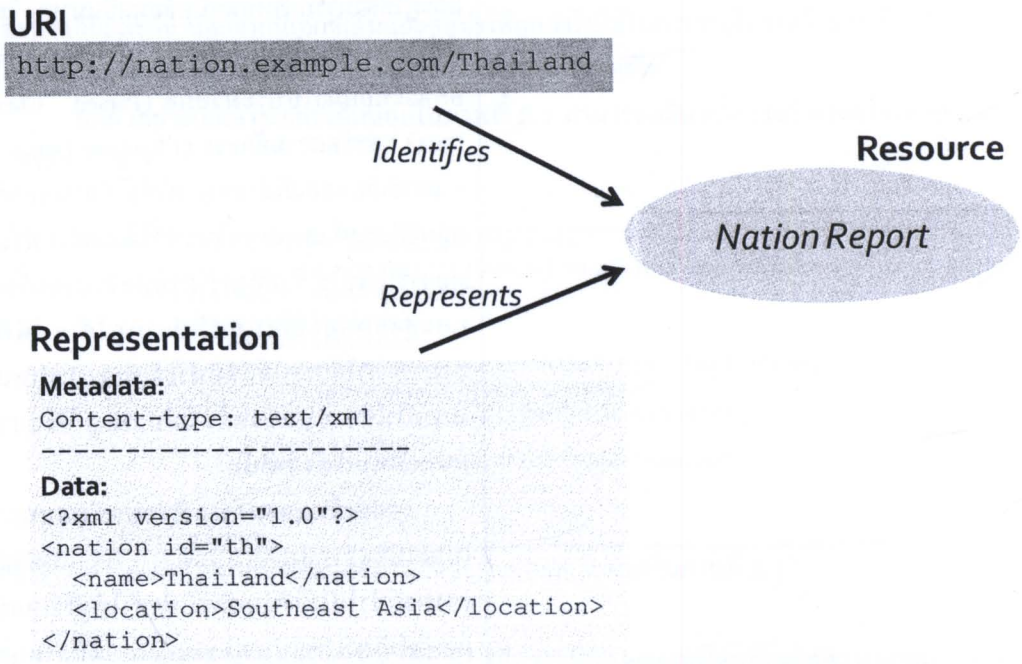
Booth, McCabe, Champion and Orchard (2004) ได้นิยามความหมายสถาปัตยกรรมของเว็บเซอร์วิส คือ แอปพลิเคชันหรือโปรแกรมที่ทำงานอย่างใดอย่างหนึ่ง ในลักษณะให้บริการ โดยจะถูกเรียกใช้งานจากแอปพลิเคชันอื่น ๆ ซึ่งการให้บริการจะมีเอกสารที่อธิบายคุณสมบัติของบริการกำกับไว้ โดยภาษาที่ถูกใช้เป็นสื่อในการแลกเปลี่ยนคือเอกซ์เอ็มแอล (XML) ทำให้เราสามารถเรียกใช้ส่วนประกอบใด ๆ ในระบบ หรือแพลตฟอร์มใด ๆ บนโปรโตคอลอินเทอร์เน็ต เช่น เอชทีทีพี (HTTP), เอชทีทีพีเอส (HTTPS), เอสเอสแอล (SSL) หรือ เอสเอ็มทีพี (SMTP) เป็นต้น

เว็บเซอร์วิสสามารถแบ่งเป็นสองกลุ่มได้แก่ เรสต์เว็บเซอร์วิส (REST Web Services) และโซพเว็บเซอร์วิส (SOAP Web Services)

2.1 เรสต์เว็บเซอร์วิส (REST Web Services)

เรสต์ (Representation State Transfer: REST) เป็นรูปแบบสถาปัตยกรรมบนเว็บที่ใช้ในการกระจายสื่อและแสดงถึงสถานะที่เปลี่ยนไปเมื่อมีการร้องขอผ่านยูอาร์ไอ (URI) บนโปรโตคอลเอชทีทีพี (HTTP) ซึ่งสถานะดังกล่าวอาจเป็นข้อความ รูปภาพ ไฟล์เสียง หรือข้อมูลใด ๆ โดยมีการดำเนินการผ่านมาตรฐานเก็ต (GET) โพสต์ (POST) พุต (PUT) หรือดีลีท (DELETE) สำหรับเอชทีทีพี (Fielding, 2000)

เรสต์เว็บเซอร์วิส เป็นการนิยามเพิ่มเติมจากเรสต์ เพื่อให้ครอบคลุมการใช้ยูอาร์ไอบนโปรโตคอลอื่น ๆ และอยู่ภายใต้สถาปัตยกรรมของเว็บที่มีการนิยามการแสดงผลของทรัพยากรที่ถูกระบุด้วยยูอาร์ไออยู่ในรูปแบบของเอกซ์เอ็มแอลดังภาพที่ 2.4 (Jacobs, 2003)



ภาพที่ 2.4 การระบุและแสดงผลของทรัพยากรของเรสต์เว็บเซอร์วิส

2.2 โซพเว็บเซอร์วิส (SOAP Web Services)

โซพเว็บเซอร์วิส เป็นเว็บเซอร์วิสที่ใช้โปรโตคอลโซพ (Simple Object Access Protocol: SOAP) ในการแลกเปลี่ยนข้อมูลระหว่างผู้ให้บริการและผู้ขอรับบริการ โดยรูปแบบของข้อมูลจะถูกนิยามในวิสเดิล (Web Services Description Language: WSDL) และสามารถค้นหาบริการได้ด้วยยูดีดีไอหรือทะเบียนที่เก็บรวบรวมบริการของเว็บเซอร์วิส (Universal Description, Discovery, and Integration: UDDI)

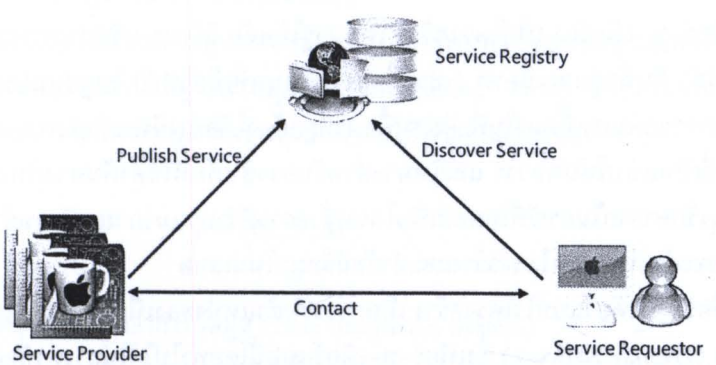
เทคโนโลยีพื้นฐานของโซพเว็บเซอร์วิสมีรายละเอียดโดยย่อดังนี้

(1) เอกซ์เอ็มแอล (XML) เป็นภาษามาตรฐานที่ทุกระบบสนับสนุน ทำให้ข้อมูลที่มีโครงสร้างของเอกซ์เอ็มแอล จะถูกนำไปประมวลผลต่ออย่างอัตโนมัติได้อย่างง่ายดาย เอกซ์เอ็มแอลจึงถูกนำมาใช้เป็นภาษามาตรฐานในการแลกเปลี่ยนข้อมูลของเว็บเซอร์วิส

(2) โซพ (SOAP) เป็นมาตรฐานโปรโตคอลหนึ่งที่มีรูปแบบที่เรียบง่ายและมีโครงสร้างสำหรับการแลกเปลี่ยนข้อมูลระหว่างผู้ให้บริการและผู้ขอรับบริการโดยใช้เอกซ์เอ็มแอลแลกเปลี่ยนข้อมูล (Box et al., 2000)

(3) วิสเดิล (WSDL) เป็นภาษามาตรฐานที่ใช้เอกซ์เอ็มแอลในการอธิบายการบริการในเว็บเซอร์วิส โดยจะอธิบายรายละเอียดเกี่ยวกับที่อยู่ บริการที่มี รูปแบบการเข้ารหัสข้อความ รูปแบบชนิดข้อความที่ต้องติดต่อสื่อสาร และโปรโตคอลที่ใช้ติดต่อของเว็บเซอร์วิส (Christensen, Curbera, Meredith & Weerawarana, 2001)

(4) ยูดีดีไอ (UDDI) เป็นทะเบียนของกลุ่มเว็บเซอร์วิสที่เก็บรายละเอียดข้อมูลธุรกิจหรือหน่วยงานอื่น ๆ รวมถึงรายละเอียดการทางเทคนิคของเว็บเซอร์วิสที่มาลงทะเบียนด้วย เพื่อให้ผู้ใช้สามารถสืบค้นข้อมูลธุรกิจหรือบริการที่ต้องการ และสามารถติดต่อขอดำเนินการใช้บริการได้โดยอัตโนมัติผ่านทางเว็บเซอร์วิส (Bellwood, 2002)



ภาพที่ 2.5 ภาพรวมสถาปัตยกรรมของโซพเว็บเซอร์วิส

จากภาพที่ 2.5 แสดงภาพรวมสถาปัตยกรรมของโซพเว็บเซอร์วิส โดยมีขั้นตอนตั้งแต่เริ่มจัดทำโซพเว็บเซอร์วิส ถึงการเรียกใช้บริการโซพเว็บเซอร์วิส ดังนี้

(1) ผู้ให้บริการจัดทำระบบหรือบริการที่เป็นเว็บเซอร์วิสขึ้นมา

(2) ผู้ให้บริการลงทะเบียนเว็บเซอร์วิสกับหน่วยงานที่ให้บริการระบบทะเบียนบริการ (UDDI หรือ Registry Service) โดยในขั้นตอนนี้ไม่จำเป็นเสมอไป

(3) ผู้ให้บริการระบุปลายทางของเอกสารวิสเดิลกับระบบทะเบียนบริการ (UDDI) ที่ได้ลงทะเบียนไว้ โดยในขั้นตอนนี้ไม่จำเป็นเสมอไป

(4) ผู้ใช้หรือผู้ร้องขอบริการทำการค้นหาระบบหรือบริการที่ต้องการจากระบบยูดีดีไอ โดยในขั้นตอนนี้ไม่จำเป็นเสมอไป

(5) เมื่อผู้ใช้หรือผู้ร้องขอบริการได้พบระบบหรือบริการที่ต้องการจะนำเอกสารวิสเดิลไปเรียนรู้วิธีการเรียกใช้ผ่านระบบของตน

(6) ผู้ใช้ หรือผู้ร้องขอบริการทำการติดต่อและเรียกใช้ระบบหรือบริการจากผู้ให้บริการได้โดยตรงผ่านโซพในระบบของตน

3. เพียร์-ทู-เพียร์ (Peer-to-Peer)

Schollmeier (2002) ได้ให้คำนิยามของเพียร์-ทู-เพียร์ไว้ว่า เป็นสถาปัตยกรรมเครือข่ายแบบกระจายเมื่อเพียร์ที่มีส่วนร่วมแบ่งปันบางส่วนของทรัพยากร เช่น การประมวลผล พื้นที่ ไฟล์ เครื่องพิมพ์ เป็นต้น ทรัพยากรที่ถูกแบ่งปันเหล่านี้สามารถเข้าถึงโดยเพียร์อื่น ๆ ภายในเครือข่ายโดยตรง

Androutsellis-Theotokis and Spinellis (2004) ได้สำรวจเกี่ยวกับเทคโนโลยีการกระจายเนื้อหาบนเครือข่ายเพียร์-ทู-เพียร์ให้คำนิยามไว้ว่า ระบบเพียร์-ทู-เพียร์เป็นการกระจายระบบ ประกอบด้วยโหนดที่จัดการตัวเองได้เชื่อมต่อกันในรูปแบบเครือข่าย โดยมีวัตถุประสงค์เพื่อแบ่งปันทรัพยากรให้ใช้ร่วมกันได้ เช่น เนื้อหา ไฟล์ การประมวลผล การจัดเก็บข้อมูล และแบนด์วิดธ์ เป็นต้น รวมถึงวัตถุประสงค์ในการปรับตัวเข้ากับความล้มเหลว การรองรับโหนดที่เพิ่มขึ้น ซึ่งระบบเพียร์-ทู-เพียร์จะต้องทำงานได้ในขณะที่ยังมีการเชื่อมต่อและมีประสิทธิภาพการทำงานโดยไม่จำเป็นต้องอาศัยศูนย์กลาง

ระบบเพียร์-ทู-เพียร์สามารถจัดหมวดหมู่ได้สองประเภท คือ ระบบเพียร์-ทู-เพียร์แบบใช้โครงสร้าง (Structured System) หรือใช้ดีเอชที (DHT-based) และระบบเพียร์-ทู-เพียร์แบบไร้โครงสร้าง (Unstructured System)

(1) เพียร์-ทู-เพียร์แบบใช้โครงสร้าง เป็นการใช้หลัก วิธีการ หรือกระบวนการ โดยจับคู่ระหว่างเนื้อหา (เช่น ไฟล์) กับที่อยู่ของเนื้อหา (เช่น ที่อยู่ของโหนดที่เก็บไฟล์) ในรูปแบบของตารางเส้นทางการกระจาย เพื่อให้สามารถค้นหาเนื้อหาในที่อยู่ของเนื้อหาที่ถูกต้องได้อย่างมีประสิทธิภาพ เหมาะสำหรับการขยายขนาดของระบบที่เป็นแบบโหนดถาวร และเหมาะสำหรับการสืบค้นที่ใช้คำค้นตรงกับเนื้อหา (Exact-Match Queries) แต่บำรุงรักษายากในกรณีที่โหนดที่เป็นโหนดชั่วคราวมีจำนวนมาก และมีการเข้าร่วม-การยกเลิกกลุ่มบ่อย เพราะระบบจะต้องปรับปรุงโครงสร้างและทำดัชนีข้อมูลใหม่เสมอ

(2) เพียร์-ทู-เพียร์แบบไร้โครงสร้าง เป็นการระบุตำแหน่งของเนื้อหา (เช่น ไฟล์) ที่ไม่เกี่ยวข้องกับโครงข่ายซ้อนทับ (Overlay Network) ซึ่งเพียร์-ทู-เพียร์แบบนี้โดยทั่วไปใช้ในการหาที่อยู่ของเนื้อหาที่ต้องการด้วยกระบวนการค้นหาต่าง ๆ เหมาะสำหรับโหนดที่เข้าร่วมระบบเป็นโหนดชั่วคราว คือ มีการเข้าร่วมหรือยกเลิกกลุ่มบ่อย บำรุงรักษาง่าย และสามารถใช้คำสืบค้นทั่ว ๆ ไปได้

ถึงแม้ในระบบเพียร์-ทู-เพียร์ที่แท้จริงจะเป็นระบบที่ไม่ต้องอาศัยศูนย์กลาง แต่ในทางปฏิบัติระบบเพียร์-ทู-เพียร์สามารถมีศูนย์กลางในระดับต่าง ๆ โดยสามารถแบ่งเป็นสามหมวดหมู่ตามสถาปัตยกรรมดังนี้

3.1 สถาปัตยกรรมที่ไม่มีศูนย์กลางอย่างแท้จริง (Purely Decentralized Architectures)

สถาปัตยกรรมนี้ ทุกโหนดในเครือข่ายสามารถทำหน้าที่เป็นทั้งเซิร์ฟเวอร์และไคลเอนท์ได้ในตัวโหนดเอง และไม่มีศูนย์กลางที่ทำหน้าที่จัดการกิจกรรมต่าง ๆ ของโหนดเหล่านั้น ในเพียร์-ทู-เพียร์แบบนี้

โหนดในเครือข่ายจะเรียกว่า "เซอร์เวนท์" (SERVENTS: SERVers + cliENTS) ที่ทำหน้าที่เป็นทั้งเซิร์ฟเวอร์และไคลเอนท์ในตัวโหนดเอง

3.2 สถาปัตยกรรมที่มีศูนย์กลางบางส่วน (Partially Centralized Architectures)

พื้นฐานของสถาปัตยกรรมแบบนี้มีลักษณะเดียวกับสถาปัตยกรรมที่ไม่มีศูนย์กลางอย่างแท้จริง แต่จะมีบางโหนดที่สำคัญทำหน้าที่เป็นศูนย์กลางจัดการดัชนีที่อยู่ไฟล์ของเพียร์ที่ศูนย์กลางนั้นดูแล โดยศูนย์กลางการจัดการเพียร์นี้เรียกว่า "โหนดพิเศษ" (Supernodes) ซึ่งบทบาทของโหนดพิเศษจะถูกกำหนดโดยเครือข่ายระบบนั้น ๆ อย่างไรก็ตามโหนดพิเศษจะไม่เป็นปัญหาในเรื่องความล้มเหลวที่จุดเดียว (Single Points of Failure) ในเครือข่ายเพียร์-ทู-เพียร์แบบนี้ เพราะโหนดพิเศษใหม่จะถูกกำหนดขึ้นใหม่โดยแทนที่โหนดพิเศษเดิมอีกครั้งโดยอัตโนมัติเมื่อระบบค้นพบความผิดพลาดของโหนดพิเศษเดิม

3.3 สถาปัตยกรรมที่ไม่มีศูนย์กลางแบบลูกผสม (Hybrid Decentralized Architectures)

ในสถาปัตยกรรมแบบนี้จะมีเซิร์ฟเวอร์ศูนย์กลาง (Center Server) เพื่ออำนวยความสะดวกในการทำงานระหว่างเพียร์ โดยการดูแลรายละเอียดของข้อมูลโดยย่อ (Meta-data) การอธิบายเกี่ยวกับการแบ่งปันไฟล์บนเพียร์โหนด (Peer Nodes) ถึงแม้จะเป็นการติดต่อของโหนดสุดท้ายกับโหนดสุดท้าย (End-to-End Interaction) หรือแลกเปลี่ยนไฟล์โดยตรงระหว่างเพียร์โหนด ซึ่งการติดต่อระหว่างเพียร์โหนดกับเซิร์ฟเวอร์ศูนย์กลางเป็นการติดต่อเพื่อค้นหาและระบุโหนดที่เก็บไฟล์ที่ต้องการ ในบางครั้งระบบนี้จะถูกเรียกว่า "เพียร์-ผ่าน-เพียร์" (Peer-through-Peer) หรือ "ใช้ตัวกลาง" (Broker Mediated) แต่ในสถาปัตยกรรมแบบนี้หากมีการจัดการควบคุมไม่ดีพออาจจะเกิดปัญหาเรื่องความล้มเหลวที่จุดเดียวในเครือข่ายเพียร์-ทู-เพียร์ขึ้นได้

4. การบูรณาการข้อมูล (Data Integration)

Lenzerini (2002) ได้นิยามการบูรณาการข้อมูลว่า เป็นการรวมข้อมูลที่อยู่ในแหล่งข้อมูลที่แตกต่างกันและเป็นการนำเสนอข้อมูลเหล่านี้ให้ผู้ใช้ผ่านมุมมองรวม (Unified View) โดยสามารถเขียนอยู่ในรูปแบบความสัมพันธ์ของระบบบูรณาการข้อมูล ซึ่งเป็นมุมมองรวมได้ดังความสัมพันธ์ที่ [2.1]

$$I = \langle G, S, M \rangle \qquad [2.1]$$

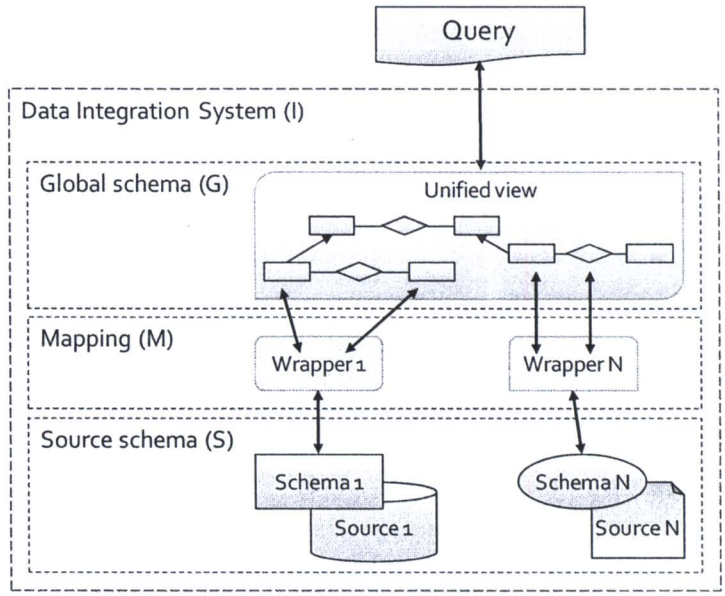
โดยที่ I คือ ระบบบูรณาการข้อมูล (Data Integration System)

G คือ โครงสร้างเอกสารกลางที่เป็นภาพรวมของแหล่งข้อมูลทั้งหมด (Global Schema)

S คือ โครงสร้างเอกสารของข้อมูลแต่ละแหล่งข้อมูล (Source Schema)

M คือ การเชื่อมโยงระหว่าง G และ S (Mapping)

จากความสัมพันธ์ที่ [2.1] สามารถแสดงภาพรวมของระบบบูรณาการข้อมูลได้ดังภาพที่ 2.6



ภาพที่ 2.6 ภาพรวมของระบบบูรณาการข้อมูล

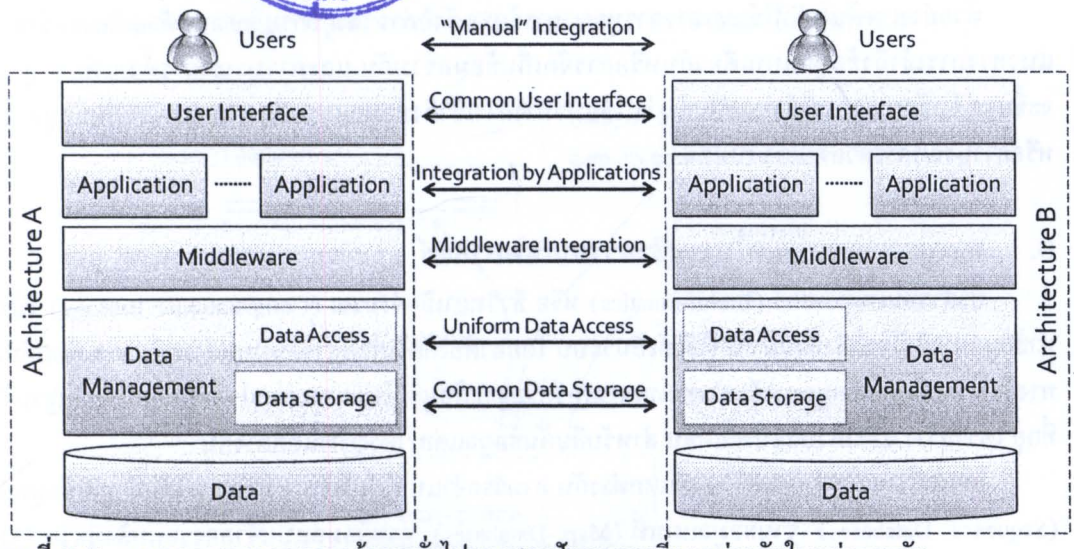
จากภาพที่ 2.6 ในส่วนของการเชื่อมโยงระหว่าง G และ S มีส่วนห่อหุ้ม (Wrapper) ที่แบ่งเป็น 2 มุมมอง ได้แก่

- มุมมองแบบภาพรวม (Global as View: GAV) โครงสร้างเอกสารกลางจะถูกนิยามขึ้นในขอบเขตของแหล่งข้อมูลที่มี ซึ่งคำสืบค้นจะถูกประมวลผลที่แหล่งข้อมูล
- มุมมองแบบท้องถิ่น (Local as View: LAV) แหล่งข้อมูลทั้งหมดจะถูกนิยามในขอบเขตของโครงสร้างเอกสารกลาง ซึ่งส่วนการเชื่อมโยงจะหาทุกวิธีการที่จะสร้างคำสืบค้นจากมุมมอง

งานวิจัยเกี่ยวกับเรื่องการบูรณาการข้อมูลมีมายาวนานกว่า 30 ปี โดย Ziegler and Dittrich (2004) นำเสนอปัญหาและแนวทางในการบูรณาการข้อมูลในช่วง 30 ปีที่ผ่านมา ที่มีเป้าหมายในการบูรณาการข้อมูลในระบบที่แตกต่างกัน ให้อยู่ในมุมมองรวมแก่ผู้ใช้งาน ซึ่งในมุมมองของผู้ใช้จะเสมือนใช้งานเพียงระบบเดียว แต่ในความเป็นจริงผู้ใช้ได้ติดต่อกับระบบที่แตกต่างกัน

โดยทั่วไป ข้อมูลไม่ได้ถูกออกแบบสำหรับการบูรณาการ ดังนั้นเมื่อต้องการบูรณาการเข้ากับระบบที่แตกต่างกัน ต้องมีการปรับส่วนเชื่อมโยงให้เหมาะสม เพื่อให้ข้อมูลดูเป็นอันหนึ่งอันเดียวกันทั้งระบบ ซึ่งปัญหาที่พบในการบูรณาการข้อมูลบนระบบที่แตกต่างกันอาจขึ้นอยู่กับ

- มุมมองทางสถาปัตยกรรมของระบบข้อมูล
- เนื้อหาและการทำงานของระบบที่เป็นส่วนประกอบ
- ชนิดของข้อมูลที่ถูกจัดการโดยระบบ เช่น ข้อมูลตัวเลข ข้อมูลสื่อต่าง (แบบมีโครงสร้าง กึ่งโครงสร้าง และไม่มีโครงสร้าง) เป็นต้น
- ข้อกำหนดเกี่ยวกับการจัดการตนเองของระบบ
- จุดประสงค์การใช้งานระบบบูรณาการข้อมูล เช่น การเข้าเข้าถึงแบบอ่านอย่างเดียว หรือสามารถเขียนได้ด้วย เป็นต้น
- ความต้องการประสิทธิภาพการทำงาน
- ทรัพยากรที่มีอยู่ เช่น เวลา เงิน ทรัพยากรบุคคล ความรู้ เป็นต้น



ภาพที่ 2.7 แนวทางการบูรณาการข้อมูลทั่วไปของสถาปัตยกรรมที่แตกต่างกันในแต่ละระดับ

Ziegler and Dittrich (2004) ได้เสนอแนวทางการบูรณาการข้อมูลของสถาปัตยกรรมที่แตกต่างกันในแต่ละระดับ ดังตัวอย่างในภาพที่ 2.7 ซึ่งมีรายละเอียดดังนี้

(1) การบูรณาการด้วยตนเอง (Manual Integration) ในระดับนี้ผู้ใช้จะจัดการกับข้อมูลที่เกี่ยวข้องโดยตรง และทำการรวมข้อมูลเอง นั่นคือ ผู้ใช้จะมีการจัดการกับส่วนติดต่อที่แตกต่างกันและชุดคำสั่งสืบค้นที่แตกต่างกันด้วย นอกจากนี้ผู้ใช้จำเป็นต้องมีความรู้เกี่ยวกับข้อมูลที่เข้าถึงด้วย

(2) การบูรณาการด้วยส่วนติดต่อผู้ใช้ทั่วไป (Common User Interface) ในกรณีนี้ ผู้ใช้สามารถเข้าถึงข้อมูลผ่านส่วนติดต่อทั่วไป เช่น เว็บเบราว์เซอร์ ซึ่งจะให้ข้อมูลที่เหมือนกันในความรู้สึกของผู้ใช้ ในความเป็นจริง ข้อมูลที่ได้จากระบบที่เกี่ยวข้องยังคงแสดงผลแยกต่างหาก แต่การรวมข้อมูลและการทำให้ข้อมูลเป็นเนื้อเดียวกันยังมีการกระทำโดยผู้ใช้ เช่น เครื่องมือในการค้นหา

(3) การบูรณาการด้วยแอปพลิเคชัน (Integration by Applications) วิธีการนี้จะใช้โปรแกรมประยุกต์ที่สามารถเข้าถึงแหล่งข้อมูลที่หลากหลายและให้ผลลัพธ์ทั้งหมดแก่ผู้ใช้ ซึ่งแนวทางนี้เหมาะสำหรับระบบส่วนประกอบขนาดเล็ก

(4) การบูรณาการด้วยมิดเดิลแวร์ (Integration by Middleware) โดยทั่วไปจะใช้ในการแก้ปัญหาลักษณะเฉพาะของการรวมกันแล้ว ในกรณีที่ในระบบมีเครื่องมือมิดเดิลแวร์ที่แตกต่างกัน จะต้องได้นำมารวมกันเพื่อสร้างระบบบูรณาการ

(5) การบูรณาการด้วยการเข้าถึงข้อมูลแบบเดียวกัน (Uniform Data Access) ในกรณีนี้ การรวมตรรกะของข้อมูลถูกทำให้เสร็จสิ้นที่ระดับการเข้าถึงข้อมูล การใช้งานของแอปพลิเคชัน ทั่วไปจะเข้าถึงข้อมูลเสมือนได้ในระดับนี้เท่านั้น ซึ่งเป็นระดับที่แสดงมุมมองรวมของข้อมูลที่กระจายกันอยู่ทางกายภาพ อย่างไรก็ตามการแสดงมุมมองรวมของข้อมูลทางการภาพที่ถูกรวมจะใช้เวลานาน เพราะจะใช้เวลาตั้งแต่การเข้าถึงข้อมูล การทำข้อมูลให้เป็นชุดเดียว และการรวมข้อมูล

(6) การบูรณาการด้วยการจัดเก็บข้อมูลร่วมกัน (Common Data Storage) ในระดับนี้ การรวมข้อมูลทางกายภาพจะถูกนำไปแปลงเพื่อจัดเก็บข้อมูลใหม่ ซึ่งแหล่งข้อมูลท้องถิ่น (Local Sources) สามารถยกเลิกหรือยังมีอยู่ในระบบก็ได้ โดยทั่วไปการรวมข้อมูลทางกายภาพจะสามารถเข้าถึงข้อมูลได้รวดเร็ว อย่างไรก็ตามหากแหล่งข้อมูลท้องถิ่นถูกยกเลิกไป การเข้าถึงข้อมูลเหล่านั้นจะต้องถูกย้ายไปยังที่เก็บข้อมูลใหม่เช่นกัน

ตัวอย่างเทคโนโลยีที่ใช้แนวทางการบูรณาการข้อมูลดังกล่าว เช่น การบูรณาการด้วยเว็บเซอร์วิส ซึ่งใช้แนวทางการเข้าถึงข้อมูลแบบเดียวกันหรือการจัดเก็บข้อมูลร่วมกัน และการบูรณาการด้วยเพียร์-ทู-เพียร์ จะขึ้นอยู่กับลักษณะการใช้งาน เป็นต้นว่า ใช้แนวทางการเข้าถึงข้อมูลแบบเดียวกัน การจัดเก็บข้อมูลร่วมกัน หรือการบูรณาการด้วยตนเอง เป็นต้น

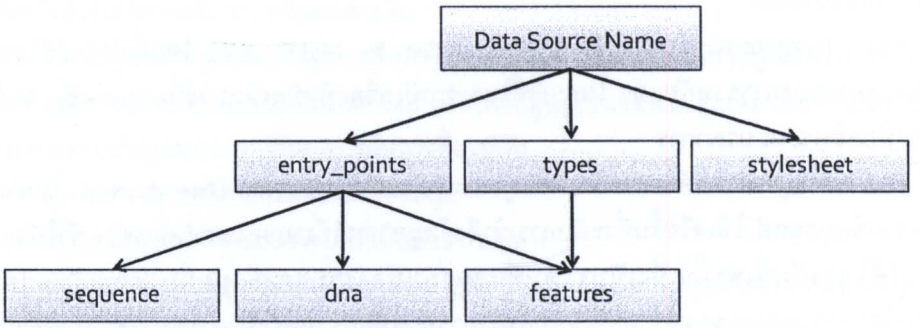
5. การบูรณาการข้อมูลในงานชีวสารสนเทศศาสตร์

ชีวสารสนเทศศาสตร์ (Bioinformatics) หรือ ชีววิทยาเชิงคำนวณ (Computational Biology) เป็นการนำข้อมูลทางชีววิทยาจำนวนมากมาใช้ให้เป็นระบบ โดยอาศัยเทคโนโลยีสารสนเทศเข้ามาจัดการเพื่อแก้ปัญหาทางชีววิทยา ซึ่งฐานข้อมูลทางชีวสารสนเทศศาสตร์หรือฐานข้อมูลจีโนมของแต่ละโครงการถูกเก็บในฐานข้อมูลที่อยู่ โครงสร้าง และมีเว็บแอปพลิเคชัน สำหรับสืบค้นข้อมูลแต่ละโครงการแตกต่างกัน

โครงสร้างของฐานข้อมูลจีโนมที่แตกต่างกัน สามารถจำแนกได้เป็น 5 หมวดหมู่ ได้แก่ ฐานข้อมูลลำดับ (Sequence Databases) ฐานข้อมูลแผนที่ (Map Databases) ฐานข้อมูลแบบจำลองระบบสิ่งมีชีวิต (Model Organism Databases) ฐานข้อมูลบรรณานุกรม (Bibliographic Databases) และฐานข้อมูลของฐานข้อมูล (Databases of Databases) (Cheang, Choi & Tang, 1994) และเนื่องจากโครงสร้างข้อมูลที่แตกต่างกัน จึงได้มีการนำเอาเอกซ์เอ็มแอลเข้ามาใช้ในการแลกเปลี่ยนและบูรณาการข้อมูล

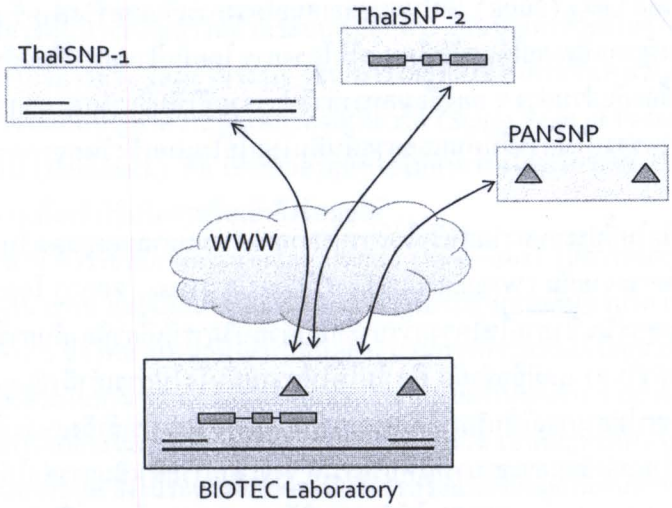
แนวทางในการนำเอกซ์เอ็มแอลมาใช้ในงานชีวสารสนเทศศาสตร์เพื่อบูรณาการข้อมูลครั้งแรก ถูกเสนอโดย Achard, Vaysseix and Barillot (2001) ในการนำมาใช้อธิบายและแลกเปลี่ยนข้อมูลชีววิทยา ระหว่างแอปพลิเคชัน ซึ่งการใช้เอกซ์เอ็มแอลในยุคนั้นมีความใกล้เคียงกับฐานข้อมูลเชิงวัตถุมาก

ระบบที่ใช้เอกซ์เอ็มแอลในงานชีวสารสนเทศศาสตร์ระบบแรก ถูกนำเสนอโดย Dowell, Jokerst, Day, Eddy & Stein (2001) ซึ่งเป็นระบบสัญลักษณ์ทางชีวสารสนเทศศาสตร์แบบกระจาย (Distributed Annotation System: DAS) หรือแดส ระบบนี้ใช้เอกซ์เอ็มแอลเป็นตัวกลางในการแลกเปลี่ยนข้อมูลตามโครงสร้างที่กำหนดไว้ตามมาตรฐาน โดยการสืบค้นข้อมูลจะสืบค้นผ่านทางยูอาร์แอล (URL) และให้ผลลัพธ์กลับมาเป็นเอกซ์เอ็มแอลตามโครงสร้างเอกสารที่ระบุไว้ในมาตรฐาน โดยลำดับการทำงานของชุดคำสั่งในระบบแสดงในภาพที่ 2.8 ซึ่งปัจจุบันโครงสร้างเอกสารของแดสใช้มาตรฐานรุ่นที่ 1.53 (Stein, Eddy & Dowell, 2002)



ภาพที่ 2.8 ลำดับการทำงานของชุดคำสั่งในระบบแดส

ระบบแดสมีแนวคิดในการบูรณาการข้อมูลชีวสารสนเทศศาสตร์ดังแสดงในภาพที่ 2.9



ภาพที่ 2.9 แนวคิดในการบูรณาการข้อมูลของแดส

จากภาพที่ 2.9 มีชุดข้อมูลสนิปของคนไทยกระจายอยู่บนระบบ 3 แหล่งข้อมูล ซึ่งหากแลปของไบโอเทคต้องการดูข้อมูลสนิปคนไทยที่มีทั้งหมดในระบบ ไคลเอนท์จะทำการดึงข้อมูลจากทุกแหล่งข้อมูลที่มีแล้วแสดงข้อมูลที่ถูกบูรณาการแล้ว

ในปัจจุบัน มีการพัฒนาศูนย์กลางเพื่อรวบรวมโครงการทางชีวสารสนเทศศาสตร์ด้วยแดส โดยศูนย์กลางนี้มีโครงการที่ใช้ระบบของแดสอยู่มากกว่า 250 โครงการ และมีโปรแกรมประยุกต์ที่ใช้ร่วมกับระบบของแดสมากกว่า 10 โปรแกรม (Prlić, Down, Kulesha, Finn & Kähäri, 2007) แต่ระบบแดสยังมีข้อจำกัดอยู่สองประการ คือ โครงสร้างเอกสารมาตรฐานเป็นแบบดีทีดี และการสืบค้นข้อมูลผ่านการร้องขอบนยูอาร์แอล ดังนี้

ข้อจำกัดที่ 1 : โครงสร้างเอกสารมาตรฐานของแดสรุ่นที่ 1.53 ได้กำหนดไว้เป็นแบบดีทีดี และบางโปรแกรมประยุกต์ไม่ได้ให้บริการแดสตามโครงสร้างเอกสารมาตรฐาน ทำให้โปรแกรมประยุกต์บางตัวทำงานผิดพลาดด้วยข้อผิดพลาดนี้

ข้อจำกัดที่ 2 : การสืบค้นข้อมูลผ่านการร้องขอบนยูอาร์แอล และส่งผลลัพธ์กลับมาในรูปแบบของข้อจำกัดในข้อที่ 1 ทำให้ระบบแดสไม่สามารถใช้ร่วมกับเครื่องมือในการสร้างขั้นตอนการทำงาน (Work Flow Application) ทั่วไป ซึ่งหากจะใช้เครื่องมือเหล่านั้นจะต้องเขียนโปรแกรมเพิ่มเติมทำให้เสียเวลา

Stein (2002) ได้นำเสนอบทความการใช้แบบจำลองของเว็บเซอร์วิสเข้ามาใช้ในงานชีวสารสนเทศศาสตร์เพื่อให้สามารถทำงานและแลกเปลี่ยนข้อมูลได้โดยอัตโนมัติตามสถาปัตยกรรมเว็บเซอร์วิสเป็นครั้งแรก และได้กล่าวถึงวิธีการแลกเปลี่ยนข้อมูลโดยอัตโนมัติที่ผ่านมา ด้วยการสร้างโปรแกรมสคริปต์เพื่อไปอ่านข้อมูลจากหน้าเว็บที่เรียกว่า “สกรีนสแครปปิง” แล้วนำข้อมูลที่ได้มาประมวลผลหรือวิเคราะห์ข้อมูลตามที่ต้องการ แต่ด้วยวิธีการนี้มีข้อจำกัดคือ เมื่อหน้าเว็บที่โปรแกรมสคริปต์ไปอ่านข้อมูลเกิดการเปลี่ยนแปลง จะต้องทำการแก้ไขโปรแกรมสคริปต์เพื่อให้เหมาะสมกับหน้าเว็บใหม่ จึงจะสามารถนำข้อมูลมาประมวลผลหรือวิเคราะห์ข้อมูลต่อไปได้

การใช้เทคโนโลยีเว็บเซอร์วิสที่มีส่วนติดต่อมาตรฐาน มีเอกสารอธิบายบริการและโครงสร้างข้อมูล และใช้เอ็กซ์เอ็มแอล จึงเหมาะแก่การใช้โปรแกรมภาษาใด ๆ ที่สามารถอ่านภาษาเอ็กซ์เอ็มแอลให้มีการทำงาน

อัตโนมัติได้ ซึ่งจะช่วยให้นักวิจัยสามารถนำข้อมูลมาบูรณาการและวิเคราะห์ข้อมูลได้เร็วขึ้น เพราะแอปพลิเคชันที่พัฒนาขึ้นมาจะสามารถทำงานและแลกเปลี่ยนข้อมูลได้โดยอัตโนมัติ

Wilkinson and Links (2002) ได้นำเสนอข้อเสนอโครงการเว็บเซอร์วิสสำหรับงานด้านชีววิทยา (ไบโอโมบี้) ที่ใช้ซอฟต์แวร์แบบเปิดเผยโค้ดรหัสทั้งหมดในโครงการ โดยมีเป้าหมายเพื่อแก้ปัญหาการค้นหาและการเรียกข้อมูลที่เกี่ยวข้องกับส่วนต่าง ๆ ของข้อมูลทางชีววิทยาจากผู้ให้บริการและบริการที่หลากหลาย ด้วยการสร้างส่วนติดต่อมาตรฐานในการสืบค้นและการค้นคืนโดยใช้โมเดลเชิงวัตถุจากการวิเคราะห์ (Consensus Object Models)

โครงการไบโอโมบี้ประสบความสำเร็จในการบูรณาการความหลากหลายของเว็บเซอร์วิสในงานชีวสารสนเทศศาสตร์ที่กระจัดกระจายกัน (Wilkinson, Schoof, Ernst & Haase, 2005) โดยมีการทำงานตามรูปแบบสถาปัตยกรรมเว็บเซอร์วิส ซึ่งภายในโครงการประกอบด้วยบริการที่เกี่ยวข้องกับการวิเคราะห์ข้อมูลชีววิทยาและฐานข้อมูลทางชีววิทยา แต่มีข้อจำกัด คือ ไม่ได้ให้ความมั่นใจในการเข้าถึงข้อมูลของผู้ใช้ ในกรณีที่บางฐานข้อมูลทางชีววิทยาไม่สามารถให้บริการได้ หรือการเพิ่มช่องทางในการเข้าถึงแหล่งข้อมูลให้รวดเร็วขึ้น ซึ่งทำให้ผู้ใช้ต้องรอจนกว่าแหล่งข้อมูลจะสามารถให้บริการได้ จึงจะสามารถดำเนินการต่อได้

Neerincx and Leunissen (2005) ได้นำเสนอวิวัฒนาการของเว็บเซอร์วิสในงานชีวสารสนเทศศาสตร์ตั้งแต่เริ่มมีความพยายามให้โปรแกรมทำงานระหว่างโปรแกรมโดยอัตโนมัติผ่านส่วนติดต่อเว็บทั่วไปที่ใช้ภาษาเอชทีเอ็มแอล (HTML) โดยใช้วิธีการสร้างภาษาสคริปต์ไปแปลงข้อมูลจากเว็บ ซึ่งเรียกว่า “สกรีนสแครปปิง” (Screen Scrapping) และหลังจากที่เว็บเซอร์วิสเข้ามามีบทบาท จึงได้เปลี่ยนจากการติดต่อโดยใช้เอชทีเอ็มแอลไปเป็นเอกซ์เอ็มแอลในการแลกเปลี่ยนข้อมูลแทน และใช้เว็บเซอร์วิสเป็นเทคโนโลยีที่ใช้แลกเปลี่ยนข้อมูล โดยในปัจจุบันมีโครงสร้างเอกสารเอกซ์เอ็มแอลสำหรับข้อมูลชีววิทยามากมายและรวมถึงแดสด้วย

Pillai, et al. (2005) ได้นำเสนอการใช้โซพเว็บเซอร์วิสในสถาบันชีวสารสนเทศศาสตร์แห่งยุโรป (European Bioinformatics Institute: EBI) โดยให้เหตุผลการใช้งานว่า สามารถทำงานผ่านไฟร์วอลและมาตรฐานเอชทีทีพีธรรมดาได้ อีกทั้งยังมีเอกสารในการอธิบายบริการและชนิดข้อมูล และมีการแลกเปลี่ยนข้อมูลผ่านโซพที่เป็นเอกซ์เอ็มแอลได้ ซึ่งการนำเสนอครั้งนี้ได้นำเสนอบริการการให้ข้อมูลและการประมวลผลข้อมูลผ่านทางโซพเว็บเซอร์วิส ต่อมา Labarga, Valentin, Anderson & Lopex (2007) ได้นำเสนอการใช้งานเว็บเซอร์วิสที่มีอยู่ในสถาบันแห่งนี้ ซึ่งได้มีการพัฒนาเว็บเซอร์วิสที่ให้บริการฐานข้อมูลมากกว่า 200 ฐานข้อมูล และมีบริการเว็บเซอร์วิสของแอปพลิเคชันทางด้านชีวสารสนเทศศาสตร์มากกว่า 150 แอปพลิเคชัน โดยทั้งหมดนี้ผู้ใช้สามารถจะบูรณาการข้อมูล หรือสร้างขั้นตอนการทำงานอัตโนมัติตามความต้องการได้ แต่ยังมีข้อจำกัด คือ การค้นหาฐานข้อมูลและบริการดังกล่าว กระทำผ่านการค้นหาของยูดีดีไอ ซึ่งเป็นแบบศูนย์กลาง ดังนั้นหากยูดีดีไอไม่สามารถให้บริการได้ ผู้ใช้ก็จะไม่สามารถค้นหาฐานข้อมูลและบริการได้

กล่าวโดยสรุป ในงานวิจัยต่าง ๆ ดังกล่าวพบประเด็นปัญหา 3 ประเด็น ดังนี้

(1) ปัญหาของการใช้เอกสารดีทีดีในการอธิบายชนิดข้อมูล มีบางงานวิจัยใช้เอกสารดีทีดีในการนิยามความหมายของข้อมูล ซึ่งการอธิบายชนิดข้อมูลของงานวิจัยนี้ มีความกำกวม และไม่ชัดเจน ทำให้ผู้ที่นำเอกสารดีทีดีไปใช้งานต่อ ต้องไปแปลความหมายชนิดข้อมูลและเขียนโปรแกรมจัดการกับข้อผิดพลาดที่อาจเกิดขึ้น เนื่องจากโครงการต่าง ๆ ที่เข้าร่วมกับงานวิจัยนี้มีการแปลความหมายชนิดข้อมูลจากเอกสารดีทีดีไม่เหมือนกัน

ปัญหาอีกประการของการใช้เอกสารดีทีดีในการอธิบายชนิดข้อมูลคือ การที่ไม่สามารถนำไปใช้งานร่วมกับโปรแกรมประเภทสร้างขั้นตอนการทำงานโดยอัตโนมัติได้ทันที เนื่องจากโปรแกรมประเภทนี้จะใช้เอกสารอธิบายชนิดข้อมูลที่เรียกว่า “เอ็กซ์เอ็มแอลสกีมา” ซึ่งสามารถอธิบายข้อมูลได้ละเอียดกว่า อนุญาตให้

ผู้ใช้สร้างชนิดข้อมูลเองได้ และเป็นที่นิยมใช้ในเว็บเซอร์วิสเพื่ออธิบายโครงสร้างบริการและชนิดข้อมูลในปัจจุบัน ในขณะที่เอกสารดีทีดีไม่อนุญาตให้ผู้ใช้สร้างชนิดข้อมูลใหม่ได้

(2) ปัญหาของการค้นหาเว็บเซอร์วิสด้วยระบบศูนย์กลาง เป็นปัญหาที่พบในงานวิจัยส่วนใหญ่ที่มีระบบศูนย์กลางเพื่อค้นหาบริการหรือข้อมูลในระบบ เช่น การใช้ยูดีดีไอในการค้นหาเว็บเซอร์วิส ซึ่งการใช้ศูนย์กลางในการค้นหาอาจจะเกิดปัญหาความล้มเหลวที่จุดเดียวได้ (Single Point of Failure) และปัญหาการเข้าถึงที่จุดเดียวมากเกินไป (Bottleneck) คือ เซิร์ฟเวอร์ยูดีดีไอไม่สามารถให้บริการได้ ทำให้ผู้ใช้ไม่สามารถค้นหาเว็บเซอร์วิสที่ต้องการเพื่อเข้าใช้บริการหรือเข้าถึงข้อมูลได้

การแก้ปัญหานี้ ควรใช้วิธีการกระจายข้อมูลโดยย่อ (Meta-data) ในการค้นหาบริการหรือข้อมูลไปไว้ยังโหนดต่าง ๆ ที่อยู่ในระบบ และเมื่อเกิดปัญหาดังกล่าว ผู้ใช้จะสามารถค้นหาบริการหรือข้อมูลได้จากข้อมูลโดยย่อตามโหนดต่าง ๆ ซึ่งวิทยานิพนธ์นี้ได้นำเสนอกระบวนการที่จะช่วยแก้ปัญหาดังกล่าวด้วยการใช้แนวคิดของเพียร์-ทู-เพียร์ช่วยในการกระจายข้อมูลโดยย่อของระบบไปยังโหนดต่าง ๆ อย่างเท่าเทียมกัน

(3) ปัญหาของการจัดการในการเข้าถึงหรือใช้บริการเว็บเซอร์วิส เป็นปัญหาที่พบในงานวิจัยส่วนใหญ่ที่มีระบบให้บริการและค้นหาข้อมูล ซึ่งในระบบนั้นอาจจะมีบริการหรือแหล่งข้อมูลที่ให้บริการซ้ำกันอยู่ แต่ยังไม่มีงานวิจัยใดที่นำเสนอเรื่องการจัดการและแนะนำผู้ใช้ เพื่อให้ใช้บริการหรือเข้าถึงแหล่งข้อมูลจากผลลัพธ์การค้นหาให้รวดเร็วที่สุด

การแก้ปัญหานี้ วิทยานิพนธ์ได้นำเสนอกระบวนการตัดสินใจ เพื่อแนะนำเว็บเซอร์วิสที่มีอยู่ซ้ำกันให้ผู้ใช้ได้พิจารณาก่อนตัดสินใจเข้าถึงหรือใช้บริการเว็บเซอร์วิสนั้น ๆ

จากประเด็นปัญหาทั้ง 3 ประเด็นดังกล่าว วิทยานิพนธ์นี้ได้นำเสนอวิธีการแก้ปัญหาด้วยการใช้เว็บเซอร์วิสร่วมกับเพียร์-ทู-เพียร์ โดยใช้เอกซ์เอ็มแอลสกีมาในการอธิบายโครงสร้างและชนิดข้อมูล และมีการกระจายข้อมูลโดยย่อของเว็บเซอร์วิสที่อยู่ในระบบไปยังโหนดต่าง ๆ ทำให้สามารถค้นหาเว็บเซอร์วิสได้ แม้ในกรณีที่ศูนย์กลางการค้นหาไม่สามารถให้บริการได้ อีกทั้งระบบยังมีการจัดการในการเข้าถึงเว็บเซอร์วิสที่ซ้ำกันในระบบ โดยจะแนะนำเว็บเซอร์วิสที่ผู้ใช้สามารถเข้าถึงได้เร็วที่สุด