# บทที่ 2
# งานวิจัยและทฤษฎีที่เกี่ยวข้อง

## 2.1 งานวิจัยที่เกี่ยวข้อง

### 2.1.1 WebTrader: Discovery and Programmed Access to Web-Based Services [7]

เว็บเทรดเดอร์เป็นการพัฒนาต้นแบบของตัวกลางในการค้นหาบริการบนเว็บที่มีคำอธิบายไม่ขึ้นกับสถาปัตยกรรมระบบกระจายใดๆ โดยการนำเสนอคำอธิบายในรูปของเอ็กซ์เอ็มแอลแทนที่คำอธิบายของแต่ละสถาปัตยกรรมเช่น คอร์บา จาวาอาร์เอ็มไอ (Java RMI) หรือ ซีจีไอ-สคริปต์ (CGI-Script) เว็บเทรดเดอร์นั้นพัฒนาขึ้นอยู่บนพื้นฐานของบริการเทรดเดอร์ทั่วไป นั่นคือผู้ให้บริการจะทำการโฆษณาบริการของตนเองผ่านทางบริการเทรดเดอร์ ส่วนผู้รับบริการสามารถที่จะค้นหาบริการโดยการร้องขอไปยังบริการเทรดเดอร์ ผู้ให้บริการจะทำการโฆษณาบริการของตนเองโดยใช้หน้าโฆษณาบริการ (Service Advertisement Page (SAP)) ที่มีเอกสารเอ็กซ์เอ็มแอล โดยเอกสารนี้จะมีข้อมูลสามส่วนคือ ส่วนต่อประสาน เมต้าดาต้า (Metadata) ที่เป็นคุณสมบัติ และคำสำคัญในการค้นหา (Search Keywords) ผู้รับบริการจะสามารถค้นหาบริการผ่านทางเว็บเทรดเดอร์ โดยใช้การส่งเอกสารเอ็ชทีเอ็มแอล (Hypertext Markup Language (HTML)) ที่มีส่วนหนึ่งเป็นเอกสารเอ็กซ์เอ็มแอล โดยเอกสารเอ็กซ์เอ็มแอลนี้จะต้องเป็นไปตามคำอธิบายชนิดเอกสารที่เรียกว่าดีทีดี (Document Type Description (DTD)) ที่ถูกกำหนดไว้แล้ว จากนั้นเว็บเทรดเดอร์จะทำการเปรียบเทียบและหาบริการที่ตรงกับความต้องการของผู้รับบริการ และส่งข้อมูลที่อยู่อ้างอิงและรายละเอียดของบริการคืนให้กับผู้ขอรับบริการ

แม้ว่าเว็บเทรดเดอร์จะสามารถใช้ในการค้นหาบริการบนเว็บโดยมีคำอธิบายไม่ขึ้นกับสถาปัตยกรรมระบบกระจายใดๆ อย่างไรก็ตามคำอธิบายบริการของเว็บเทรดเดอร์นั้นไม่สามารถนำไปอธิบายข้อมูลบริการของคอร์บาได้ทั้งหมด และไม่สามารถทำตามข้อกำหนดมาตรฐานของสถาปัตยกรรมคอร์บาได้ รวมถึงคำอธิบายบริการที่เป็นเอกสารเอ็กซ์เอ็มแอลยังต้องถูกสร้างด้วยมืออีกด้วย

### 2.1.2 Service Location in an Open Distributed Environment [3]

แนวความคิดในการค้นหาที่อยู่ของบริการ (Service Location) ที่นำเสนอโดยงานวิจัยนี้เป็นตัวแบบที่ทำให้การค้นหาที่อยู่ของบริการที่อยู่ภายใต้สถาปัตยกรรมที่แตกต่างกันทำได้สมบูรณ์ขึ้น โดยตัวแบบเสนอการเก็บข้อมูลชื่อของส่วนต่อประสานของบริการ (Interface Name) และข้อมูลวิธีในการเข้าถึง (Access Information) ของบริการซึ่งเก็บอยู่ในบริการค้นหาที่อยู่นี้

ข้อมูลชื่อของส่วนต่อประสานของบริการจะประกอบไปด้วย

- ชื่อของระบบของบริการ (Naming System Type) เช่นบริการไดเร็กทอรี่ของดีซีอี (DCE Directory Service)

- สิ่งแวดล้อมที่อ้างถึงของบริการ (Context Reference) เช่นที่อยู่ของบริการในรูปยูอาร์ไอ (Uniform Resource Information (URI))
- ชื่อของบริการภายในสิ่งแวดล้อมนั้น (Name)

ส่วนข้อมูลวิธีในการเข้าถึงจะประกอบไปด้วย

- ชนิดของเกณฑ์วิธีการเรียกใช้งาน (Access Protocol Type) เช่นเอ็ชทีทีพี หรืออาร์พีซี (Remote Procedure Call (RPC))
- ชนิดของที่อยู่และที่อยู่ (Address Type and Address) เช่นชนิดของที่อยู่เป็นตัวอักษร หรือมีที่อยู่เป็นอินเทอร์เน็ตโปรโตคอล (Internet Protocol (IP))
- คำสั่งในการใช้งาน (Access Instruction)

ตามตัวอย่างดังนี้

Naming System Type = DCE Directory Service

Context Reference = "groucho.dstc.edu.au"

Name = "/.:/example/dce_service"

Access Protocol Type = "DCE TCP RPOC Access Protocol Type"

Address Type = String

Address = "ncacn_ip_tcp:130.102.176.33[1049]"

Access Instructions = NULL

แม้ว่าข้อมูลที่เพิ่มขึ้นเหล่านี้สามารถบ่งบอกถึงสถาปัตยกรรมและการเรียกใช้งานภายใต้สถาปัตยกรรมนั้นได้ แต่ตัวแบบนี้ก็เป็นเพียงบริการที่ทำหน้าที่ค้นหาที่อยู่ของบริการโดยใช้ชื่อของบริการเท่านั้น รวมถึงวิธีในการเข้าไปใช้ข้อมูลนี้ยังคงต้องผ่านสถาปัตยกรรมใดสถาปัตยกรรมหนึ่งที่พัฒนาขึ้นเพื่อรองรับบริการค้นหานี้

### 2.1.3 A Component Search Engine Model on Internet [8]

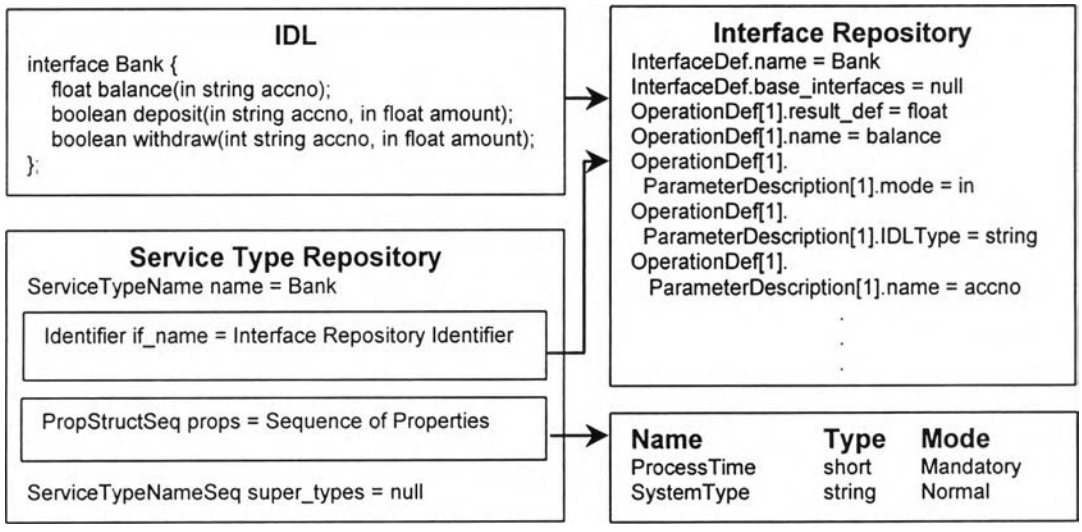
ในงานวิจัยนี้ได้นำเสนอระบบค้นหาคอมโพเนนท์ โดยอาศัยหลักการจับคู่ชื่อ (Name Matching) และการจับคู่ความหมาย (Semantic Matching) ในการจับคู่ชื่อ จะอาศัยสัญนิยมการตั้งชื่อ (Naming Convention) เป็นหลัก ทำให้การพิจารณาชื่อที่เหมือนกันทำได้ง่าย ส่วนในการจับคู่ความหมายนั้นทำได้โดยระบบการเขียนเทอมใหม่ (Term Rewriting System (TRS)) ซึ่งคำอธิบายคอมโพเนนท์และการสอบถามข้อมูลจะอยู่ในรูปข้อกำหนดรูปนัยเชิงพีชคณิต (Algebraic Formal Specification) โดยที่หากคอมโพเนนท์มีความสอดคล้องกับข้อมูลที่สอบถาม ระบบค้นหาจะสามารถเขียนข้อกำหนดรูปนัยของการสอบถามได้ใหม่ (Rewrite) โดยอาศัยข้อกำหนดของคอมโพเนนท์ได้ อย่างไรก็ดี แม้ว่างานวิจัยนี้สร้างอยู่บนพื้นฐานของคอร์บาเทรดเดอร์ แต่คำอธิบายและการสอบถามข้อมูลอยู่ในรูปของข้อกำหนดรูปนัย จึงทำให้ยากที่จะเข้าใจโดยระบบงานอื่น และยากในการนำไปใช้ในสถาปัตยกรรมอื่นที่ต่างออกไป

## 2.2 ทฤษฎีที่เกี่ยวข้อง

### 2.2.1 บริการเทรดเดอร์ (OMG CORBA Trading Object Service) [1][2]

บริการเทรดเดอร์ของคอร์บา เป็นบริการที่สนับสนุนการทำงานในระบบกระจายที่กำหนดโดยโอเอ็มจี ซึ่งให้บริการที่เปรียบเสมือนสมุดหน้าเหลือง ซึ่งอาศัยคุณสมบัติของบริการในการค้นหาบริการ องค์ประกอบหลักของบริการเทรดเดอร์มีดังนี้คือ

1. คลังชนิดของบริการ (Service Type Repository) ที่อาจอยู่ในรูปของแฟ้มข้อมูล หรือฐานข้อมูล เพื่อเก็บข้อมูลที่อธิบายชนิดของบริการดังนี้คือ (รูปที่ 2.1)
   1.1 ชื่อชนิดของบริการ (Service Type Name)
   1.2 ส่วนชี้ตำแหน่งในคลังส่วนต่อประสาน ( Interface Repository Identifier)
   1.3 ชื่อส่วนต่อประสานที่ได้รับการสืบทอดมา (Base Service Type Name)
   1.4 ชื่อคุณสมบัติของชนิดของบริการ (Property Name) ที่แสดงอยู่ในรูปของตัวอักษร
   1.5 ชนิดของคุณสมบัติ (Type of Property) กำหนดชนิดของคุณสมบัติที่เป็นได้ ทั้งชนิดที่กำหนดเป็นพื้นฐานไว้แล้วในคำอธิบายส่วนต่อประสานที่เรียกว่าไอดีแอล หรือกำหนดขึ้นเองภายหลัง (User-defined)
   1.6 แบบของคุณสมบัติ ( Property Mode) ซึ่งแบ่งเป็น
      - แบบปกติสามารถมีหรือไม่ก็ได้ (Normal)
      - แบบอ่านได้อย่างเดียวโดยไม่มีการแก้ไข (Readonly)
      - แบบที่ต้องระบุ (Mandatory)
      - แบบที่ต้องระบุและอ่านได้อย่างเดียวโดยไม่มีการแก้ไข (Mandatory Readonly)




**รูปที่** 2.1 ตัวอย่างข้อมูลภายในคลังชนิดของบริการและคลังส่วนต่อประสาน

2. ส่วนจัดเก็บข้อเสนอบริการ (Offer List) ซึ่งเป็นส่วนจัดการข้อมูลสามชนิดคือ

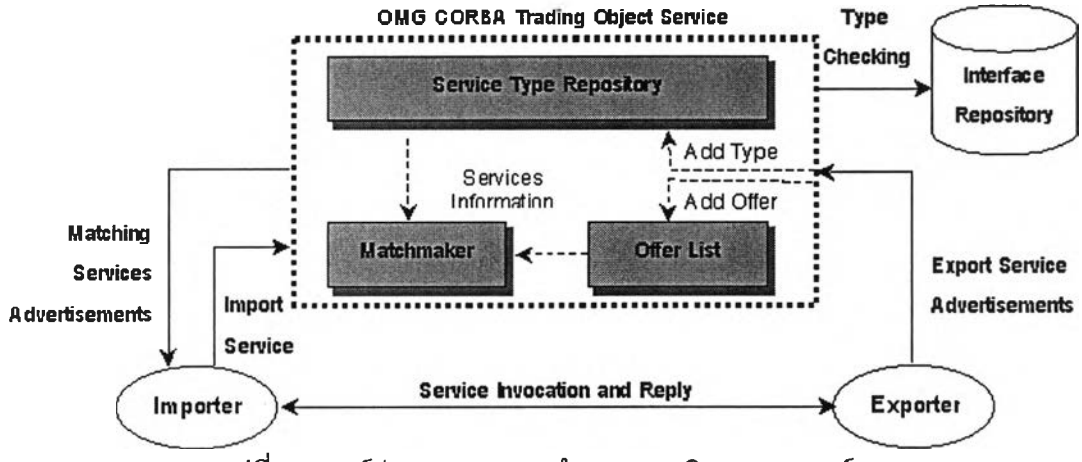
   2.1 ข้อมูลในการอ้างถึงวัตถุ (Object Reference) ที่แสดงที่อยู่ของข้อเสนอบริการ

   2.2 ชื่อชนิดของบริการ (Service Type Name) เช่นเดียวกับในคลังชนิดของบริการ

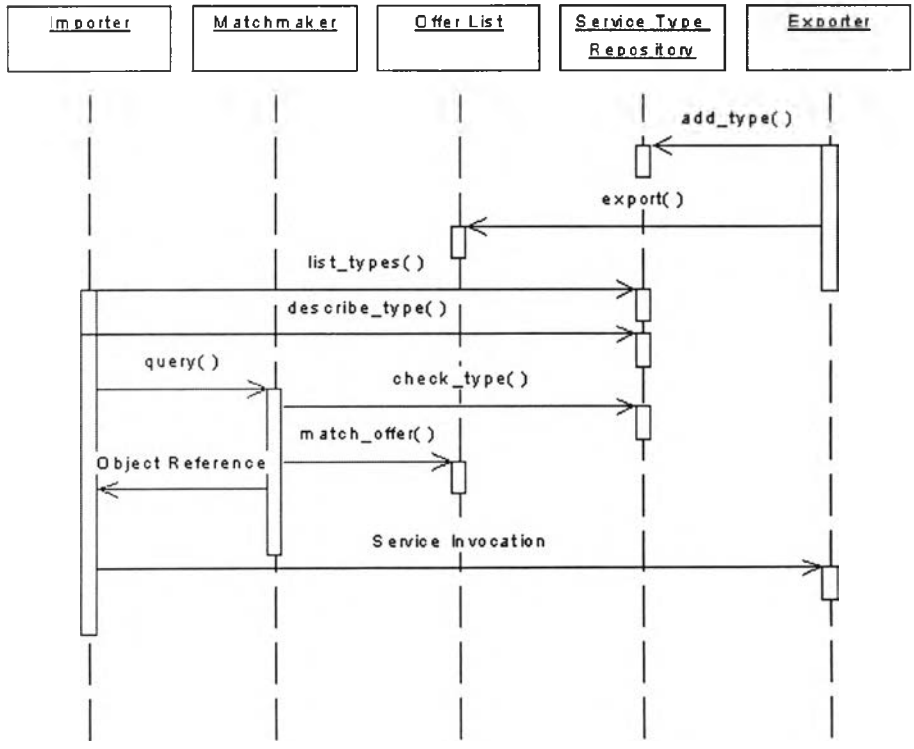
   2.3 รายการคู่ลำดับของชื่อคุณสมบัติและค่าคุณสมบัติ (Properties of the Service)

3. ส่วนคัดเลือกบริการ (Matchmaker) ทำการค้นหาบริการที่ตรงกับความต้องการของผู้รับบริการ

เมื่อผู้รับบริการทำการร้องขอบริการ บริการเทรดเดอร์จะคัดเลือกบริการที่ตรงกับความต้องการมากที่สุดตามที่มีผู้ให้บริการมาโฆษณาไว้เพื่อส่งกลับไปให้ผู้รับบริการ ดังแสดงในรูปที่ 2.2 และรูปที่ 2.3




รูปที่ 2.2 องค์ประกอบและการทำงานของบริการเทรดเดอร์



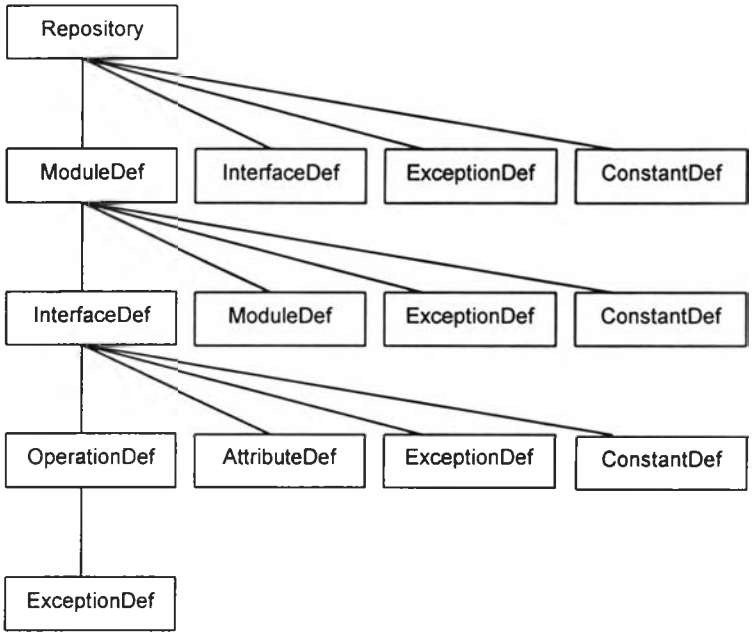

รูปที่ 2.3 แผนผังแสดงลำดับเหตุการณ์เมื่อมีการเรียกใช้บริการเทรดเดอร์

จากรูปข้างต้น ข้อมูลคำอธิบายชนิดของบริการจะสามารถสืบค้นได้จากภายในบริการเทรดเดอร์ คือได้จากคลังชนิดของบริการ และจากบริการภายนอกคือได้จากคลังส่วนต่อประสานซึ่งจะจัดเก็บข้อมูลต่างๆ ที่เกี่ยวข้องกับส่วนต่อประสานไว้ดังจะกล่าวในหัวข้อถัดไป

2.2.2 **คลังส่วนต่อประสาน** (Interface Repository) [1]

คลังส่วนต่อประสานทำหน้าที่ในการจัดเก็บนิยามส่วนต่อประสานไว้ โดยข้อมูลเหล่านี้ถือเป็นข้อมูลที่สำคัญสำหรับการอธิบายคุณสมบัติ และการเรียกใช้บริการ ข้อมูลหลักที่จัดเก็บอยู่ในคลังจัดเก็บส่วนต่อประสานที่จะนำมาใช้ในการสร้างคำอธิบายชนิดของบริการให้เป็นเอกสารเอ็กซ์เอ็มแอลดังรูปที่ 2.4 มีดังนี้คือ

1. Repository เป็นส่วนจำเพาะ (Module) ในระดับบนสุดเพื่อใช้เป็นตัวบรรจุนิยามส่วนจำเพาะ (ModuleDef) นิยามส่วนต่อประสาน (InterfaceDef) นิยามค่าคงที่ (ConstantDef) นิยามข้อยกเว้น (ExceptionDef)
2. ModuleDef เป็นตัวบรรจุในเชิงตรรกะของนิยามส่วนต่อประสาน นอกจากนั้นยังเป็นตัวบรรจุของนิยามค่าคงที่ นิยามข้อยกเว้น และส่วนจำเพาะอื่นๆ
3. InterfaceDef คือนิยามส่วนต่อประสาน ซึ่งเป็นนิยามหลักสำหรับการอธิบายส่วนต่อประสานของชนิดของบริการหนึ่งๆ โดยนิยามส่วนต่อประสานนี้จะบรรจุรายละเอียดสำหรับส่วนต่อประสานคือ นิยามค่าคงที่ นิยามลักษณะประจำ (AttributeDef) นิยามข้อยกเว้น และนิยามตัวกระทำการ (OperationDef)
4. ConstantDef คือนิยามค่าคงที่ของส่วนต่อประสานที่สามารถนำข้อมูลชนิด และค่าของค่าคงที่มาจากนิยามลักษณะประจำนี้
5. AttributeDef คือนิยามลักษณะประจำของส่วนต่อประสานโดยจะสามารถเข้าถึงข้อมูลของลักษณะประจำ (Attribute) เช่นภาวะ และชนิดของลักษณะประจำ
6. OperationDef คือนิยามตัวกระทำการของส่วนต่อประสาน เป็นตัวบรรจุรายการพารามิเตอร์ ข้อยกเว้นที่อาจเกิดขึ้นจากการกระทำการ และคอนเท็กซ์ของตัวกระทำการ
7. ExceptionDef คือนิยามข้อยกเว้นที่อาจเกิดขึ้นจากการกระทำการ โดยมีรายละเอียดของสมาชิกของข้อยกเว้น และชนิดของสมาชิก
8. ParameterDescription คือคำอธิบายพารามิเตอร์สำหรับตัวกระทำการที่ประกอบไปด้วยชื่อ ชนิด และภาวะของพารามิเตอร์




**รูปที่** 2.4 ความสัมพันธ์ในลักษณะการบรรจุ (Containment Relationship) ของข้อมูลในคลังส่วนต่อประสาน

2.2.3 **เอ็กซ์เอ็มแอล** (Extensible Markup Language (XML)) [6]

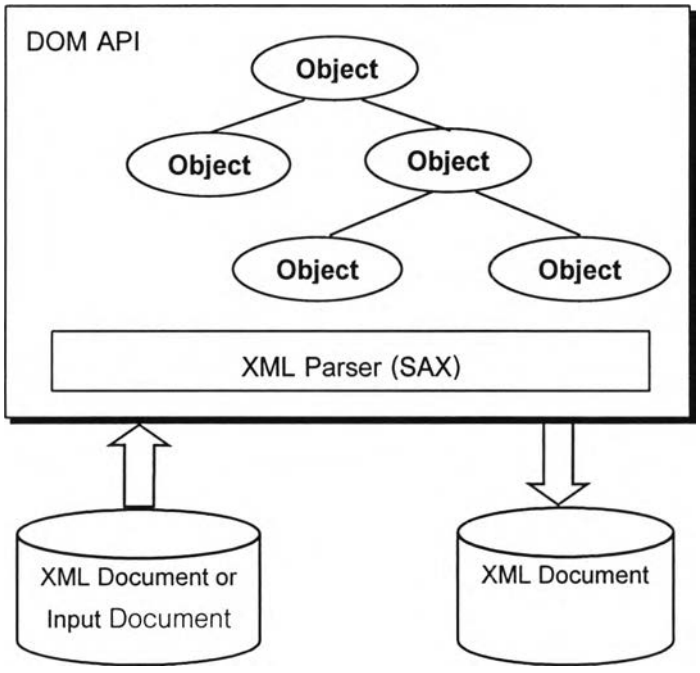
เอ็กซ์เอ็มแอลเป็นภาษาที่ถูกออกแบบเพื่อใช้อธิบายข้อมูลของวัตถุ และพฤติกรรม (Behavior) ของโปรแกรม โดยเป็นภาษาที่มีพื้นฐานจากเอสจีเอ็มแอล (Standard Generalized Markup Language (SGML)) แต่ได้ทำการลดทอนความซับซ้อนลง วัตถุประสงค์หลักของเอ็กซ์เอ็มแอลนั้นเพื่อให้เป็นภาษาที่ใช้ง่าย สามารถประมวลผลได้ทั้งโดยโปรแกรมประยุกต์ และมนุษย์

เอ็กซ์เอ็มแอลจะประกอบด้วยส่วนประกอบสองส่วนคือ คำอธิบายชนิดของเอกสารที่เรียกว่าดีทีดี (Document Type Description (DTD)) ที่ทำหน้าที่กำหนดไวยากรณ์ของเอกสาร ส่วนที่สองคือตัวเอกสารเอ็กซ์เอ็มแอลซึ่งเป็นโครงสร้างทางตรรกะ (Logical Structure) โดยอธิบายคุณลักษณะต่างๆ ที่สอดคล้องกับดีทีดีในรูปของโครงสร้างลำดับชั้น (Hierarchy) จากการที่เอ็กซ์เอ็มแอลมีการแบ่งส่วนของดีทีดี และโครงสร้างทางตรรกะที่แยกออกจากส่วนของการแสดงผล ทำให้ได้เอกสารที่มีเค้าร่าง (Schema) เป็นโครงสร้าง (Structure) ที่ง่ายต่อการนำข้อมูลในเอกสารไปค้นหา และนำกลับไปใช้ใหม่ได้หลายๆ ครั้ง ภายใต้สื่อต่างชนิดกัน

เอ็กซ์เอ็มแอลนั้นเป็นเอกสารที่มีความยืดหยุ่นสำหรับงานประยุกต์ที่มีพื้นฐานบนเว็บ เอ็กซ์เอ็มแอลมีรูปแบบการนำเสนอในรูปแบบข้อความ (Text) จึงไม่ขึ้นกับระบบปฏิบัติการ และสถาปัตยกรรมของคอมพิวเตอร์ และมีแนวโน้มที่จะเป็นมาตรฐานใหม่ในระบบเปิด ในขณะเดียวกันแท็ก (Tag) ของเอกสารที่เป็นข้อความนั้นก็สามารถแสดงคำอธิบายเชิงความหมาย ทำให้เอ็กซ์เอ็มแอลมีความยืดหยุ่นในการเขียนเมตาดาต้า เพื่อการจัดการข้อมูล

2.2.4 ดีโอเอ็ม (Document Object Model (DOM)) [9][10]

ดีโอเอ็ม คือข้อกำหนดของส่วนต่อประสานในการเขียนโปรแกรมประยุกต์ หรือเอพีไอ (Application Programming Interface (API)) เพื่อการจัดการข้อมูลเอกสารเอ็กซ์เอ็มแอลให้อยู่ในรูปของวัตถุที่สามารถเขียนชุดคำสั่งเพื่อเข้าถึง และแก้ไขเนื้อความภายในได้ ข้อกำหนดของดีโอเอ็มระดับหนึ่ง (DOM Level 1) ประกอบไปด้วยส่วนต่อประสานสำคัญสองส่วนคือ ส่วนแรกใช้ในการจัดการเอกสารเอ็กซ์เอ็มแอลในรูปของวัตถุ และส่วนที่สองคือส่วนต่อประสานที่ทำงานร่วมกับส่วนแรก เพื่อเพิ่มความสามารถในการ จัดการกับแท็กของเอกสารเอ็ชทีเอ็มแอล ขั้นตอนการสร้างเอกสารเอ็กซ์เอ็มแอลโดยใช้ดีโอเอ็มอาจแสดงได้ดังรูปที่ 2.5




**รูปที่** 2.5 การสร้างเอกสารเอ็กซ์เอ็มแอลโดยใช้ดีโอเอ็ม

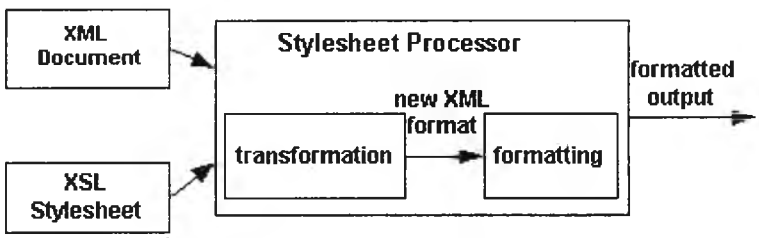
การจัดการเอกสารโดยใช้ดีโอเอ็มอาจเริ่มจากการอ่านเอกสารเอ็กซ์เอ็มแอลที่มีอยู่แล้วเพื่อนำมาแก้ไข เมื่อมีการอ่านเอกสารเอ็กซ์เอ็มแอลที่มีอยู่แล้วเข้ามา เอกสารนั้นจะผ่านการตรวจสอบไวยากรณ์โดยเอพีไอที่เรียกว่าเอสเอเอ็กซ์ (Simple API for XML (SAX)) จากนั้นจึงมีการสร้างวัตถุที่มีเนื้อความภายในเช่นเดียวกับเอกสารขึ้นมา เมื่อผ่านการแก้ไขหรือเพิ่มเติมเนื้อความภายในแล้วเราสามารถนำเนื้อความภายในวัตถุมาสร้างเอกสารเอ็กซ์เอ็มแอล หรือเราสามารถใช้ดีโอเอ็มนี้ไปใช้ในการสร้างวัตถุเปล่าขึ้นมาเพื่อเตรียมการสร้างเอกสารเอ็กซ์เอ็มแอลได้อีกด้วย การใช้ดีโอเอ็มจะช่วยให้การจัดการ และการเขียนเอกสารเอ็กซ์เอ็มแอลที่อยู่ในรูปวัตถุทำได้ง่ายขึ้น โดยสามารถนำส่วนต่อประสานโปรแกรมประยุกต์เหล่านี้มาพัฒนาสร้างโปรแกรมประยุกต์ให้สามารถนำข้อมูลในรูปของวัตถุมาแสดงผลในรูปของส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphical User Interface (GUI)) หรือนำเสนอผ่านสื่อในลักษณะอื่นได้

ในปัจจุบันได้มีการพัฒนาดีโอเอ็มภายใต้ข้อกำหนดของดับเบิลยูทรีซี (World Wide Web Consortium (W3C)) ออกมามากมายเช่น จาวาโปรเจคเอ็กซ์ (Java Project X) จากซันไมโครซิสเต็ม (Sun

Microsystems) ออราเคิลเอ็กซ์เอ็มแอลพาร์สเซอร์ (Oracle XML Parser) จาก ออราเคิล (Oracle) และเอ็มเอสไอดีไฟว์ดีโอเอ็ม (MSIE5DOM) จากไมโครซอฟท์ (Microsoft) โดยการพัฒนาเหล่านี้ยังได้เพิ่มเติมส่วนจัดการเอกสารที่เป็นประโยชน์ เช่นการสร้างแฟ้มข้อมูลเอกสารเอ็กซ์เอ็มแอล ทำให้สามารถนำการพัฒนาเหล่านี้ไปใช้ในการจัดการเอกสารเอ็กซ์เอ็มแอลได้โดยตรง

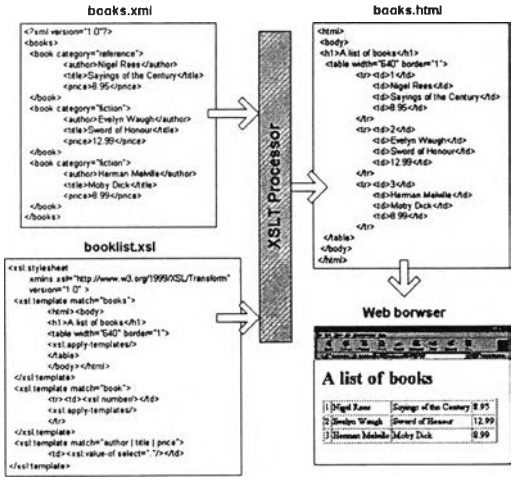
2.2.5 **เอ็กซ์เอสแอล** (Extensible Stylesheet Language (XSL)) **และ เอ็กซ์เอสแอลที** (XSL Transformations (XSLT)) [11][12]

เอ็กซ์เอสแอลเป็นเทคโนโลยีสำหรับการเปลี่ยนรูปแบบ (Transformation) ของเอกสารเอ็กซ์เอ็มแอล และจัดรูปแบบใหม่ (Formatting) ให้กับเอกสาร โดยเอ็กซ์เอสแอลจะเป็นตัวกำหนดว่าข้อมูลโครงสร้างใดในเอกสารเดิมจะถูกนำไปเปลี่ยนให้อยู่ในรูปแบบใดในอีกเอกสารหนึ่ง ในการแปลงและการจัดรูปแบบนี้เอ็กซ์เอสแอล และเอ็กซ์เอ็มแอลจะถูกนำไปประมวลผลผ่านตัวประมวลผลสไตล์ชีต (Stylesheet Processor) เพื่อแปลงเป็นเอกสารเอ็กซ์เอ็มแอลอีกรูปแบบหนึ่ง และจัดรูปแบบใหม่เพื่อให้สามารถนำไปแสดงผลภายใต้สื่ออื่นๆ ต่อไป ดังในรูปที่ 2.6




**รูปที่** 2.6 รูปแบบการทำงานของเอ็กซ์เอสแอล

สำหรับเอ็กซ์เอสแอลทีนั้นเป็นข้อกำหนดสำหรับการแปลงเอกสารเอ็กซ์เอ็มแอลรูปแบบหนึ่งไปเป็นเอกสารเอ็กซ์เอ็มแอลอีกรูปแบบหนึ่ง ซึ่งเอ็กซ์เอสแอลทีนี้ถูกออกแบบเพื่อใช้เป็นส่วนหนึ่งของเอ็กซ์เอสแอล ดังในรูปที่ 2.6 หรือสามารถนำไปใช้แยกจากเอ็กซ์เอสแอลเพื่อใช้ในการแปลงเอกสารเอ็กซ์เอ็มแอลเพียงอย่างเดียว ดังในรูปที่ 2.7




**รูปที่** 2.7 ตัวอย่างการทำงานของเอ็กซ์เอสแอลที