

การส่งออกและแสดงข้อมูลของระบบสารสนเทศโดยใช้เอกซ์เอ็มแอล

Exporting and Displaying Management Information System Data Using XML

กานดา รุณนะพงศา สายแก้ว^{1*}, พัชรพร ด้านธรรณ²,
อันธิมา เปรมเกิด³, รุจชัย อึ้งอารุณยะวี⁴

^{1,4} ภาควิชาวิศวกรรมคอมพิวเตอร์ ^{2,3} ศูนย์คอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น ถนนมิตรภาพ
อำเภอเมือง จังหวัดขอนแก่น 40002

บทคัดย่อ

ในปัจจุบันองค์กรต่างๆ มักจะมีระบบสารสนเทศเพื่อจัดการและบริหาร โดยที่ข้อมูลนั้นเก็บในฐานข้อมูลและมีการพัฒนาเว็บแอปพลิเคชันเพื่อดึงข้อมูลนั้นออกมาใช้ภายในองค์กร ในขณะเดียวกันองค์กรต่างๆ จำเป็นต้องมีการแลกเปลี่ยนข้อมูลระหว่างกัน โดยที่ไม่ต้องการให้องค์กรอื่นเข้ามาจัดการและใช้ฐานข้อมูลของตนเอง ดังนั้นระบบสารสนเทศขององค์กรจำเป็นต้องมีวิธีการแลกเปลี่ยนข้อมูลโดยที่ข้อมูลนั้นอยู่ในรูปแบบมาตรฐานที่เป็นที่ยอมรับและใช้กันทั่วไป มาตรฐานดังกล่าวก็คือมาตรฐานภาษาเอกซ์เอ็มแอล

บทความนี้นำเสนอวิธีการส่งออกข้อมูลจากฐานข้อมูลให้อยู่ในรูปแบบภาษาเอกซ์เอ็มแอลและแสดงข้อมูลโดยใช้ภาษาเอกซ์เอสแอล การแสดงข้อมูลโดยใช้ภาษาเอกซ์เอสแอลและตัวประมวลผลเอกซ์เอสแอลซึ่งเป็นส่วนหนึ่งของเว็บเบราว์เซอร์อยู่แล้วนั้นทำให้ผู้ใช้สามารถเห็นข้อมูลในรูปแบบที่อ่านได้ง่าย วิธีการนี้สามารถนำไปประยุกต์ใช้กับระบบสารสนเทศโดยทั่วไปที่พัฒนาโดยใช้ภาษาพีเอชพีและฐานข้อมูลมายเอสคิวแอลซึ่งเป็นโอเพนซอร์ส นอกจากนี้ยังนำเสนอผลการทดลองของวิธีการเขียนเอกสารเอกซ์เอ็มแอลด้วยวิธีใหม่ที่มีประสิทธิภาพมากกว่าวิธีที่มักนิยมใช้กัน

คำสำคัญ: ระบบสารสนเทศ การแลกเปลี่ยนข้อมูล การแสดงข้อมูล เอกซ์เอ็มแอล เอกซ์เอสแอล การแปลงข้อมูล

บทนำ

ในปัจจุบันองค์กรต่างๆ ไม่ว่าจะเป็นองค์กรภาครัฐหรือภาคเอกชน มักจะมีระบบสารสนเทศเพื่อจัดการและบริหารองค์กร [1] โดยที่ข้อมูลนั้นเก็บในฐานข้อมูลและมีการพัฒนาเว็บแอปพลิเคชันเพื่อดึงข้อมูลนั้นออกมาใช้ภายในองค์กร รูปแบบการพัฒนาเว็บแอปพลิเคชันที่เป็นที่นิยมคือใช้ภาษาพี

เอชพี (PHP) ซึ่งย่อมาจาก [2] ในการพัฒนาโปรแกรมและใช้มายเอสคิวแอล (MySQL) [3] เป็นระบบฐานข้อมูลเพื่อจัดการข้อมูลขององค์กร เหตุผลที่หลายหน่วยงานเลือกใช้ PHP และ MySQL เนื่องจากทั้ง PHP และ MySQL สามารถนำมาใช้ได้ฟรี เป็นโอเพนซอร์สที่ใช้ง่ายและเรียนรู้ได้อย่างรวดเร็ว อีกทั้งมีแหล่งข้อมูลให้ศึกษาอย่างหลากหลายและมากมายบนอินเทอร์เน็ต

ในขณะเดียวกันองค์กรต่างๆ จำเป็นต้องมีการแลกเปลี่ยน

* E-mail: krunapon@kku.ac.th

ข้อมูลระหว่างกัน โดยที่ไม่ต้องการให้องค์กรอื่นเข้ามาจัดการ และใช้ฐานข้อมูลของตนเอง ดังนั้นระบบสารสนเทศขององค์กรจำเป็นต้องมีวิธีการส่งออกข้อมูลโดยที่ข้อมูลนั้นอยู่ในรูปแบบมาตรฐานที่เป็นที่ยอมรับและใช้กันทั่วไปซึ่งก็คือภาษา เอกซ์เอ็มแอล

เอกซ์เอ็มแอล (XML ซึ่งย่อมาจาก Extensible Markup Language) [4] คือภาษามาร์กอัปซึ่งสามารถใช้งานโดยทั่วไป ด้วยคุณสมบัติที่ผู้ใช้สามารถกำหนดความหมายของข้อมูลด้วยตนเอง XML เป็นภาษาพื้นฐานของหลายภาษาที่มีการใช้เฉพาะเจาะจงในแต่ละวัตถุประสงค์ที่แตกต่างกันออกไป ยกตัวอย่างเช่น RSS (Really Simple Syndication) ซึ่งเป็นรูปแบบข้อมูลล่าสุดของเว็บไซต์ต่างๆ [5] MathML (Mathematical Markup Language) ซึ่งเป็นรูปแบบข้อมูลทางคณิตศาสตร์ [6] MusicXML ซึ่งเป็นรูปแบบข้อมูลทางดนตรี [7] เป็นต้น

ปัจจุบัน XML ใด? เขามีบทบาทและเป็นมาตรฐานในการแลกเปลี่ยนข้อมูล เนื่องจากมีความสามารถในการอธิบายความหมายของข้อมูลและมีความยืดหยุ่นในการใช้งาน โดย XML เป็นข้อความตัวอักษร (text) ซึ่งเครื่องคอมพิวเตอร์ไม่ว่าจะเป็นแพลตฟอร์มใดหรือมีระบบในการใช้งานที่ต่างก็ก็สามารถอ่านข้อความตัวอักษรได้ XML เป็นเทคโนโลยีที่ไม่ขึ้นกับแพลตฟอร์มใดๆ เช่น เครื่องคอมพิวเตอร์ที่ใช้ภาษา Visual Basic บนระบบปฏิบัติการวินโดวส์ของ ไมโครซอฟต์ (Windows Operating System) สามารถติดต่อแลกเปลี่ยนข้อมูลเอกซ์เอ็มแอลกับเครื่องคอมพิวเตอร์ที่ใช้ภาษาจาวา (Java) บนระบบปฏิบัติการลินุกซ์ (Linux System) ได้ จากจุดเด่นของ XML นี้เองจึงเป็นโอกาสที่จะนำไปใช้ประโยชน์ในเชิงพาณิชย์ เช่นในการพัฒนาเว็บเซอร์วิส (Web Service) ซึ่งเป็นเทคโนโลยีที่ทำให้โปรแกรมที่พัฒนาโดยภาษาและบนแพลตฟอร์มที่ต่างก็สามารถติดต่อและทำงานร่วมกันได้โดยอัตโนมัติ นอกจากนี้ยังทำให้โปรแกรมสามารถถูกเรียกใช้ได้จากอุปกรณ์อิเล็กทรอนิกส์ใดๆ ที่ติดต่อกันผ่านระบบอินเทอร์เน็ต

บทความนี้จะนำเสนอวิธีการส่งออกข้อมูลจากฐานข้อมูลให้อยู่ในรูปแบบภาษา XML และแสดงข้อมูลโดยใช้ภาษาเอกซ์เอ็มแอล (XSL ซึ่งย่อมาจาก Extensible Stylesheet Language) [8] ซึ่งเป็นภาษาที่อยู่ในรูปแบบภาษา XML ภาษาหนึ่ง XSL ถูกออกแบบมาเพื่อใช้ในการระบุรูปแบบการ

แสดงข้อมูลของข้อมูล XML โดยที่จะอาศัยตัวประมวลผล เอกซ์เอสแอล (XSLT ซึ่งย่อมาจาก Extensible Stylesheet Language Transformation) [9] ซอฟต์แวร์ XSLT จะรับ อินพุตไฟล์ 2 ไฟล์ซึ่งได้แก่ไฟล์ XML ที่มีเนื้อหาของข้อมูลที่จะแสดงผล อีกไฟล์หนึ่งคือไฟล์ XSL ที่มีคำสั่งเพื่อระบุวิธีการแสดงข้อมูล ส่วนเอาต์พุตของไฟล์นั้นจะเป็นไฟล์เท็กซ์โดยทั่วไป หรือเป็นไฟล์เอกซ์เอ็มแอล หรือเป็นไฟล์เอชทีเอ็มแอล (HTML ซึ่งย่อมาจาก HyperText Markup Language) [10] ในที่นี้เราจะแปลงข้อมูลให้อยู่ในรูปแบบ HTML ซึ่งสามารถเปิดดูโดยใช้เว็บเบราว์เซอร์ได้ XSLT เป็นส่วนหนึ่งของเว็บเบราว์เซอร์อยู่แล้วนั้นทำให้ผู้ใช้สามารถเห็นข้อมูลในรูปแบบที่อ่านได้ง่าย

บทความนี้จะนำเสนอการดึงข้อมูลจากฐานข้อมูลของระบบสารสนเทศ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น (<http://www.en.kku.ac.th/mis>) ในส่วนของบทความที่ดีพิมพ์ของบุคลากรในคณะ ซึ่งข้อมูลนี้จะถูกส่งออกจากฐานข้อมูลของระบบสารสนเทศเพื่ออาจจะนำไปแสดงผลข้อมูลที่เว็บเพจของบุคลากรโดยอัตโนมัติ หากไม่มีวิธีการที่นำเสนอนี้ บุคลากรของคณะก็จะต้องกรอกข้อมูลการตีพิมพ์บทความของตนเองทั้งที่ระบบสารสนเทศของคณะ และที่เว็บเพจของตนเอง แต่อย่างไรก็ตาม วิธีการที่นำเสนอในบทความนี้สามารถนำไปประยุกต์ใช้กับระบบสารสนเทศโดยทั่วไปที่พัฒนาโดยใช้ภาษาพีเอชพีและฐานข้อมูลมายเอสคิวแอลซึ่งเป็นโอเพนซอร์ส นอกจากนี้บทความนี้จะนำเสนอการเขียนเอกสาร XML อย่างมีประสิทธิภาพโดยใช้โค้ดภาษา PHP

โครงสร้างของบทความมีดังต่อไปนี้ หัวข้อที่ 2 อธิบายงานที่เกี่ยวข้องและความรู้พื้นฐาน หัวข้อที่ 3 อธิบายภาพรวมของระบบสารสนเทศ หัวข้อที่ 4 อธิบายระบบย่อยที่ใช้ในการส่งออกและแสดงข้อมูล XML และสรุปในหัวข้อที่ 5

งานที่เกี่ยวข้องและความรู้พื้นฐาน

งานที่เกี่ยวข้อง

การส่งออกข้อมูลระหว่างระบบโดยข้อมูลนั้นดึงออกจากฐานข้อมูลแบบสัมพันธ์ (Relational database) นั้นสามารถทำได้หลายวิธี อาทิเช่น

1) การส่งออกข้อมูลในรูปแบบไฟล์ SQL ซึ่งมีคำสั่งในการสร้างตาราง และคำสั่งในการใส่ข้อมูลเข้าไปในตาราง ข้อ

ด้อยของวิธีนี้คือทุกครั้งที่มีข้อมูลมีการเปลี่ยนแปลง ผู้ส่งออกข้อมูลจะต้องทำไฟล์ SQL ใหม่เพื่อให้มีการอัปเดตข้อมูลที่ล่าสุดที่อยู่ในตาราง อีกทั้งหากมีผู้ขอข้อมูลที่มีความต้องการหลากหลาย ก็จะต้องมีไฟล์ SQL หลายๆไฟล์ตามแต่ความต้องการของผู้ขอข้อมูล ข้อด้อยของวิธีนี้อีกประการหนึ่งคือการเปิดเผยโครงสร้างของฐานข้อมูลว่าระบบสารสนเทศมีการเก็บข้อมูลอะไรบ้าง หากต้องการจะเปิดเผยข้อมูลเพียงบางคอลัมน์หรือบางแถว ก็จะต้องมีการใช้คำสั่ง SQL ที่สลับซับซ้อนมากยิ่งขึ้น

2) การส่งออกข้อมูลในรูปแบบเท็กซ์ที่มีรูปแบบเฉพาะ ข้อมูลในรูปแบบเท็กซ์นี้มักจะประกอบด้วยหลายๆแถวโดยที่แต่ละแถวมีแท็บคั่นกลางระหว่างข้อมูลของแต่ละคอลัมน์ ข้อด้อยของวิธีนี้คือจะต้องมีการพัฒนาโปรแกรมเพื่อเขียนข้อมูลและอ่านข้อมูลที่อยู่ในรูปแบบเฉพาะ หากรูปแบบข้อมูลเปลี่ยน ผู้ใช้และผู้รับข้อมูลจะต้องแก้ไขโปรแกรม อีกทั้งเนื่องจากข้อมูลนั้นเป็นรูปแบบเฉพาะ จึงอาจทำให้ยากต่อการพัฒนาโปรแกรมสำหรับคนทั่วไปเพื่อค้นหาข้อมูลในไฟล์ลักษณะนี้ ตัวอย่างของรูปแบบนี้ได้แสดงไว้ในรูปที่ 1

ID	Firstname	Lastname	GPA	Year
4830401234	Manee	Jaidee	3.4	2
4830405678	Mana	Deejai	2.3	3

รูปที่ 1 ตัวอย่างของข้อมูลในรูปแบบเท็กซ์ที่มีรูปแบบเฉพาะ

3) การส่งออกข้อมูลในรูปแบบ HTML ปกติแล้วเว็บแอปพลิเคชันมักจะมีส่วนของการอ่านข้อมูลจากฐานข้อมูลแล้วแสดงใน HTML เพื่อให้ผู้อ่านข้อมูลสามารถเห็นได้จากการใช้เว็บเบราว์เซอร์ ถึงแม้ว่าวิธีนี้จะเหมาะสมในกรณีที่หากข้อมูลมีการอัปเดต โปรแกรมไม่จำเป็นต้องมีการแก้ไขตาม อย่างไรก็ตาม การใช้วิธีนี้ผู้อ่านข้อมูลจะต้องใช้เมนูคัดลอก (copy) และแปะวาง (paste) เพื่อได้ข้อมูลนั้น ผู้ขอข้อมูลอาจจะเขียนโปรแกรมเพื่อดึงข้อมูลมาก็ได้ แต่โปรแกรมนั้นก็จะต้องมีความสลับซับซ้อนและไม่ยืดหยุ่นเนื่องจากแท็กใน HTML เป็นแท็กเพื่อการแสดงข้อมูลไม่ได้เป็นแท็กที่สื่อความหมายของข้อมูล

บทความนี้ใช้วิธีส่งออกข้อมูลโดยให้ข้อมูลอยู่ในรูปแบบ XML โดยการเขียนโปรแกรมที่เขียนข้อมูล XML ซึ่งโปรแกรมนี้นี้มีคำสั่งในการดึงข้อมูลจากฐานข้อมูลข้อดีของการใช้ XML คือเป็นมาตรฐานที่ยอมรับและใช้ได้ฟรี เป็นมาตรฐานที่ผู้ใช้สามารถอ่าน เขียน และแก้ไขได้ในทุกแพลตฟอร์มซึ่งทำให้มีปัญหาน้อยในการแลกเปลี่ยนข้อมูลระหว่างองค์กรที่อาจมีการพัฒนาระบบโดยใช้เครื่องมือและแพลตฟอร์มที่แตกต่างกัน นอกจากนี้การใช้ภาษา XML ก็มีข้อดีในแง่ที่ว่าแท็กใน XML สื่อความหมายข้อมูลในตัวมันเอง ทำให้การพัฒนา

โปรแกรมเพื่ออ่านและค้นหาข้อมูล XML สามารถทำได้โดยอัตโนมัติและทำได้ง่าย ดังจะเห็นได้จากเว็บไซต์ข่าวสารและเว็บบล็อกต่างๆ ในปัจจุบันมีการส่งออกข้อมูลในรูปแบบ RSS ซึ่งเป็นรูปแบบหนึ่งของ XML เพื่อให้ผู้ที่สมัครรับข้อมูลอัปเดตข่าวสารของเว็บไซต์สามารถรับรู้ข้อมูลอัปเดตล่าสุดได้รวดเร็วโดยไม่ต้องมาที่เว็บไซต์นั้น

ความรู้พื้นฐาน

1. ความรู้พื้นฐานเกี่ยวกับ XML

XML เป็นมาตรฐานภาษาซึ่งข้อความ XML มีคำอธิบายในตัวของมันเอง รูปแบบของ XML จะคล้ายคลึงกับ HTML ซึ่งผู้พัฒนาโปรแกรมเว็บแอปพลิเคชันรู้จักเป็นอย่างดี กล่าวคือมีแท็กและมีข้อความที่อยู่ระหว่างแท็ก แต่แท็กของ HTML จะทำหน้าที่กำหนดรูปแบบการแสดงผล ในขณะที่แท็กของ XML จะทำหน้าที่สื่อความหมายตามที่ผู้เขียน XML ต้องการ จากรูปที่ 1 จะเห็นว่าแท็กของเอกสาร HTML เป็นแท็กที่ใช้เพื่อกำหนดรูปแบบการแสดงผลของข้อมูล อย่างเช่น แท็ก <p> มีไว้เพื่อที่จะบอกว่าข้อความที่อยู่ระหว่างแท็กเปิด <p> และแท็กปิด </p> เป็นข้อความที่จัดกลุ่มอยู่ในย่อหน้าเดียวกัน (paragraph)

```
<html>
  <head><title>MyFriends</title></head>
  <body>
    <p>Manee Jaidee</p>
  </body>
</html>
```

รูปที่ 2 ตัวอย่างเอกสาร HTML

หากพิจารณาในรูปที่ 3 แล้วจะเห็นว่าในขณะที่แท็ก XML มีส่วนของข้อมูลเกี่ยวกับ <nation>, <name> และ <location> โดยผู้สร้างข้อมูลสามารถกำหนดชื่อใดๆของแท็กได้ตามความต้องการซึ่งเป็นข้อดีของการสื่อความหมาย

ของข้อมูล ทำให้เกิดความยืดหยุ่นในการกำหนดโครงสร้างของข้อมูลและสามารถจัดโครงสร้างให้มีความเหมาะสมกับแอปพลิเคชันที่ใช้ XML

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

รูปที่ 3 ตัวอย่างเอกสาร XML

เอกสาร XML ประกอบไปด้วยอิลิเมนต์ (element) ต่างๆ โดยที่การซ้อนกันของ อิลิเมนต์ จะบ่งบอกถึงความสัมพันธ์ระหว่าง อิลิเมนต์ จากรูปที่ 4 บรรทัดที่ 1 คือ การประกาศว่าเอกสารนี้เป็นเอกสาร XML โดยที่หากต้องการจะให้ข้อมูลนั้นแสดงข้อมูลที่เป็นภาษาไทย จะต้องมีการกำหนดค่าของแอตทริบิวต์ (attribute) ที่ชื่อ encoding ให้มีค่า tis-620. แอตทริบิวต์เป็นส่วนหนึ่งของเอกสาร XML ซึ่งเป็นส่วนย่อยที่สุด กล่าวคือจะไม่มีข้อมูลส่วนย่อยอื่นอยู่ภายใต้แอตทริบิวต์

บรรทัดที่ 2 คือ คำสั่งในการประมวลผล ในไฟล์นี้คำสั่งที่ใช้ในการประมวลผลระบุให้มีการใช้สไลด์ซีดีไฟล์ที่ชนิดของข้อมูลเป็นแท็กที่อยู่ในรูปแบบ XSL โดยที่ไฟล์นี้ชื่อว่า nation_th.xml

บรรทัดที่ 3 เป็นจุดเริ่มต้นของ อิลิเมนต์ ประเทศ ซึ่งมี

จุดสิ้นสุดที่บรรทัดที่ 8 ข้างใน อิลิเมนต์ ประเทศ มี 3 อิลิเมนต์ ซ่อนอยู่ข้างในได้แก่ อิลิเมนต์ ชื่อ, อิลิเมนต์ สถานที่, และ อิลิเมนต์ ภาษา จะสังเกตได้ว่า อิลิเมนต์ จะเริ่มต้นด้วยแท็กเปิด ซึ่งภายในแท็กเปิดสามารถที่จะมีแอตทริบิวต์ได้ ตัวอย่างเช่นในบรรทัดที่ 3 จะแสดงแท็กเปิดของอิลิเมนต์ ประเทศ โดยที่แท็กเปิดนี้มีแอตทริบิวต์ รหัส ที่มีค่า “ไทย”

บรรทัดที่ 4 จะแสดงส่วนของคอมเมนต์ (comment) ซึ่งจะมีรูปแบบที่ขึ้นต้นด้วย <!-- และลงท้ายด้วย --> ในไฟล์นี้จะมีอิลิเมนต์หลายแบบเช่น อิลิเมนต์แบบที่มีเฉพาะข้อความแท็กและอิลิเมนต์ที่มีอิลิเมนต์ย่อย อิลิเมนต์ที่มีเฉพาะข้อความแท็กอยู่ภายในอิลิเมนต์นั้น อาทิ เช่น อิลิเมนต์ ชื่อ ตัวอย่างของอิลิเมนต์ที่มีอิลิเมนต์ย่อยในเอกสารนี้คืออิลิเมนต์ ประเทศ

```

1 <?xml version="1.0" encoding="tis-620"?>
2 <?xml-stylesheet type="text/xsl" href="nation_th.xsl"?>
3 <ประเทศ รหัส="ไทย">
4   <!-- มักจะเรียกว่า ประเทศไทย -->
5   <ชื่อ>ไทยแลนด์</ชื่อ>
6   <สถานที่>เอเชียตะวันออกเฉียงใต้</สถานที่>
7   <ภาษา>ไทย</ภาษา>
8 </ประเทศ>

```

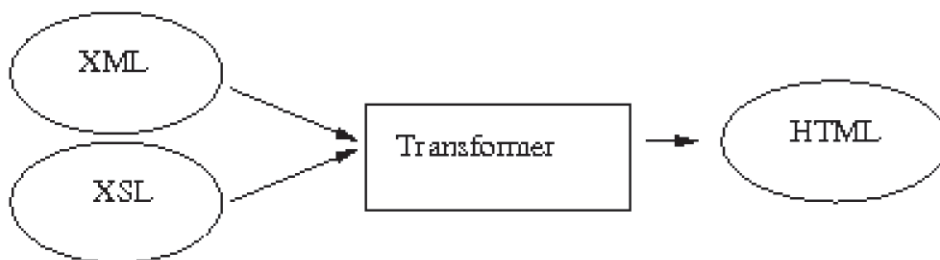
รูปที่ 4 ตัวอย่างเอกสาร XML ที่เป็นภาษาไทย

2. ความรู้พื้นฐานเกี่ยวกับ XSL

XSL เป็นภาษา XML ภาษาหนึ่งที่ใช้ร่วมกับ XSLT ซึ่งแปลงข้อมูลจากรูปแบบหนึ่งให้เป็นอีกรูปแบบหนึ่ง ข้อมูลที่อยู่ในภาษา XSL จะเรียกว่าสไลต์ชีตซึ่งสามารถกำหนดรูปแบบการแสดงผลตามที่ต้องการ ทำให้แอปพลิเคชันหลายอันสามารถใช้ข้อมูลชุดเดียวกันในหลากหลายรูป

แบบแล้วแต่สไลต์ชีตกำหนด

ในการแปลงข้อมูล XML ให้เป็น HTML นั้นจะต้องมีอินพุตไฟล์ที่เป็น XSL โดยจะมีโปรแกรมที่ใช้ในการแปลง (Transformer) หรือตัวประมวลผลภาษา XSL (XSLT) ดังแสดงไว้ในรูปที่ 4



รูปที่ 5 การแปลงข้อมูล XML ให้เป็นข้อมูล HTML

ตัวอย่างของการแปลงข้อมูลจาก XML และ XSL ให้เป็น HTML ได้แสดงไว้ผ่านตัวอย่างโดยแสดงไฟล์ XML ในรูปที่ 6, ไฟล์ XSL ในรูปที่ 7, ไฟล์ HTML ในรูปที่ 8 และการแสดงผลโดยใช้เบราว์เซอร์ในรูปที่ 9

ไฟล์ catalog.xml ในรูปที่ 6 เป็นไฟล์ที่เก็บข้อมูลเกี่ยวกับ catalog ซึ่งจะมีข้อมูล cd ที่น่าสนใจโดยที่ข้อมูล cd ประกอบด้วยข้อมูล title, artist, และ price ของ cd นั้น

```

<?xml-stylesheet type="text/xsl" href="catalog.xsl"?>
<catalog>
  <cd lang="Thai">
    <title>Time Machine</title>
    <artist>Rose</artist>
    <price>120</price>
  </cd>
  <cd lang="Thai">
    <title>Million Ways to Love</title>
    <artist>Boyd Kosiyapong</artist>
    <price>130</price>
  </cd>
</catalog>

```

รูปที่ 6 ไฟล์ตัวอย่าง catalog.xml

ไฟล์ catalog.xml ดังแสดงในรูปที่ 7 นี้ระบุว่าจะให้แสดงไฟล์ในรูปแบบ HTML จากการเขียน `<xsl:output method="html"/>` ในบรรทัดที่ 4 อิลิเมนต์ที่ไม่มี `xsl:` นำหน้าคืออิลิเมนต์ในภาษา HTML ซึ่งจะแสดงในผลลัพธ์ไฟล์ที่ได้จากการแปลง ในขณะที่อิลิเมนต์ที่ขึ้นต้นด้วย `xsl:` จะเป็น อิลิเมนต์ของภาษา XSL Transform อย่างเช่น `<xsl:for-each select="catalog/cd">` ในบรรทัดที่ 14 ระบุให้ตัวประมวล

ผล XSL ทำการอ่านและประมวลแต่ละอิลิเมนต์ `cd` ที่อยู่ภายใต้อิลิเมนต์ `catalog` ส่วน `<xsl:value-of select="title"/>` ในบรรทัดที่ 16 ระบุให้โปรแกรมที่ใช้ในการแปลงเอกสารดึงและแสดงค่าของอิลิเมนต์ `title` ในทำนองเดียวกัน `<xsl:value-of select="artist"/>` ในบรรทัดที่ 17 ระบุให้โปรแกรมที่ใช้ในการแปลงเอกสารดึงและแสดงค่าของอิลิเมนต์ `artist`

```

1 <?xml version="1.0"?>
2 <xsl:stylesheet version="1.0"
3   xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
4   <xsl:output method="html"/>
5   <xsl:template match="/">
6     <html>
7       <body>
8         <h2>My CD Collection of 2007</h2>
9         <table border="1">
10          <tr bgcolor="#9acd32">
11            <th align="left">Title</th>
12            <th align="left">Artist</th>
13          </tr>
14          <xsl:for-each select="catalog/cd">
15            <tr>
16              <td><xsl:value-of select="title"/></td>
17              <td><xsl:value-of select="artist"/></td>
18            </tr>
19          </xsl:for-each>
20        </table>
21      </body>
22    </html>
23  </xsl:template>
24 </xsl:stylesheet>

```

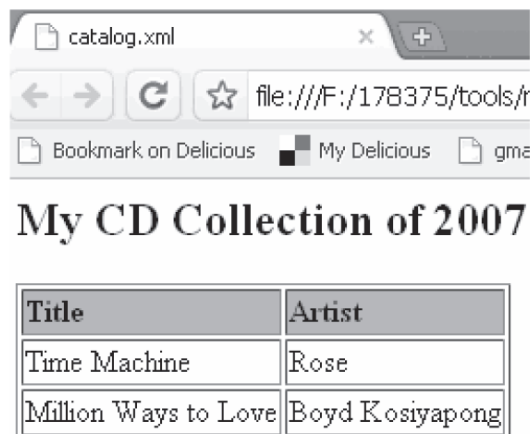
รูปที่ 7 ไฟล์ตัวอย่าง catalog.xsl

หากนำไฟล์ในรูปที่ 6 และไฟล์ในรูปที่ 7 เป็นอินพุตของโปรแกรมที่แปลงข้อมูลด้วยสไตล์ชีตในภาษา XSL จะได้

ผลลัพธ์ดังแสดงในรูปที่ 8 ซึ่งหากเปิดไฟล์ดังกล่าวด้วยเว็บเบราว์เซอร์จะได้ผลดังแสดงในรูปที่ 9

```
<html><body><h2>My CD Collection of 2007</h2>
<table border="1">
<tr bgcolor="#9acd32"><th align="left">Title</th><th align="left">Artist</th></tr>
<tr><td>Time Machine</td><td>Rose</td></tr>
<tr><td>Million Ways to Love</td><td>Boyd Kosiyapong</td></tr>
</table>
</body>
</html>
```

รูปที่ 8 ไฟล์ตัวอย่าง HTML ซึ่งเป็นผลลัพธ์จากการแปลงไฟล์ XML



รูปที่ 9 การดูข้อมูลผ่านเบราว์เซอร์ซึ่งแปลงข้อมูล XML ให้เป็น HTML

3. ความรู้พื้นฐานเกี่ยวกับ PHP

PHP คือภาษาสคริปต์ซึ่งจะทำงานในฝั่งเซิร์ฟเวอร์ แล้วส่งการแสดงผลมายังเบราว์เซอร์และนอกจากนี้มันยังเป็นสคริปต์ที่เป็นส่วนหนึ่งในเอกสาร HTML อีกด้วย ใน PHP5 จะมี extension 2 ตัวที่สามารถนำมาเขียนเอกสาร XML ซึ่งได้แก่ DOM และ XMLWriter โดยที่ DOM Extension ทำให้เราสามารถใช้ออกเอกสาร XML ได้โดยใช้ XMLWriter API ซึ่งโค้ดที่เรียกใช้ XMLWriter API จะกระชับและใช้หน่วยความจำได้อย่างมีประสิทธิภาพได้มากกว่า DOM และจากการค้นหาโดยใช้ Google Search เว็บแอปพลิเคชันของคนไทยที่ใช้ XMLWriter นั้นมีน้อยมาก บทความนี้จะช่วยเผยแพร่การเขียนเอกสาร XML อย่างมีประสิทธิภาพโดยใช้ XMLWriter ในภาษา PHP

เขียนโค้ด อีกทั้งใช้หน่วยความจำมากซึ่งทำให้ไม่เหมาะสมที่จะนำมาใช้ในโปรแกรมที่มีการอ่านและเขียนข้อมูลขนาดใหญ่ ส่วน XMLWriter Extension จะทำให้เราสามารถเขียนเอกสาร XML ได้โดยใช้ XMLWriter API ซึ่งโค้ดที่เรียกใช้ XMLWriter API จะกระชับและใช้หน่วยความจำได้อย่างมีประสิทธิภาพได้มากกว่า DOM และจากการค้นหาโดยใช้ Google Search เว็บแอปพลิเคชันของคนไทยที่ใช้ XMLWriter นั้นมีน้อยมาก บทความนี้จะช่วยเผยแพร่การเขียนเอกสาร XML อย่างมีประสิทธิภาพโดยใช้ XMLWriter ในภาษา PHP

และบทความนี้ได้นำเสนอการวัดประสิทธิภาพของการสร้างไฟล์ XML โดยใช้ DOM และ XMLWriter

การอ่านและการเขียนเอกสาร XML แต่ในบทความนี้จะนำเสนอเฉพาะการเขียนเอกสาร XML โดยเรียกใช้ DOM API สมมุติว่าถ้าเราจะสร้างเอกสาร XML ในรูปที่ 2 ด้วยการใช้ DOM เราสามารถทำได้ดังแสดงในโค้ดในรูปที่ 10

3.1 ความรู้เกี่ยวกับ DOM Extension

PHP DOM Extension เป็น Extension ที่นำไปใช้ใน

```

1 <?php
2 // create the root node of DOM tree
3 $doc = new DomDocument('1.0');
4 // create element 'nation'
5 $root = $doc->createElement('nation');
6 // assign the element 'nation' as the root node of the XML doc
7 // by add it as the child of the root node of DOM tree
8 $doc->appendChild($root);
9 // set attribute of the root node
10 $root->setAttribute('id','th');
11 // create element node 'name'
12 $name = $doc->createElement('name');
13 // assign text node with its content "Thailand"
14 $nameValue = $doc->createTextNode('Thailand');
15 // add the text name value node as a child to node 'name'
16 $name->appendChild($nameValue);
17 // create element location
18 $location = $doc->createElement('location');
19 // assign text node with its content "Southeast Asia"
20 $locationValue =
21 $doc->createTextNode('Southeast Asia');
22 // add the text node as a child to node 'location'
23 $location->appendChild($locationValue);
24 // add node 'name' as a child to node 'nation'
25 $root->appendChild($name);
26 // add node 'location' be a child to node 'nation'
27 $root->appendChild($location);
28 // save DOM tree in an XML file
29 $doc->save("nation.xml");
30 ?>

```

รูปที่ 10 โค้ดตัวอย่างที่ใช้ DOM API

ในการสร้างเอกสาร XML โดยใช้ DOM นั้นใช้หลักการสร้างโครงสร้างต้นไม้แบบ DOM โดยที่รูทโหนดของ DOM tree จะมีโหนดโหนดที่เป็นรูทโหนดของเอกสาร XML ในรูปที่ 2 โหนดโหนดที่เป็นรูทโหนดของเอกสาร XML คือโหนด nation ซึ่งมีโหนดลูกคือโหนด name และโหนด location โดยที่โหนด name จะมีเท็กซ์ภายในเป็น Thailand และโหนด location จะมีเท็กซ์ภายในเป็น Southeast Asia

จากโค้ดในรูปที่ 10 ก่อนอื่นจะต้องมีการสร้างออบเจกต์ DomDocument โดยเรียก constructor ของคลาส DomDocument ดังแสดงในบรรทัดที่ 3 จากนั้นก็ใช้ฟังก์ชัน

ของออบเจกต์ \$doc->createElement(элеmentName) เพื่อสร้างโหนดโหนด ในโครงสร้างต้นไม้ของ DOM นั้นจะแบ่งประเภทของโหนดต่างๆ ไว้หลายประเภท แต่ประเภทของโหนดที่มักจะถูกสร้างกันบ่อยคือ โหนดโหนด ซึ่งสร้างโดยเรียกใช้ฟังก์ชัน createElement เท็กซ์โหนด ซึ่งสร้างโดยเรียกใช้ฟังก์ชัน createTextNode ค่าของเท็กซ์ที่อยู่ภายในโหนด จะกำหนดได้โดยการสร้างเท็กซ์โหนดแล้วกำหนดค่าของเท็กซ์โหนดให้เป็นค่าที่ต้องการให้อยู่ภายในโหนด จากนั้นให้เท็กซ์โหนดดังกล่าวเป็นลูกของโหนดโหนดนั้น การสร้างความสัมพันธ์ระหว่างโหนดว่าเป็นพ่อและลูกกัน (parent-child relationship) สามารถทำได้โดยการเรียก

ใช้ฟังก์ชัน `appendChild` ในลักษณะ `$parent->appendChild($child)`

พิจารณาตัวอย่างการสร้างอิลิเมนต์ `<name>Thailand</name>` ก่อนอื่นสร้างอิลิเมนต์โหนดที่ชื่อว่า `name` ดังแสดงในบรรทัดที่ 12 จากนั้นสร้างเท็กโหนดที่มีค่า `Thailand` ดังแสดงในบรรทัดที่ 14 และให้เท็กโหนด `Thailand` เป็นลูกของอิลิเมนต์โหนด `name` ดังแสดงในบรรทัดที่ 16

ส่วนการสร้างแอตทริบิวต์นั้น สามารถทำได้โดยการเรียกใช้ฟังก์ชัน `setAttribute` ของอิลิเมนต์โหนดที่ต้องการให้มีแอตทริบิวต์นั้น โดยที่ฟังก์ชัน `setAttribute` จะรับทั้งชื่อและค่าของแอตทริบิวต์

หลังจากที่สร้างอิลิเมนต์ต่างๆ และแอตทริบิวต์ต่างๆ

ตามที่ต้องการแล้ว สามารถเขียนโครงสร้าง DOM นี้อยู่ในรูปแบบไฟล์ XML ได้โดยเรียกใช้ฟังก์ชัน `save` ของออบเจกต์ `DomDocument` โดยที่ฟังก์ชัน `save` จะรับอินพุตพารามิเตอร์เป็นชื่อของไฟล์ที่เราต้องการเขียนข้อมูล XML ลงไป ดังแสดงในบรรทัดที่ 29

3.2 ความรู้เกี่ยวกับ XMLWriter Extension

XML Writer Extension เป็น Extension ที่สามารถนำไปใช้ได้เฉพาะในการเขียนเอกสาร XML สมมุติว่าถ้าเราจะสร้างเอกสาร XML ในรูปที่ 3 ด้วยการใช้ XML Writer เราสามารถทำได้ดังแสดงในโค้ดรูปที่ 11

```

1 <?php
2 // create new XMLWriter object
3 $writer = new XMLWriter();
4 // save XML file name as 'nation.xml'
5 $writer->openURI('nation.xml');
6 // write XML declaration
7 $writer->startDocument('1.0');
8 // write start tag for element 'nation'
9 $writer->startElement('nation');
10 // write attribute 'id' with its value 'th'
11 $writer->writeAttribute('id', 'th');
12 // write element 'name' with its value 'Thailand'
13 $writer->writeElement('name', 'Thailand');
14 // write element 'location' with its value 'Southeast Asia'
15 $writer->writeElement('location', 'Southeast Asia');
16 $writer->endElement(); // end element nation
17 $writer->endDocument();
18 $writer->flush();
19 ?>

```

รูปที่ 11 โค้ดตัวอย่างที่ใช้ XMLWriter

ในการสร้างเอกสาร XML โดยใช้ XMLWriter นั้นใช้หลักการการเขียนข้อมูลแบบ stream ข้อมูลไหนปรากฏก่อนในเอกสาร XML ก็จะต้องถูกเขียนก่อนในโปรแกรม และหากอิลิเมนต์ใดยังไม่ปิดแล้วโปรแกรมมีการเขียนอิลิเมนต์ใหม่ อิลิเมนต์ใหม่นั้นก็จะอยู่ภายใต้อิลิเมนต์ที่ยังไม่ปิด

จากรูปที่ 11 ในการสร้างเอกสาร XML โดยใช้ XMLWriter นั้นจะต้องมีการสร้างออบเจกต์ XMLWriter ดังแสดงในบรรทัดที่ 3 จากนั้นก็ทำการเปิด URI โดยใช้ฟังก์ชัน `openURI` ซึ่งผู้พัฒนาโปรแกรมสามารถระบุชื่อไฟล์ XML ที่ต้องการให้ข้อมูลอยู่ในไฟล์นั้น ในบรรทัดที่ 7 การเรียก

ใช้ฟังก์ชัน `startDocument` ของออบเจกต์ XMLWriter จะเขียนองค์ประกอบแรกสุดของเอกสาร ซึ่งนั่นคือ XML declaration จากนั้นเริ่มเขียนอิลิเมนต์ `nation` ด้วยฟังก์ชัน `startElement` ในบรรทัดที่ 9 และเริ่มเขียนแอตทริบิวต์ด้วยฟังก์ชัน `writeAttribute` ในบรรทัดที่ 11 โดยกำหนดชื่อและค่าของแอตทริบิวต์เป็น `id` และ `th` ตามลำดับ จากนั้นสำหรับอิลิเมนต์โหนดที่มีเฉพาะค่าเท็กก็ก็สามารถเรียกใช้ฟังก์ชัน `writeElement`(ชื่ออิลิเมนต์, ค่าอิลิเมนต์) ได้เช่นการเขียนอิลิเมนต์ `name` ซึ่งมีเท็กภายในอิลิเมนต์เป็น `Thailand` ดังแสดงในบรรทัดที่ 13 หลังจากที่เขียนอิลิเมนต์ต่างๆ ครบแล้ว

ให้เรียกฟังก์ชัน endDocument() เพื่อระบุว่าสิ้นสุดการเขียนเอกสาร XML แล้วและให้เรียกฟังก์ชัน flush() เพื่อส่งข้อมูลทั้งหมดออกจากหน่วยความจำไปยังไฟล์ XML

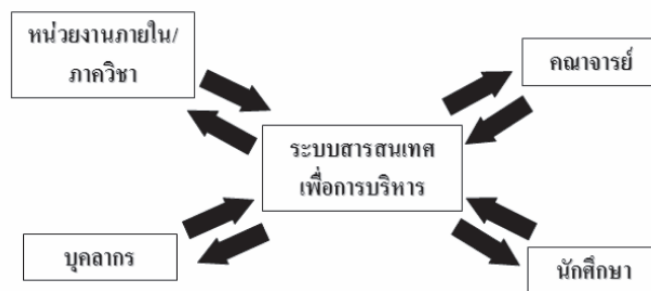
ภาพรวมของระบบสารสนเทศ

ระบบสารสนเทศเพื่อการบริหาร คณะวิศวกรรมศาสตร์ (Engineering Management Information System: ENMIS) มีเป้าหมายหลักในการดำเนินงานเพื่อใช้จัดเก็บข้อมูลของบุคลากรและนักศึกษาของคณะวิศวกรรมศาสตร์มหาวิทยาลัยขอนแก่น โดยที่ข้อมูลของอาจารย์ที่สำคัญประกอบด้วยภาระงานด้านการสอน, ด้านการวิจัยและวิชาการ, ด้านการบริการวิชาการ, ด้านการบริหาร ฯลฯ ข้อมูลของบุคลากรฝ่ายสนับสนุนประกอบด้วยข้อมูลการปฏิบัติงาน, การอบรมเพื่อเพิ่มพูนความรู้ทางวิชาการ และความชำนาญเฉพาะด้าน ฯลฯ ข้อมูลของนักศึกษาประกอบด้วยข้อมูล

เบื้องต้นนักศึกษา, โครงการนักศึกษา และกิจกรรมที่นักศึกษาเข้าร่วมตลอดหลักสูตร ฯลฯ

การใช้งานของระบบ

ระบบสารสนเทศเพื่อการบริหาร คณะวิศวกรรมศาสตร์พัฒนาโปรแกรมในรูปแบบของเว็บแอปพลิเคชันโดยใช้ภาษา PHP และใช้ MySQL เป็นระบบฐานข้อมูลเพื่อจัดการข้อมูล การเข้าใช้ระบบสารสนเทศ ผู้ใช้จะต้องกรอกชื่อผู้ใช้ (Username) และรหัสผ่าน (Password) ที่ทางระบบจัดสรรให้ ระบบจะมีการจัดแบ่งระดับความสามารถในการเข้าถึงข้อมูลเพื่อบันทึก แก้ไข ลบข้อมูล ซึ่งข้อมูลเหล่านี้จะนำไปใช้สรุปเป็นรายงาน เช่น รายงานภาระงานขั้นต่ำ, รายงานความดีความชอบ, รายงานสรุปสำหรับหน่วยงานประกันคุณภาพ (กพร., สมศ., กรอ. ฯลฯ) และรายงานสำหรับการประกันคุณภาพภายในคณะฯ ในระดับภาควิชา เป็นต้น



รูปที่ 12 ภาพรวมระบบสารสนเทศเพื่อการบริหาร คณะวิศวกรรมศาสตร์

ความต้องการในการส่งออกข้อมูลของระบบ

ระบบสารสนเทศเพื่อการบริหาร คณะวิศวกรรมศาสตร์เป็นระบบที่มีข้อมูลครบถ้วนมากที่สุด ณ ขณะนี้เนื่องจากบุคลากรในคณะทุกคนจะต้องเข้ามากรอกข้อมูลที่ระบบนี้เพื่อมีข้อมูลให้ผู้บริหารประกอบการพิจารณาในการเลื่อนขั้นหรือความดีความชอบ ด้วยเหตุผลดังกล่าว จึงทำให้มีความต้องการที่จะนำข้อมูลทางวิชาการที่อยู่ในระบบสารสนเทศนำมาใช้เพราะบ่อยครั้งที่บุคลากรในคณะมีเว็บไซต์ของตนเองและไม่ต้องการกรอกข้อมูลหลายรอบ

ปัจจุบันระบบสารสนเทศนี้ต้องเขียนโปรแกรมแบบเฉพาะเจาะจงในการตอบสนองความต้องการในการแสดงข้อมูลเป็นรายงานเอกสารตามที่หน่วยงานในคณะฯ ต้องการ และต้องมี

การล็อกอินเข้าสู่ระบบก่อนทุกครั้งจึงสามารถเข้าถึงข้อมูลที่ต้องการส่งออกได้ ดังนั้นการเข้าถึงข้อมูลในรูปแบบยืดหยุ่นที่บุคลากรต้องการจึงไม่สามารถทำได้ง่าย จะต้องมีการติดต่อขอให้นักพัฒนาโปรแกรมเปลี่ยนแปลงแก้ไขโปรแกรมทุกครั้งที่มีการแก้ไขรายงาน นอกจากนี้ความต้องการในรูปแบบการแสดงผลของแต่ละหน่วยงาน แต่ละบุคคลก็มีความแตกต่างกัน ดังนั้นจึงต้องมีโปรแกรมหลายชุดเพื่อตอบสนองความต้องการของรูปแบบการแสดงผลที่แตกต่างกันออกไป อีกทั้งข้อมูลดังกล่าวก็ไม่สามารถให้บุคคลอื่นสามารถเข้ามาดูข้อมูลที่บุคลากรภายในคณะต้องการเผยแพร่ อาทิเช่น ข้อมูลการตีพิมพ์บทความวิจัย

ด้วยเหตุนี้ระบบสารสนเทศจึงแก้ปัญหาโดยการหาวิธีส่งออกข้อมูลที่มีความยืดหยุ่นมากยิ่งขึ้น ด้วยการส่งออกข้อมูลให้อยู่ในรูปแบบ XML และใช้ไฟล์ XSL เพื่อกำหนดรูปแบบการแสดงผลของข้อมูล ข้อดีของการใช้ XML คือทำให้ผู้รับข้อมูลสามารถอ่านแล้วเข้าใจได้เพราะข้อมูล XML สามารถอธิบายข้อมูลด้วยตัวมันเอง ที่สำคัญคือสามารถสร้างโปรแกรมอ่านและค้นหาข้อมูล XML ได้ง่าย ข้อดีของการใช้ XSL ประกอบกับการใช้ XML คือสามารถใช้ไฟล์ XSL ตามความต้องการของการแสดงผลในแต่ละแบบโดยที่ไม่เปลี่ยนตัวเนื้อหาของข้อมูลซึ่งอยู่ในไฟล์ XML ทำให้ทุกคนใช้ข้อมูลที่มีเนื้อหาชุดเดียวกันแต่มีรูปแบบการแสดงผล

ตามความต้องการของแต่ละคนหรือแต่ละหน่วยงาน

ระบบเขียนและแสดงข้อมูล XML

ระบบเขียนข้อมูล XML

ในบทความนี้เราจะอธิบายระบบเขียนข้อมูล XML โดยใช้ XMLWriter เนื่องจากโค้ดสั้นและกระชับมากกว่าใช้ DOM ในที่นี้เราจะไม่อธิบายในรายละเอียดเกี่ยวกับคำสั่ง SQL ในการดึงข้อมูลจากฐานข้อมูล แต่จะเน้นในส่วนของการเขียนข้อมูลในรูปแบบ XML เช่น การแสดงบทความที่ได้รับการตีพิมพ์ โดยแยกตามภาควิชา ดังรูปที่ 13

```

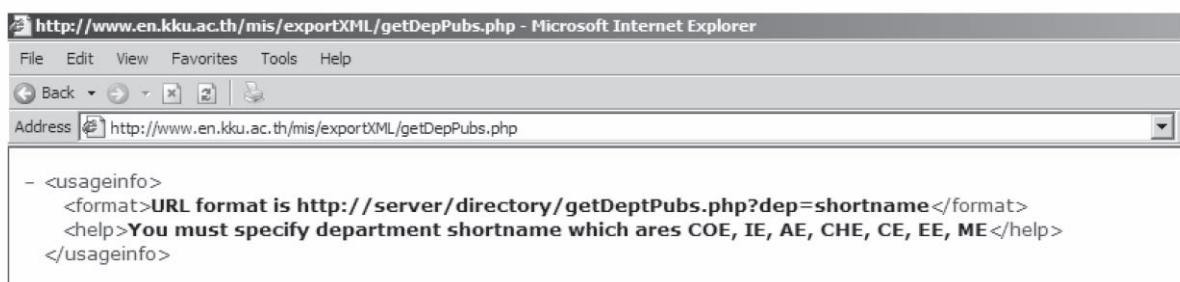
1 <?php
2
3 header('Content-type: text/xml');
4 if (!isset($_GET['dep'])) {
5     echo '<usageinfo><format>URL format is http://server/directory/getDeptPubs.php?dep=shortname</format>
6     <help>You must specify department shortname which ares COE, IE, AE, CHE, CE, EE, ME</help></usageinfo>';
7     exit;
8 }

```

รูปที่ 13 โค้ดตัวอย่างการแสดงวิธีการกรอก URL โดยเลือกตามภาควิชา

จากโค้ดในบรรทัดที่ 3 จะพบว่ามีการใช้ฟังก์ชัน header เพื่อกำหนดให้ข้อมูลที่แสดงผลจากไฟล์สคริปต์นี้เป็นข้อมูลที่เป็นเท็กซ์โดยให้อยู่ในรูปแบบ XML จากโค้ดในบรรทัดที่ 4 มี

การตรวจสอบว่ามีค่าให้ตัวแปร dep หรือไม่ ถ้าหากไม่มีการระบุค่าตัวแปรดังกล่าว โปรแกรมจะอธิบายวิธีใช้งาน ดังรูปที่ 14



รูปที่ 14 ตัวอย่างการแสดงวิธีการกรอก URL โดยเลือกตามภาควิชา

การแสดงผลข้อมูลบทความตีพิมพ์ตามภาควิชาหลังจาก Query ด้วยคำสั่ง SQL และนำมาเขียนข้อมูลในรูปแบบ XML โดยใช้ XML Writer จัดกลุ่มบทความที่ตีพิมพ์ตามภาควิชา

ให้แสดงผลข้อมูล ตามชื่อ-สกุลผู้เขียนบทความ ชื่อบทความ และปีที่ตีพิมพ์ ดังรูปที่ 15

```

44  $i = 0;
45  while ($i < $num_dep_published) {
46      $firstname = $results['firstname'];
47      $lastname = $results['lastname'];
48      $num_person_published = $staff[$firstname.$lastname];
49      $j = 0;
50
51      $writer->startElement('person');
52      $writer->writeAttribute('num_pubs', $num_person_published);
53      $writer->writeElement('firstname', $firstname);
54      $writer->writeElement('lastname', $lastname);
55      while ($j < $num_person_published) {
56          $year = $results['year'];
57          $title = $results['title'];
58          // echo "$dep, $firstname, $lastname, $year, $title<br/>";
59
60          $writer->startElement('article');
61          $writer->writeElement('title', $title);
62          $writer->writeElement('year', $year);
63          $writer->endElement(); // end element article
64          $i++;
65          $j++;
66          // echo "i = $i j = $j n = $n";
67          $results = mysql_fetch_assoc($result_pubs);
68          if ($j == $num_person_published || $i == $num_dep_published) {
69              break;
70          }
71      }
72      $writer->endElement(); // end element person
73      $writer->flush();
74  } // end while
75  $writer->endElement(); // end element people
76  $writer->endElement(); // end element dept
77  $writer->endElement(); // end element pubs
78  $writer->endDocument();
79  $writer->flush();
80  } // end while
81

```

รูปที่ 15 โค้ดตัวอย่างการแสดงผลข้อมูลบทความตีพิมพ์ โดยใช้ XML Writer

บรรทัดที่ 45 ในรูปที่ 15 เป็น while loop ตามจำนวนบทความทั้งหมดของภาควิชา (\$num_dep_published)

บรรทัดที่ 46 - 48 เป็นการกำหนดชื่อ ะ นามสกุล และจำนวนบทความของอาจารย์แต่ละท่านตามตัวแปร ชื่อ (\$firstname), สกุล (\$lastname) และ จำนวนบทความ (\$num_person_published)

บรรทัดที่ 51 - 54 เป็นการเริ่มเขียนอิลิเมนต์ person ด้วยฟังก์ชัน startElement ในบรรทัดที่ 51 เขียนแอตทริบิวต์ด้วยฟังก์ชัน writeAttribute ในบรรทัดที่ 52 โดยกำหนดชื่อและค่าของแอตทริบิวต์เป็น num_pubs และ ตัวแปร \$num_person_published เป็นตัวแปรที่เก็บค่าจำนวนบทความตีพิมพ์ต่ออาจารย์หนึ่งท่าน ตามลำดับ ในบรรทัดที่ 53

- 54 เรียกใช้ฟังก์ชัน writeElement เพื่อเขียนอิลิเมนต์ firstname และ lastname ซึ่งมีเท็กซ์ภายใน อิลิเมนต์เป็น \$firstname และ \$lastname ทั้ง \$firstname และ \$lastname เป็นตัวแปรที่เก็บค่าชื่อและนามสกุลของอาจารย์ จะได้ผลแสดงเป็น XML ดังรูปที่ 16

```
<person num_pubs="10">
  <firstname>Thavatchai</firstname>
  <lastname>Thivavarnvongs</lastname>
```

รูปที่ 16 ตัวอย่างการแสดงผลข้อมูลในรูปแบบ XML จากโค้ดบรรทัดที่ 51 -54

บรรทัดที่ 55 - 63 เป็นการวน loop เพื่อแสดงชื่อบทความที่ตีพิมพ์และวันที่มีการตีพิมพ์ของบทความ ซึ่งมีการทำงานคล้ายกับโค้ดในบรรทัดที่ 51 - 54 ดังที่ได้อธิบายข้างต้น หลังจากที่เขียนอิลิเมนต์ต่างๆ ครบแล้ว ให้เรียกฟังก์ชัน endDocument () เพื่อระบุว่าสิ้นสุดการเขียนเอกสาร XML

แล้วในโค้ดบรรทัดที่ 63 และ 72 และเรียกฟังก์ชัน flush () เพื่อส่งข้อมูลทั้งหมดออกจากหน่วยความจำไปยังไฟล์ XML ดังโค้ดในบรรทัดที่ 73 ได้มีการแสดงผลในรูปแบบ XML ดังรูปที่ 17

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xml" href="pubs.xml"?>
- <pubs>
- <dep name="COE" num_pubs="59">
- <people>
- <person num_pubs="3">
  <firstname>Bhichate</firstname>
  <lastname>Chiewthanakul</lastname>
- <article>
  <title>The Performance on the Prevention of Packet Reordering</title>
  <date>2006-01-25</date>
</article>
- <article>
  <title>การประกันประสิทธิภาพการป้องกันการจัดเรียงตัวใหม่ของแพ็กเก็ตในเราเตอร์แบบถ่วงสมดุล</title>
```

รูปที่ 17 ตัวอย่างการแสดงผลข้อมูลบทความตีพิมพ์ ซึ่งแสดงออกมาในรูปแบบ XML

ระบบแสดงข้อมูล XML

ข้อมูลที่ได้จากไฟล์ XML เป็นการแสดงข้อมูลที่มีแต่ข้อความ ดังนั้นจึงจัดรูปแบบการแสดงผลข้อมูลให้สวยงามด้วย XSL ในรูปที่ 18 บรรทัดที่ 2 จะแสดงอิลิเมนต์ stylesheet ซึ่งเป็นรูทอิลิเมนต์ของไฟล์ XSL ที่ใช้ในการระบุรูปแบบการแสดงผลข้อมูล XML ในบรรทัดที่ 3 อิลิเมนต์ output กำหนดรูปแบบการแสดงผลของเอาต์พุตจากการใช้ไฟล์ XSL นี้ซึ่งในที่นี้เรากำหนดว่าให้เป็นไฟล์ HTML ด้วย encoding= "tis-

620" เพื่อให้รองรับการแสดงผลภาษาไทย จากนั้น บรรทัดที่ 4 ของรูปที่ 17 จะมีโค้ดเป็น <xsl: template match="people"> ซึ่งถ้าหากไฟล์อินพุต XML มีอิลิเมนต์ people ตัวประมวลผลไฟล์ XSL (XSLT Processor) ก็จะทำคำสั่งในบรรทัดที่ 5-9 และ 11-12 ซึ่งในที่นี้ก็คือใส่อิลิเมนต์ต่างๆ ในบรรทัดดังกล่าวลงไปไฟล์เอาต์พุต HTML ส่วนบรรทัดที่ 10 เป็นคำสั่งที่ให้ทำคำสั่ง <xsl:template> อื่นๆ

ของอิลิเมนต์ที่อยู่ภายใต้อิลิเมนต์ `people`

จากบรรทัดที่ 14 จะมีโค้ดเป็น `<xsl:template match="dep">` ซึ่งถ้าหากไฟล์อินพุต XML มีอิลิเมนต์ `dep` ภายใต้อิลิเมนต์ `people` ตัวประมวลผลไฟล์ XSL จะทำโค้ดตามที่กำหนดไว้ใน บรรทัดที่ 15-16 โดยบรรทัดที่ 15 เป็นคำสั่งที่แสดงชื่อ `shortname` ของภาควิชาที่ได้กำหนดผ่าน URL และแสดงจำนวนบทความทั้งหมดของภาควิชา เช่น **AE (Number of publications = 106)** จำนวนบทความ ส่วนบรรทัดที่ 16 เป็นบรรทัดที่บอกให้ทำคำสั่ง `<xsl:template>`

สำหรับอิลิเมนต์ `person` ที่อยู่ภายใต้อิลิเมนต์ `people`

จากบรรทัดที่ 18 จะมีโค้ดเป็น `<xsl:template match="person">` ซึ่งถ้าหากไฟล์อินพุตมีอิลิเมนต์ `person` ตัวประมวลผลจะทำได้ตามกำหนดไว้ในบรรทัดที่ 19-28 โดยจะแสดงชื่อ-นามสกุล และจำนวนบทความตีพิมพ์ทั้งหมดของผู้เขียนแต่ละคน เช่น **Sununtha Kingpaiboon (Number of publications = 5)** สำหรับบรรทัดที่ 23-24 จะแสดงชื่อบทความและวันเดือนปีที่ตีพิมพ์บทความนั้น ดังรูปที่ 18

```

1 <?xml version='1.0'?>
2 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version='1.0'>
3 <xsl:output method="html" encoding="tis-620"/>
4 <xsl:template match="people">
5 <html>
6 <meta http-equiv="Content-Type" content="text/html; charset=tis-620"/>
7 <head><title>Publication of Faculty of Engineering, KKU</title></head>
8 <body>
9 <h1>Publication of Faculty of Engineering, KKU</h1>
10 <xsl:apply-templates/>
11 </body>
12 </html>
13 </xsl:template>
14 <xsl:template match="dep">
15 <h2><xsl:value-of select="@name"/> (Number of publications = <xsl:value-of select="@num_pubs"/>)</h2>
16 <xsl:apply-templates select="people/person"/>
17 </xsl:template>
18 <xsl:template match="person">
19 <h3><xsl:value-of select="firstname"/><xsl:text> </xsl:text>
20 <xsl:value-of select="lastname"/> (Number of publications = <xsl:value-of select="@num_pubs"/>)</h3>
21 <table border="0">
22 <tr bgcolor="#9acd32">
23 <th align="left">Title</th>
24 <th align="left">Year</th>
25 </tr>
26 <xsl:for-each select="article">
27 <tr><td><xsl:value-of select="title"/></td>
28 <td><xsl:value-of select="year"/></td></tr>
29 </xsl:for-each>
30 </table>
31 </xsl:template>
32 </xsl:stylesheet>

```

รูปที่ 18 โค้ดตัวอย่างการแสดงผลในรูปแบบ XSL

AE (Number of publications = 107)**Sununtha Kingpaiboon (Number of publications = 5)**

Title	Date
Diurnal Variation of Precipitable Water Vapor Based on GPS Observations	2005-11-12
การประยุกต์ GPS เพื่อประมาณค่าไอน้ำฝน	2006-01-25
Monitoring Precipitation from Precipitable Water Vapor(PWV) Based on GPS Observation	2007-01-22
A Suitable Schedule For Harvesting And Delivering Of Sugar Cane(Application Using Remote Sensing And GIS)	2006-10-10
Precipitable Water Vapor Change Obtained by Means of GPS and Its Application to the Monsoon Onset Decision at Khon Kaen, Thailand	2007-07-30

รูปที่ 19 แสดงชื่อบทความและวันเดือนปีที่ตีพิมพ์บทความ

การพัฒนาระบบสำหรับการแลกเปลี่ยนและแสดงข้อมูลโดยใช้ XMLWrite และ DOM กับ XSL ในระบบสารสนเทศของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น ในเบื้องต้นได้พัฒนา 3 ส่วนดังนี้

1. ส่วนของการส่งออกข้อมูลบทความที่ตีพิมพ์ของคณาจารย์ทั้งคณะวิศวกรรมศาสตร์ ด้วย XMLWrite

สามารถเข้าถึงได้โดยกำหนดให้ URL เป็น <http://www.en.kku.ac.th/mis/exportXML/getEnPubs.php> ซึ่งจะได้ผลลัพธ์แสดงในรูปที่ 20 และการส่งข้อมูลด้วย DOM กับ XSL สามารถเข้าถึงได้โดยกำหนดให้ URL เป็น <http://www.en.kku.ac.th/mis/exportXML/getEnPubsD.php> ซึ่งจะได้ผลลัพธ์แสดงในรูปที่ 21

```

<?xml version="1.0" ?>
<?xml-stylesheet type="text/xml" href="pubs.xsl"?>
- <pubs>
- <dep name="AE" num_pubs="107">
- <people>
- <person num_pubs="7">
  <firstname>Banchongsri</firstname>
  <lastname>Jeeravipoolvarn</lastname>
  - <article>
    <title>การติดตามประเมินผลจรรยาบรรณ นักศึกษาวิศวกรรมเกษตร ในการประชุมวิชาการ และนำเสนอผลงานวิจัย ประจำปี 2548</title>
    <date>2005-11-24</date>
  </article>
  - <article>
    <title>การศึกษารายการติดตามประเมินผลจรรยาบรรณนักศึกษาวิศวกรรมเกษตร</title>
    <date>2006-01-23</date>
  </article>
  - <article>
    <title>การศึกษาเรื่องการติดตามประเมินผลจรรยาบรรณ นักศึกษาวิศวกรรมเกษตร ในการประชุมวิชาการครั้งที่ 7</title>
    <date>2006-01-23</date>
  </article>
  - <article>
    <title>การศึกษารายการติดตามประเมินผล จรรยาบรรณ นักศึกษาวิศวกรรมเกษตร ในการประชุมวิชาการ การศึกษาทั่วไป ระดับประเทศ ครั้งที่ 3</title>
    <date>2006-03-24</date>
  </article>
  - <article>
    <title>การใช้ประโยชน์จากโรงงานน้ำตาลเพื่อผลิตเชื้อเพลิงอัดแท่ง</title>
    <date>2006-07-29</date>
  </article>
  - <article>
    <title>การศึกษาเรื่องการติดตามประเมินผล จรรยาบรรณนักศึกษาวิศวกรรมเกษตร</title>
    <date>2006-12-03</date>
  </article>
- </person>
- </people>
- </dep>
- </pubs>

```

รูปที่ 20 ข้อมูลบทความตีพิมพ์ทั้งหมดภายในคณะ (ในรูปแบบ XML)

AE (Number of publications = 107)

Banchongsri Jeeravipoolvarn (Number of publications = 7)

Title	Date
การติดตามประเมินผลจรรยาบรรณ นักศึกษาวิศวกรรมเกษตร ในการประชุมวิชาการ และนำเสนอผลงานวิจัย ประจำปี 2548	2005-11-24
การศึกษาเรื่องการติดตามประเมินผลจรรยาบรรณ นักศึกษาวิศวกรรมเกษตร ในการประชุมวิชาการครั้งที่ 7	2006-01-23
การศึกษาการติดตามประเมินผลจรรยาบรรณนักศึกษาวิศวกรรมเกษตร	2006-01-23
การศึกษาติดตามประเมินผล จรรยาบรรณ นักศึกษาวิศวกรรมเกษตร ในการประชุมวิชาการ การศึกษาทั่วไป ระดับประเทศ ครั้งที่ 3	2006-03-24
การใช้ประโยชน์จากอ้อยจากโรงงานน้ำตาลเพื่อผลิตเชื้อเพลิงอัดแท่ง	2006-07-29
การศึกษาเรื่องการติดตามประเมินผล จรรยาบรรณนักศึกษาวิศวกรรมเกษตร	2006-12-03
ศึกษากาการทะเลเปลี่ยนเกลือสมุทรสู่ป่า	2007-01-22

Seree Wongpichet (Number of publications = 11)

Title	Date
ผลกระทบจากความเร็วรอบของปืนทอพรตต่อความสูญเสียของเส้นใยในการทอพรต : กรณีศึกษาโรงงานผลิตพรต	2004-10-18
คุณสมบัติทางกายภาพบางประการของอ้อย	2005-10-10
ผลของการตั้งค่าเครื่องกะเทาะแบบลูกยางต่อปริมาณการแตกหักของข้าวกล้องและปริมาณข้าวเปลือกที่ไม่ถูกกะเทาะสำหรับเครื่องกะเทาะระดับห้องปฏิบัติการ	2005-10-10
การศึกษาเพื่อพัฒนาคุณภาพด้านขนาดของยางแผ่นดิบ	2006-08-01
การศึกษาเพื่อพัฒนาคุณภาพด้านขนาดของยางแผ่นดิบ	2006-08-20
Simple Modeling of Thin Paddy Layer for Catalytic Infrared Drying	2007-01-22
การศึกษาแนวทางการคัดแยกดินเพื่อลดการปนเปื้อนในกระบวนการผลิตน้ำตาลทราย	2007-01-22
A Study on the Approach of Soil Removal to Reduce Contaminatio in Sugar Processing	2008-01-28
Drying Characteristics and Milling Quality Aspects of Paddy	2008-05-30

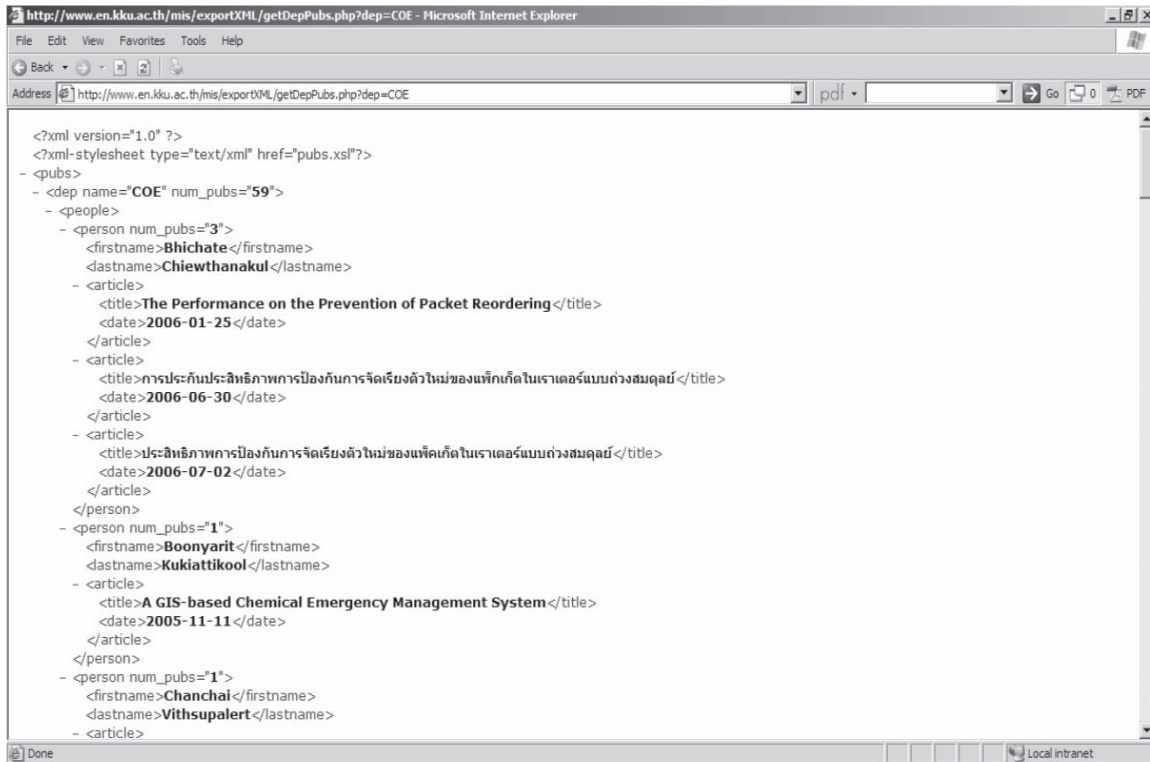
รูปที่ 21 ข้อมูลบทความตีพิมพ์ทั้งหมดภายในคณะ (ในรูปแบบ HTML)

จากรูปที่ 20-21 จะเห็นผลของการแสดงข้อมูลโดยแยกตามภาควิชา (ในรูปแบบภาควิชาวิศวกรรมเกษตร ซึ่งมีชื่อย่อว่า AE) และมีจำนวนบทความตีพิมพ์ของบุคลากรทั้งหมดในภาควิชา นั้น (ในรูปแบบที่ตีพิมพ์คือ 107) จากนั้นจะเป็นชื่อของบุคลากร และจำนวนบทความที่ตีพิมพ์ของบุคลากรทั้งนั้น และข้างล่างชื่อของบุคลากร จะเป็นรายชื่อของบทความและวันเดือนปีที่บทความนั้นตีพิมพ์

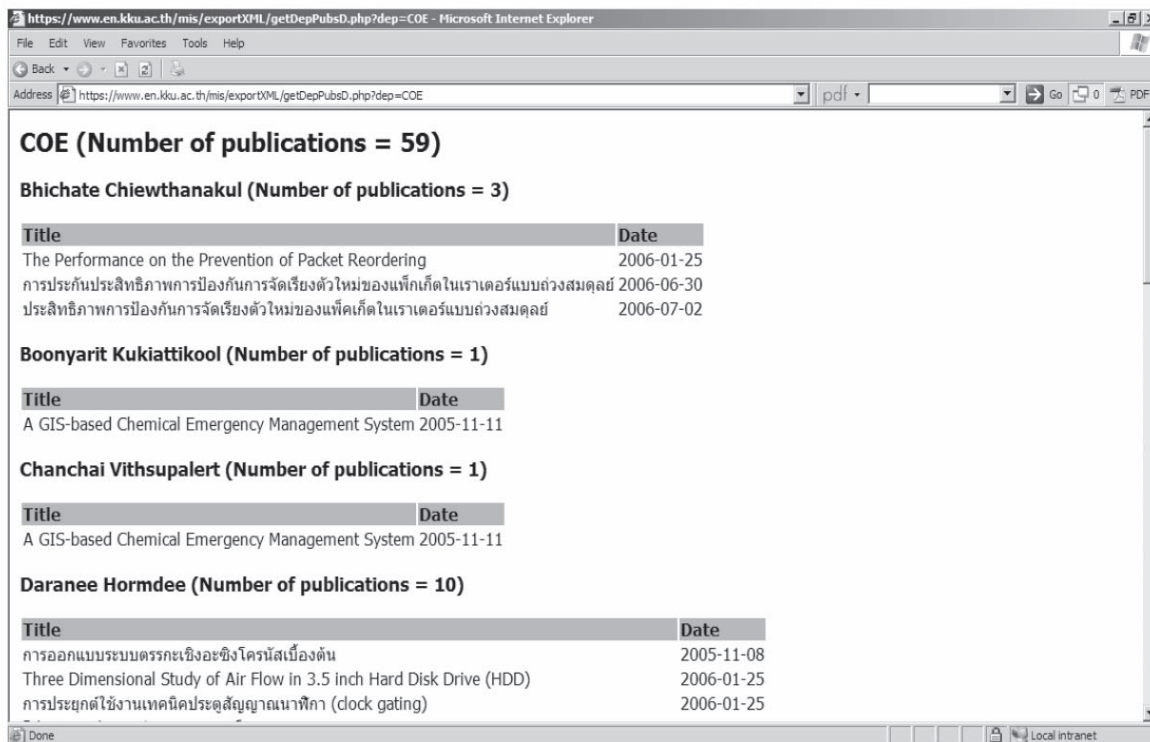
2. ส่วนของการส่งออกข้อมูลบทความที่ตีพิมพ์โดยระบบภาควิชา ด้วย XMLWrite เช่น หากต้องการเลือกภาควิชาวิศวกรรมคอมพิวเตอร์ซึ่งมีชื่อย่อว่า COE สามารถใช้ URL เป็น <http://www.en.kku.ac.th/mis/exportXML/getDepPubs.php?dep=COE> ซึ่งจะได้ผลลัพธ์แสดงในรูปแบบที่

22 และการส่งออกข้อมูลการตีพิมพ์โดยระบบภาควิชาด้วย DOM กับ XSL สามารถเข้าถึงได้โดยกำหนดให้ URL เป็น <http://www.en.kku.ac.th/mis/exportXML/getDepPubsD.php?dep=COE> ซึ่งจะได้ผลลัพธ์แสดงในรูปแบบที่ 23

รูปที่ 22-23 เป็นผลจากการที่ระบุว่าต้องการจะได้ข้อมูลบทความที่ตีพิมพ์ของบุคลากรในภาควิชาวิศวกรรมคอมพิวเตอร์ ซึ่งมีชื่อย่อเป็น COE ซึ่งโปรแกรมที่ทำการส่งออกข้อมูลก็จะไปดึงข้อมูลเฉพาะบทความของบุคลากรของภาควิชาวิศวกรรมคอมพิวเตอร์ และแสดงผลในรูปแบบเดียวกันกับรูปแบบการแสดงผลของข้อมูลบทความของบุคลากรทั้งคณะ



รูปที่ 22 ข้อมูลบทความตีพิมพ์ภายในภาควิชาหนึ่ง (ในรูปแบบ XML)



รูปที่ 23 ข้อมูลบทความตีพิมพ์ภายในภาควิชาหนึ่ง (ในรูปแบบ HTML)

3. ส่วนของการส่งออกข้อมูลบทความที่ตีพิมพ์โดยระบุชื่อ-สกุลอาจารย์ ด้วย XMLWriter เช่น บทความที่ตีพิมพ์โดยอาจารย์ กานดา สายแก้ว ระบุ URL เป็น <http://www.en.kku.ac.th/mis/exportXML/getStaffPubs.php?fname=Kanda&lname=Saikaew> ซึ่งจะได้ผลลัพธ์แสดงใน

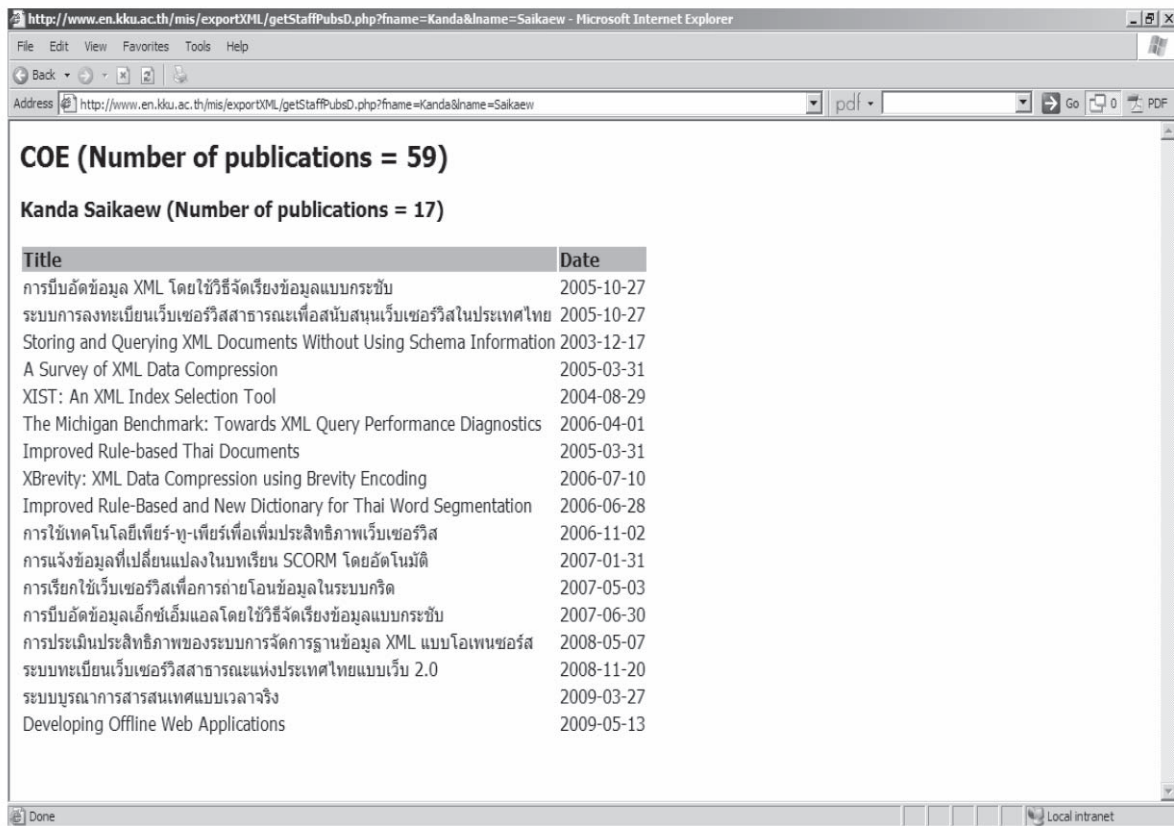
รูปที่ 24 และการส่งออกข้อมูลการตีพิมพ์โดยระบุชื่อ-สกุลอาจารย์ ด้วย DOM กับ XSL สามารถเข้าถึงได้โดยกำหนดให้ URL เป็น <http://www.en.kku.ac.th/mis/exportXML/getStaffPubsD.php?fname=Kanda&lname=Saikaew> ซึ่งจะได้ผลลัพธ์แสดงในรูปที่ 25

```

<?xml version="1.0" ?>
<?xml-stylesheet type="text/xml" href="pubs.xsl"?>
- <pubs>
- <dep name="COE" num_pubs="59">
- <people>
- <person num_pubs="17">
  <firstname>Kanda</firstname>
  <lastname>Saikaew</lastname>
  - <article>
    <title>การสืบค้นข้อมูล XML โดยใช้วิธีจัดเรียงข้อมูลแบบกระชับ</title>
    <date>2005-10-27</date>
  </article>
  - <article>
    <title>ระบบการลงทะเบียนเว็บเซอร์วิสสาธารณะเพื่อสนับสนุนเว็บเซอร์วิสในประเทศไทย</title>
    <date>2005-10-27</date>
  </article>
  - <article>
    <title>Storing and Querying XML Documents Without Using Schema Information</title>
    <date>2003-12-17</date>
  </article>
  - <article>
    <title>A Survey of XML Data Compression</title>
    <date>2005-03-31</date>
  </article>
  - <article>
    <title>XIST: An XML Index Selection Tool</title>
    <date>2004-08-29</date>
  </article>
  - <article>
    <title>The Michigan Benchmark: Towards XML Query Performance Diagnostics</title>
    <date>2006-04-01</date>
  </article>
  - <article>

```

รูปที่ 24 ข้อมูลบทความตีพิมพ์ของบุคลากรคนหนึ่ง (ในรูปแบบXML)



รูปที่ 25 ข้อมูลบทความตีพิมพ์ของบุคลากรคนหนึ่ง (ในรูปแบบHTML)

ผลการวัดประสิทธิภาพของโปรแกรม

ในการทดสอบวัดประสิทธิภาพของโปรแกรมนั้น ได้ใช้คลาส StopWatch ซึ่งเรียกใช้ฟังก์ชัน PHP microtime เพื่อจับเวลา [11] กับโปรแกรมที่รันบนเครื่องคอมพิวเตอร์ที่มี CPU 2.80 GHz, RAM 2 GB ใช้ซอฟต์แวร์ปฏิบัติการ Linux 2.6 เซิร์ฟเวอร์ Apache Version 2.2.9 และมีการ clear cache

ก่อนการทดลองทุกครั้ง

โดยที่ทดสอบกับไฟล์ข้อมูลโดยทำการประมวลผล 5 ครั้งต่อ 1 คำสั่งพร้อมจับเวลา ในผลการทดสอบทั้ง 5 ครั้ง ตัดค่าเวลาน้อยที่สุด และมากที่สุด แล้วหาค่าเฉลี่ยของ 3 ครั้งที่เหลือ ดังแสดงผลในตารางที่ 1

ตารางที่ 1 การเปรียบเทียบระยะเวลาในการได้คำตอบของการสร้างข้อมูล XML โดยใช้ XML Writer และ DOM Stem Cells: A N

ไฟล์ข้อมูล	ขนาดข้อมูล (KB)	ระยะเวลาในการได้คำตอบ (วินาที)	
		DOM	XML Writer
บทความทั้งหมดของคณะ	213	0.2100	0.0900
บทความของภาควิชาหนึ่ง	10	0.0149	0.0115
บทความของบุคลากรหนึ่ง	2	0.0123	0.0088

จากตารางที่ 1 จะเห็นได้ว่าหากใช้ XML Writer จะทำให้ได้ผลลัพธ์ที่ทำให้ระยะเวลาลดลงเกือบ 60% ในกรณีของการดึงข้อมูลบทความทั้งหมดของคณะ ส่วนการดึงข้อมูลของบทความของภาควิชาใดภาควิชาหนึ่ง หรือของบุคลากรคนใดคนหนึ่งนั้น ระยะเวลาใช้ไม่มาก เนื่องจากข้อมูลมีจำนวนไม่มาก เมื่อเปรียบเทียบกับกรณีของข้อมูลบทความทั้งคณะ

สรุป

บทความนี้แนะนำเสนอวิธีการส่งออกข้อมูลจากฐานข้อมูลให้อยู่ในรูปแบบ XML และแสดงข้อมูลโดยใช้ XSL ซึ่งทำให้ผู้รับข้อมูลสามารถดูข้อมูลและเข้าใจง่ายผ่านเว็บเบราว์เซอร์ นอกจากนี้ผู้รับข้อมูลเอกซ์เอ็มแอลยังสามารถค้นหาข้อมูลต่างๆ ได้ง่ายและรวดเร็วเพราะข้อมูล XML อธิบายข้อมูลด้วยตัวมันเอง

วิธีการที่นำเสนอในบทความนี้สามารถนำไปประยุกต์ใช้กับระบบสารสนเทศโดยทั่วไปที่พัฒนาโดยใช้ภาษาพีเอชพี และฐานข้อมูลมายเอสคิวแอลซึ่งเป็นโอเพนซอร์ส นอกจากนี้บทความนี้ยังนำเสนอวิธีการเขียนเอกสารเอกซ์เอ็มแอลโดยการใช้ XML Writer ซึ่งเขียนโค้ดได้กระชับกว่าและใช้เวลาน้อยกว่า DOM ในการสร้างเอกสาร XML

งานที่น่าสนใจที่จะทำในอนาคตคือการสร้างโปรแกรมยึดหยุ่นที่รับข้อมูลและรูปแบบการแสดงผลตามความต้องการของผู้ใช้ จากนั้นสร้างและส่งออกข้อมูลตามที่ผู้ใช้ระบุ

เอกสารอ้างอิง

1. สุชาติ สกีระนันท์ (2541) *เทคโนโลยีสารสนเทศสถิติ*:

ข้อมูลในระบบสารสนเทศ. ส. กรุงเทพฯ: ส. โรงพิมพ์ จุฬาลงกรณ์มหาวิทยาลัย.

2. The PHP Group. (2009). *PHP: Hypertext Preprocessor*. [online], Available : <http://www.php.net/> [2009, May 18].
3. Sun Microsystems. (2009). *MySQL: The world's most popular open source database*. [online], Available : <http://www.mysql.com/> [2009, May 18].
4. W3C. (2009). *Extensible Markup Language (XML)*. [online], Available : <http://www.w3.org/XML/> [2009, May 18].
5. Wikipedia. (2009). *Residual sum of squares (RSS)*. [online], Available : [http://en.wikipedia.org/wiki/RSS_\(file_format\)](http://en.wikipedia.org/wiki/RSS_(file_format)) [2009, May 18].
6. W3C. (2009). *MathML*. [online], Available : <http://www.w3.org/Math/> [2009, May 18].
7. Wikipedia. (2009). *MusicXML*. [online], Available : <http://en.wikipedia.org/wiki/MusicXML> [2009, May 18].
8. W3C. (2009). *The Extensible Stylesheet Language Family (XSL)*. [online], Available : <http://www.w3.org/Style/XSL/> [2009, May 18].
9. W3C. (1999). *XSL Transformation (XSLT)*. [online], Available <http://www.w3.org/Style/XSL/> [2009, May 18].
10. W3C. (2009). *W3C XHTML2 Working Group Home Page*. [online], Available : <http://www.w3.org/MarkUp/> [2009, May 18].
11. The PHP Group. (2009). *PHP: microtime - Manual*. [online], Available : <http://th2.php.net/microtime> [2009, May 19].