

# การบีบอัดข้อมูลเอ็กซ์เอ็มแอลโดยใช้วิธีจัดเรียงข้อมูลแบบกระชับ

## XML Data Compression using Brevity Encoding

ประพันธ์ เลขาโสภณ และ กานดา รุณนะพงศา

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น 40002

Tel./Fax : 0-4336-2160 E-mail : (superjing@gmail.com) , (krunapon@kku.ac.th)

---

**ABSTRACT** – XML becomes the standard language for data representation and exchange on the Internet because it is simple, flexible, and platform neutral. However, XML data is often large and verbose since it consists of many repeated tags which are used for self-describe data. The large size of data results in an excessive amount of space required for storing data on the disk space and time required for transmitting data over the network. Thus, it is necessary to find an effective compression technique for XML data. Previous compression techniques generate the compressed XML data that is in the binary format or in the proprietary format to only that XML compressor tool. Such format is often incomprehensible to others or difficult to automatically parse. In this work, we propose XBrevity, an XML compressor which supports compressing and uncompressing XML data. XBrevity adopts a novel encoding method that has the compressed XML data in the XML format. Thus, the compressed XML data still preserves the advantage features of XML but the XML document has the smaller size.

**KEYWORDS** – XML, Compression, Data exchange, Brevity

**บทคัดย่อ** – ภาษาเอ็กซ์เอ็มแอล (Extensible Markup Language, XML) ได้วิวัฒนาการมาเป็นภาษามาตรฐานในการเสนอและแลกเปลี่ยนข้อมูลบนอินเทอร์เน็ตเนื่องจากภาษา XML เป็นภาษาที่ใช้ง่ายมีความยืดหยุ่น และไม่ขึ้นอยู่กับระบบปฏิบัติการของเครื่องคอมพิวเตอร์ แต่ว่าข้อมูลที่เป็นภาษา XML นั้นมักจะมีขนาดใหญ่และมีข้อมูลที่ซ้ำซ้อนอันเนื่องมาจากการใช้แท็กที่ซ้ำ ๆ ในการอธิบายข้อมูล เอกสาร XML จึงมีขนาดใหญ่ซึ่งทำให้ต้องการพื้นที่ในการจัดเก็บในปริมาณมากและใช้เวลานานในการส่งข้อมูล ดังนั้นงานวิจัยที่ต้องการจะบีบอัดข้อมูลในภาษา XML จึงมีความจำเป็นอย่างยิ่ง งานวิจัยที่ได้ทำมาแล้วนั้น ข้อมูลที่ถูกบีบอัดแล้วอาจจะอยู่ในรูปแบบของไบนารีหรือรูปแบบที่เป็นที่เข้าใจกับผู้พัฒนาเครื่องมือเท่านั้น ในงานวิจัยนี้ได้นำเสนอวิธีบีบอัดข้อมูลแบบใหม่ชื่อว่า “เอ็กซ์เบรวิตี้ (XBrevity)” ซึ่งเป็นวิธีการที่ใช้ในการบีบอัดข้อมูลหรือคลายการบีบอัดข้อมูลโดยที่ข้อมูลที่ถูบบีบอัดสามารถเป็นที่เข้าใจได้กับคนทั่วไป การบีบอัดข้อมูลโดยที่เอกสารที่ถูบบีบอัดอยู่ในรูปแบบของภาษา XML แบบย่อนั้นจะทำให้เอกสารนั้นยังคงข้อดีของภาษา XML ในขณะที่ขนาดของเอกสารเล็กลงข้อมูลที่ถูบบีบอัดอยู่ในรูปแบบของภาษา XML

**คำสำคัญ** – เอ็กซ์เอ็มแอล, การบีบอัดข้อมูล, การแลกเปลี่ยนข้อมูล, การย่อ

---

## 1. บทนำ

ปัจจุบันภาษาเอ็กซ์เอ็มแอล (Extensible Markup Language, XML) [15] ได้เข้ามามีบทบาทและเป็นมาตรฐานในการแลกเปลี่ยนข้อมูล เนื่องจาก XML มีความสามารถในการอธิบายความหมายของข้อมูลและมีความยืดหยุ่นในการใช้งาน การนำ XML มาใช้งานสามารถทำได้โดยการใช้แท็กเป็นตัวกำกับและการตั้งชื่อแท็กที่สื่อถึงความหมายของข้อมูล ทำให้เอกสารที่ถูกสร้างขึ้นเข้าใจได้ง่าย จึงเป็นส่วนสำคัญที่ทำให้การเข้าถึงข้อมูลได้ง่ายขึ้น แต่จากการที่มีการใช้แท็กเข้ามาช่วยในการสื่อถึงความหมายทำให้เกิดการบันทึกแท็กชนิดเดียวกันบ่อยครั้งในเอกสาร

```
<?xml version="1.0" encoding="UTF-8"?>
<weblog>
  <entry>
    <host>133.5.33.76</host>
    <referer>-</referer>
    <userAgent>-</userAgent>
    <dateTime>9/Aug/2001:00:05:30</dateTime>
    <reqID>-0500</reqID>
    <reqType>GET</reqType>
    <resource>/publish/programming/pippy1.gif
      </resource>
    <protocol>HTTP/1.1</protocol>
    <statusCode>200</statusCode>
    <byteCount>5440</byteCount>
  </entry>
  <entry>
    <host>133.5.33.76</host>
    <referer>-</referer>
    <userAgent>-</userAgent>
    <dateTime>9/Aug/2001:00:05:30</dateTime>
    <reqID>-0500</reqID>
    <reqType>GET</reqType>
    <resource>/publish/programming/pippy_multi
```

```
eval.gif</resource>
<protocol>HTTP/1.1</protocol>
<statusCode>200</statusCode>
<byteCount>4236</byteCount>
</entry>
</weblog>
```

รูปที่ 1 ตัวอย่างเอกสาร XML ที่ใช้งานจริง

จากรูปที่ 1 จะเห็นได้ว่ามีการบันทึกข้อมูลประเภทเดียวกันหลายครั้ง ซึ่งในแต่ละครั้งจะมีรายละเอียดแตกต่างกัน ด้วยเหตุนี้ ขนาดของเอกสารจึงมีขนาดใหญ่เมื่อเทียบกับขนาดข้อมูลจริงภายในเอกสารนั้น ส่งผลให้สิ้นเปลืองเนื้อที่หากต้องการจัดเก็บเอกสารและสิ้นเปลืองเวลาในการรับส่ง หากต้องการแลกเปลี่ยนข้อมูลระหว่างองค์กรผ่านระบบเครือข่าย

โดยทั่วไปแล้วข้อมูล XML จะเก็บอยู่ในรูปแบบของไฟล์ ดังนั้นข้อมูล XML ที่จะมีการนำไปใช้กันได้จริงควรจะเป็นการบีบอัดไฟล์ XML ที่ทำให้มีขนาดเล็กและนำไปใช้งานได้ง่าย เนื่องจากข้อมูล XML จำนวนมากจะเก็บไว้ในไฟล์ ดังนั้นการบีบอัดไฟล์ XML จึงมีความจำเป็นอย่างยิ่ง

โดยในปัจจุบัน XML ได้ถูกนำมาใช้งานในหลายสาขาวิชาชีพ ไม่ว่าจะเป็นทางธุรกิจซึ่งมีการนำอีบีเอ็กซ์เอ็มแอล (Electronic Business XML, ebXML) [14] และ บีเพล (Business Process Execution Language, BPEL) [2] ไปใช้ ด้านกราฟฟิกซึ่งมีการนำเอสวีจี (Scalable Vector Graphics, SVG) [4] ไปใช้ หรือแม้กระทั่งด้านวิทยาศาสตร์เคมี ซึ่งมีการนำภาษาซีเอ็มแอล (Chemical Markup Language, CML) [10] ไปใช้ ไม่ว่าจะเป็ BPEL, SVG, หรือ CML ต่างก็เป็นภาษา XML ประเภทหนึ่ง

นอกจากนี้แอปพลิเคชันอีกอันหนึ่งที่สำคัญของภาษา XML ซึ่งก็คือเว็บเซอร์วิส (Web Services) [16] เป็นซอฟต์แวร์ที่ได้นำเอามาใช้อย่างมากในปัจจุบัน และคาดว่าจะมีการใช้อย่างแพร่หลายมากขึ้นในอนาคต ในปัจจุบันได้มีการนำเว็บเซอร์วิสเอามาใช้ในการให้บริการ ผ่านอินเทอร์เน็ตโดยบริษัทชั้นนำ อาทิเช่น Yahoo Search Web Services [18], Google Web APIs [6], Amazon Web Services [1], และ eBay API [3] จุดเด่นของเทคโนโลยีเว็บเซอร์วิสคือ การที่มันทำให้โปรแกรมที่พัฒนาโดยภาษาและใช้แพลตฟอร์มที่แตกต่างกันสามารถติดต่อและ

ทำงานร่วมกันได้โดยใช้ภาษา XML เป็นภาษากลางในการแลกเปลี่ยนข้อมูล ฉะนั้นจะเห็นได้ว่าข้อมูล XML จะมีปริมาณเพิ่มมากขึ้นและขนาดของข้อมูลของ XML ที่มักจะมียกขนาดใหญ่จะส่งผลกระทบต่อประสิทธิภาพการทำงานของแอปพลิเคชันที่ใช้ XML เป็นภาษาในการบันทึกข้อมูล

จากปัญหาในเรื่องขนาดของข้อมูลแนวทางที่สามารถนำมาใช้ในแก้ปัญหาได้คือการบีบอัดข้อมูล XML (XML data compression) เพื่อลดขนาดของเอกสารซึ่งเป็นการบีบอัดข้อมูล XML โดยเฉพาะทำให้สามารถเพิ่มประสิทธิภาพในการบีบอัดได้ดีกว่าการใช้วิธีการบีบอัดข้อมูลทั่วไป

ในบทความนี้ในหัวข้อที่ 2 จะได้นำเสนอกระบวนการบีบอัดแบบต่างๆ ที่มีอยู่ในปัจจุบันซึ่งได้แก่ XMill, XGrind, XPRESS และ XPACK จากนั้นหัวข้อที่ 3 จะมีเนื้อหาในส่วนของแนวทางในการพัฒนาเทคนิคใหม่ๆ ในการบีบอัดข้อมูล ผลการทดลองจะนำเสนอในหัวข้อที่ 4 บทสรุปอยู่ในหัวข้อที่ 5 และในหัวข้อที่ 6 กล่าวถึงแนวทางในการพัฒนาในอนาคต

## 2. งานวิจัยที่เกี่ยวข้อง

โดยปกติแล้วในการบีบอัดไฟล์ข้อมูลทั่วไปนั้น จะนิยมใช้ gzip [5] เนื่องจากเป็นซอฟต์แวร์ที่นำไปใช้ได้โดยไม่ต้องเสียค่าใช้จ่าย อีกทั้งสามารถบีบอัดข้อมูลได้ดี และไม่จำเป็นต้องมีข้อมูลเกี่ยวกับโครงสร้างของเอกสาร แต่ทำการบีบอัดไฟล์ XML โดยใช้ gzip นั้นมีข้อจำกัดที่ gzip ไม่สามารถตรวจสอบพอลิเมนต์ที่ซ้ำๆ ที่อาจจะไม่ได้ยึดติดกันแต่จะเป็นการบีบอัดไฟล์ทั่วไป ซึ่งไม่ได้ออกแบบมาให้ใช้กับ XML โดยเฉพาะ

งานวิจัยที่กล่าวถึงการบีบอัดข้อมูล XML ในปัจจุบันนี้มีอยู่หลายวิธีการได้แก่ XMill [7], XGrind [13], XPRESS [9] และ XPACK [8] ซึ่งมุ่งเน้นในการลดขนาดของข้อมูลที่เป็น XML ให้เล็กลง โดยแต่ละวิธีจะใช้เทคนิคที่แตกต่างกันไป

### 2.1 XMill

XMill เป็นวิธีการแรกที่ได้มีการนำเสนอวิธีการและเทคนิคในการบีบอัดเอกสาร XML โดยเริ่มจากการแยกส่วนแท็ก ซึ่งภายในประกอบไปด้วยอีลิเมนต์และแอตทริบิวต์ ออกจากข้อมูลที่เป็นตัวอักษร จากนั้นจะทำการจัดความสัมพันธ์ของ

กลุ่มข้อมูลเป็นเหมือนภาชนะบรรจุ หรือเรียกว่า containers โดยจัดให้ข้อมูลที่เป็นแบบเดียวกันอยู่ในกลุ่มเดียวกัน ในขั้นตอนต่อมาจะนำ containers แต่ละกลุ่มมาทำการบีบอัด ซึ่งจะได้ containers ที่อยู่ในรูปของข้อมูลที่บีบอัดแล้ว จากนั้นให้นำ containers ทั้งหมดโดยอาศัย gzip ทำการบีบอัดอีกครั้ง เพื่อให้ได้ข้อมูลที่ออกมาเป็นไฟล์เดียว เนื่องจากการจัดกลุ่มข้อความต้องอาศัยความเข้าใจของความหมายของข้อมูล ซึ่งขึ้นอยู่กับชนิดของแอปพลิเคชัน ฉะนั้น XMill จึงมักต้องให้ผู้ใช้โปรแกรมพิจารณาความหมายของข้อมูล อีกทั้งผู้ใช้โปรแกรม ไม่สามารถค้นหาข้อมูลที่ต้องการจากเอกสารที่ถูกบีบอัดแล้ว ถึงกระนั้นก็ตาม XMill ถือว่าเป็นบทความวิจัยแรกๆ ที่ทำให้ผู้วิจัยทั้งหลายตระหนักถึงความสำคัญของปัญหาและวิธีการแก้ปัญหาในการบีบอัดข้อมูล XML ซึ่งได้มีผู้นำเทคนิคบางเทคนิคของ XMill ไปใช้ในการเก็บและค้นหาข้อมูล XML [11]

### 2.2 XGrind

XGrind มีข้อดีที่ XMill ยังไม่สามารถทำได้ซึ่งก็คือ ผู้ใช้ XGrind สามารถค้นหาข้อมูลได้จากเอกสารที่ถูกบีบอัด คุณสมบัตินี้เป็นผลมาจากการที่ข้อมูลที่ถูกบีบอัดแล้ว ยังคงมีโครงสร้างของเอกสารเดิม แต่อย่างไรก็ตามผู้ใช้โปรแกรม XGrind ไม่สามารถตอบคำถามบางประเภทโดยไม่ได้คัดลอกการบีบอัดของข้อมูล ตัวอย่างของคำถามที่ต้องมีการคัดลอกการบีบอัดข้อมูลคือ คำถามที่ถามช่วงของค่า (range query) หรือคำถามที่ถามหาข้อมูลค้นหาบางส่วนที่ตรงกับข้อมูลที่มีอยู่ (partial match) นอกจากนี้ XGrind จะบีบอัดเฉพาะเอกสาร XML ที่มีเอกสาร DTD (Document Type Definition) ซึ่งเป็นเอกสารที่ระบุโครงสร้างของเอกสาร XML ซึ่งเอกสาร XML บางเอกสารอาจจะไม่มี DTD ก็ได้ ฉะนั้นผู้ใช้โปรแกรม XGrind จะต้องสร้าง DTD สำหรับเอกสาร XML ที่ยังไม่มี DTD

### 2.3 XPRESS

XPRESS มีข้อดีเช่นเดียวกับ XGrind ตรงที่สามารถจะค้นหาข้อมูลจากเอกสารที่ถูกบีบอัดแล้ว แต่ XPRESS ไม่ได้ใช้ DTD XPRESS ได้นำเสนอแนวคิดใหม่ที่ใช้การเข้ารหัสทางคณิตศาสตร์แบบผันกลับ (reverse arithmetic encoding) เป็นวิธีการในการจัดเรียงข้อมูลเพื่อให้การหาคำตอบสำหรับนิพจน์ของ XPath เป็นไปได้มีประสิทธิภาพ นอกจากนี้

XPRESS ได้พัฒนาการหาชนิดของข้อมูลโดยไม่ต้องอาศัยข้อมูลอินพุตจากผู้ใช้โปรแกรม แต่อย่างไรก็ตาม XPRESS มีข้อจำกัดที่ XPRESS จะไม่สามารถเข้าใจเอกสาร XML ที่มีการใช้ ID และ IDREF และไม่ได้กล่าวถึงวิธีการขยายข้อมูลที่ถูกรีบอัดให้เป็นข้อมูล XML ปกติ

### 2.4 XPACK

XPACK เป็นการบีบอัดเอกสาร XML โดยใช้วิธีการเชิงไวยากรณ์ในการบีบอัดและการขยายเอกสาร XML ส่วนประกอบหลักของ XPACK คือ Grammar Generator ทำหน้าที่ในการสร้างกฎไวยากรณ์ ส่วนที่สองคือ Compressor ทำหน้าที่บีบอัดเอกสาร และในส่วนสุดท้ายจะเป็น Decompressor ซึ่งทำหน้าที่ขยายเอกสารที่ผ่านการบีบอัดโดยอาศัยโครงสร้างเดิมของเอกสาร ข้อจำกัดของ XPACK ก็คือโปรแกรมไม่สามารถบีบอัดจัดการกับเอกสาร XML ที่มีอีลีเมนต์เป็นแบบผสมผสาน (Mixed-content) ซึ่งคืออีลีเมนต์ที่มีทั้งอีลีเมนต์และตัวอักษรอยู่ข้างใน และผู้ใช้โปรแกรมไม่สามารถค้นหาข้อมูลจากเอกสาร XML ที่ถูกบีบอัดแล้ว

### 3. เทคนิคการบีบอัดแบบ XBrevity

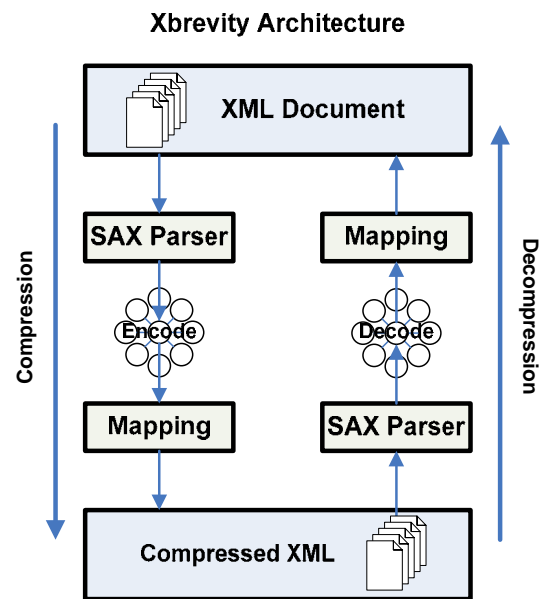
ในเทคนิคการบีบอัดด้วย XBrevity นั้นจะประกอบไปด้วย ส่วนของการบีบอัดเอกสาร XML (Compressor) และการขยายเอกสาร (Decompressor) เพื่อทำให้เอกสารที่ถูกบีบอัดด้วย XBrevity กลับมาเป็นเอกสาร XML ดั้งฉบับเดิมได้

ในโครงสร้างของ XBrevity จะใช้ SAX Parser (Simple API for XML) เป็นตัวอ่านเอกสารทั้งในกระบวนการการบีบอัดเอกสารและขยายการบีบอัด โดยในแต่ละกระบวนการจะมีการเข้ารหัสและถอดรหัสเอกสาร XML ที่ได้อ่าน ซึ่งโครงสร้างการทำงานของ XBrevity ได้แสดงไว้ในรูปที่ 2

#### 3.1 โปรแกรมบีบอัดข้อมูล (Compressor)

มีหน้าที่ในการบีบอัดเอกสาร XML ซึ่งจะใช้ SAX Parser เป็นตัวอ่านวิเคราะห์เอกสารที่จะทำการบีบอัดเข้ามา ในระหว่างที่ทำการอ่านวิเคราะห์เอกสารอยู่นั้น และจะทำการเข้ารหัสโดยการ mapping เพื่อเปลี่ยนรูปแบบให้เป็นตัวอักษรแล้วเขียนค่าลงไปบนไฟล์ โดยในการสร้างไฟล์ที่มีข้อมูลที่ถูกรีบอัดแล้ว

นั้น จะยังคงอยู่ในรูปแบบของภาษา XML และถูกต้องตามแบบฟอร์ม (Well-formed)



รูปที่ 2 แสดงโครงสร้างการทำงานของ XBrevity

โดยในเอกสารใหม่ที่ได้ประกอบไปด้วย `<d>...</d>` ซึ่งเป็นส่วนของข้อมูลที่ถูกรีบอัดแล้วกับ `<m .../>` ซึ่งเป็นส่วนของคำอธิบายความหมายของข้อมูลภายในเอกสาร (Metadata) ซึ่งเอกสารที่ถูกรีบอัดนี้สามารถไปจัดเก็บ หรือแลกเปลี่ยนข้อมูลระหว่างเครือข่ายได้ และสามารถเข้าใจได้ง่ายโดยไม่ต้องขยายข้อมูลกลับไปเป็นต้นฉบับเดิม เนื่องจากอยู่ในรูปแบบของเอกสาร XML แล้วและมีคำอธิบายข้อมูลอยู่ภายในให้ หรือจะทำการขยายเอกสารให้เป็นเอกสารเดิมได้ด้วยตัวขยายเอกสาร (Decompressor) ที่ออกแบบไว้

#### 3.2 โปรแกรมคลายการบีบอัดข้อมูล (Decompressor)

เมื่อเอกสารที่ถูกรีบอัดด้วย XBrevity ต้องการขยายให้กลับมาเป็นเอกสาร XML ดั้งฉบับสามารถทำได้โดยใช้ SAX Parser ที่ออกแบบมาเพื่อวิเคราะห์เอกสาร โดยทำการถอดรหัสเอกสารที่ถูกรีบอัดนั้นออกมา จากนั้นทำการ mapping ค่าที่ได้ ในแท็ก `<m .../>` แล้วนำค่าที่อ่านได้ไปแทนที่ตัวแปรเดิมที่อยู่ภายใน `<d>...</d>` เพื่อให้กลับออกมาเป็นเอกสารต้นฉบับ

### 3.3 ตัวอย่างการทำงานของ XBrevity

ในงานวิจัยนี้ได้นำเสนอการบีบอัดข้อมูล โดยไม่ใช่โครงสร้างของเอกสาร XML (XML Schema) [16,17,18] ซึ่งแสดงตามรูปที่ 3 ซึ่งเป็นตัวอย่างของเอกสารที่ประกอบไปด้วย อิลิเมนต์และแอตทริบิวต์ต่างๆ จะเห็นได้ว่าการใช้งานของแท็กที่ซ้ำๆ กันแต่ข้อมูลภายในแตกต่างกัน ทำให้โครงสร้างดูซับซ้อนและเอกสารมีขนาดใหญ่

```
<?xml version="1.0"?>
<bib>
  <article>
    <authors>
      <author aid="a1" eid="e1">
        <name>Pissamai<nickname>Aom
        </nickname></name>
      </author>
      <author aid="a2">
        <name>Veerasuk</name>
      </author>
    </authors> <year>2005</year>
  </article>
  <article>
    <authors>
      <author aid="a3">
        <name>Patumrut</name>
      </author>
    </authors>
  </article>
</bib>
```

รูปที่ 3 เอกสาร XML ตัวอย่าง A

ในการบีบอัดข้อมูลนั้นจะมีข้อมูลเมตาดาต้า (Meta data) เพื่อใช้อ้างอิงอิลิเมนต์และแอตทริบิวต์ ซึ่งทำให้โครงสร้างที่มีอยู่เดิมสั้นลง ทำให้ขนาดของเอกสารเล็กลงตามไปด้วย

```
<c>
  <d>
    <e1e2e3 a4="a1" a5="e1" e6v="Pissamai"
      e7v="Aom" />
    <xe1e2 e3a4="a2" e6v="Veerasuk" />
    <xe1 e8v="2005" />
    <e1e2e3 a4="a3" e6v="Patumrut" />
  </d>
  <m f="bib article authors author aid eid name
    nickname year "
    b="0 1 2 3 a4 a5 6 7 8" />
</c>
```

รูปที่ 4 เอกสาร XML ตัวอย่าง A ที่ถูกบีบอัดแล้ว

- โดยที่
- c** = ข้อมูลที่ถูกบีบอัด (Compressed)
  - d** = ข้อมูล (Data)
  - m** = ข้อมูลที่อธิบายข้อมูล (Metadata)
  - f** = ชื่อเต็ม (Full name)
  - b** = ชื่อย่อ (Brevity name)
  - v** = ค่าของข้อมูล (Value)
  - x** = การอ้างอิงอิลิเมนต์ที่เรียงซ้อนกันก่อนหน้า

จากรูปที่ 4 ก่อนการบีบอัดข้อมูล (แท็ก c) จะทำการ mapping ข้อมูลต่างๆ ให้เป็นตัวแปร และทำให้ข้อมูลเมื่อบีบอัดแล้วจะได้ขนาดที่เล็กลง ซึ่งข้อมูลจริงจะถูกบีบอัดอยู่ในแท็ก d โดยแท็กที่ขึ้นต้นด้วย e เป็นแท็กที่เป็นชื่อย่อของอิลิเมนต์ ส่วนแท็กที่ขึ้นต้นด้วย a เป็นแท็กที่เป็นชื่อย่อของแอตทริบิวต์ v เป็นตัวอักษรที่ชี้ให้เห็นว่าจะมีการเก็บค่าที่อยู่ในอิลิเมนต์ ตัวอักษร x เป็นตัวอักษรที่บ่งบอกว่าเป็นอิลิเมนต์อ้างอิงเพื่อใช้ในการเรียงลำดับของการซ้อนกันของอิลิเมนต์ที่เป็นบรรพบุรุษ (ancestor elements) ในอิลิเมนต์ที่เพิ่งพบก่อนหน้า ตัวอย่างเช่น <e1e2e3 a5="a1" .../> ตามด้วย <xe1e2 e3a4="a2" .../> นั้นบ่งบอกว่า <xe1e2 e3a4="a2" .../> มีการเรียงลำดับอิลิเมนต์ที่อยู่ภายใต้ e1e2 ในขณะที่บรรพบุรุษที่ติดกันมานั้นมีการจัดเรียงลำดับอิลิเมนต์เป็น <xe1 e8v .../> ตามลำดับไปจนจบ ส่วน e0 ไม่

จำเป็นต้องอยู่ในข้อมูลของการจัดเรียง เนื่องจากในแต่ละเอกสารมี root element ได้เพียงตัวเดียวเท่านั้น ดังนั้นในเอกสารที่ถูกบีบอัดจึงมี `<e1e2e3 .../>` แทนที่จะเป็น `<e0e1e2e3 .../>` และในการอ่านทำความเข้าใจเอกสารนั้นมีวิธีการคือ ในหนึ่งแท็กของเอกสารที่ถูกบีบอัด เช่น `<e1e2e3 a4="a1" a5="e1" e6v="Pissamai" e7v="Aom" />` จะอ่านโดยไล่จาก e1 ซึ่งไม่มีเนื้อหา ภายใต้อี e1 มี e2 ซึ่งไม่มีเนื้อหาเช่นกัน จะเรียงชิดติดกันกับ e1 ภายใต้อี e2 มี e3 ซึ่งมีแอตทริบิวต์อยู่ 2 ค่าด้วยกัน เมื่อเริ่มมีค่าของข้อมูลจะมีการแยกด้วยช่องว่าง เพื่อให้อยู่ในรูปของแอตทริบิวต์แล้วจึงแสดงค่าของข้อมูลออกมา จากนั้นในทุกๆค่าที่ถัดมาจะแยกด้วยช่องว่าง ภายใต้อี e3 ยังมี e6 ซึ่งมีเนื้อหาประกอบอยู่ด้วย และภายใต้อี e6 ยังมี e7 ที่มีเนื้อหาซ้อนอยู่อีกชั้น ซึ่งเป็นชั้นสุดท้ายของแท็กนี้ โดยการไล่ลำดับชั้นของข้อมูลจะเริ่มจาก e ตัวแรกเป็นต้นไป ตัวอย่างถ้าต้องการทราบค่า "Aom" ภายใต้อีลิเมนต์ใดบ้าง จะทำการอ่านเรียงค่า e ภายใต้อีแท็กไปเรื่อยๆจนถึงค่าของข้อมูลที่ต้องการทราบ ซึ่งจะได้ว่า "Aom" ภายใต้อี e1, e2, e3 และ e6 เรียงลำดับกันไปนั่นเองในกรณีที่มีข้อมูลมีความยาวหรือมีการปิดแท็กเกิดขึ้นในเอกสารจะเกิดแท็กใหม่และมี x เป็นตัวอ้างอิงลำดับชั้นของอีลิเมนต์ก่อนหน้าที่จะแสดงข้อมูลต่อท้าย เช่น `<xe1e2 e3a4="a2" e6v="Veerasuk" />` จะเห็นได้ว่า a2 ซึ่งเป็นค่าของข้อมูลใน a4 จะอยู่ภายใต้อี e1, e2 และ e3 ตามลำดับ ในขณะที่ Veerasuk ซึ่งเป็นค่าของข้อมูลใน e6 จะอยู่ภายใต้อี e1, e2 และ e6 ตามลำดับ และมีข้อสังเกตว่า ถ้าเป็นแท็ก `<e1e2e3 a4="a1" .../>` e3 กับ a4 จะอยู่แยกกัน แต่ถ้ามีการอ้างอิงถึงอีลิเมนต์ `<e1e2e3 ...>` ก่อนหน้านั้นจะทำให้แท็กของ อีลิเมนต์นี้เขียนอยู่ในรูปแบบ `<xe1e2 e3a4="a2" .../>` ซึ่ง e3 กับ a4 จะอยู่ติดกัน

จะเห็นได้ว่าโครงสร้างยังคงเป็นเอกสาร XML จากการบีบอัดด้วยวิธีการนี้เห็นได้ว่า เอกสารใหม่มีความกระชับและขนาดเล็กกว่าต้นฉบับ อีกทั้งยังเป็นเอกสารที่สามารถอ่านเข้าใจได้ง่าย และยังคงอยู่ในรูปแบบของเอกสาร XML ที่ถูกต้องตามหลักด้วย (Well-formed)

ในส่วนของเอกสารที่มีอีลิเมนต์แบบ mixed-content นั้นสามารถใช้ XBrevity ในการบีบอัดได้เช่นกัน โดยมีตัวอย่างของเอกสารดังกล่าวที่อยู่ในรูปที่ 5 ซึ่งเป็นตัวอย่างของเอกสารที่มีอีลิเมนต์แบบ mixed-content โดยในเอกสารมี `<mtitle>`

เป็นอีลิเมนต์แบบ mixed-content ซึ่งมีทั้งอีลิเมนต์ `<mname>` และเนื้อหา (content) ซึ่งมีค่าเป็น "the prisoner of" และ "redemption" นอกจากนี้ในเอกสารตัวอย่างในรูปที่ 5 ยังแสดงให้เห็นถึงตัวอย่างของเอกสารที่อีลิเมนต์ที่ซ้อนกัน ในเอกสารนี้มีอีลิเมนต์ actor ซึ่งซ้อนกันอยู่หลายชั้น โดยที่แต่ละ actor มีเนื้อหาที่แตกต่างกันไป

```
<?xml version="1.0"?>
<movie>
  <mtitle year="1994">the prisoner of
    <mname>shawshank</mname>redemption</mtitle>
    <actor>Tim Robbins <actor>Morgan Freeman
    <actor>Bob Gunton </actor>William Sadler
    <actor>Clancy Brown </actor>Gil Bellows</actor>
  </actor>
</movie>
```

รูปที่ 5 เอกสาร XML ตัวอย่าง B

```
<c>
  <d>
    <e1 a2="1994" v="the prisoner of"
      e3v="shawshank"/> <xe1 v="redemption"/>
    <e4 v="Tim Robbins" e4v="Morgan Freeman"
      e4v2=" Bob Gunton" />
    <xe4e4 v="William Sadler" e4v="Clancy Brown"/>
    <xe4e4 v="Gil Bellows" />
  </d>
  <m f="movie mtitle year mname actor"
    b="0 1 a2 3 4 "/>
</c>
```

รูปที่ 6 เอกสาร XML ตัวอย่าง B ที่ถูกบีบอัดแล้ว

เมื่อมีเอกสารที่มีอีลิเมนต์ที่เป็นแบบ mixed-content เกิดขึ้น XBrevity จะพิจารณาบีบอัดเอกสารดังกล่าว ถ้าพบแท็กปิดเกิดขึ้นในตัวอย่างนี้คือ `</mname>` ซึ่งจะได้ `<e1 a2="1994"`

$v = \text{"the prisoner of"}$   $e3v = \text{"shawshank"}$  จะจัดให้อยู่ในอิลิเมนต์เดียวกัน ส่วน "redemption" จะแยกเป็นอิลิเมนต์ใหม่โดยภายในจะแทนด้วย  $x$  เพื่อบ่งบอกว่าเป็นเนื้อหาที่อยู่ในอิลิเมนต์เดียวกันกับอิลิเมนต์อ้างอิงก่อนหน้านี้ซึ่งก็คือ  $\langle mtitle \rangle$  นั่นเองในส่วนแท็ก  $\langle actor \rangle$  ที่มีชื่ออิลิเมนต์เดียวกันและมีจำนวนชั้นซ้อนกันตั้งแต่ 3 ชั้นขึ้นไป จะแยกเป็น 1 อิลิเมนต์ คือ  $e4$  และมี 3 แอตทริบิวต์ คือ  $v$  ซึ่งเป็นค่าของ actor ตัวแรก ตัวถัดมาคือ  $e4v$  เป็นค่าของ actor ตัวที่สอง และ  $e4v2$  เป็นค่าของ actor ตัวสุดท้าย จะสังเกตได้ว่าเมื่อมีอิลิเมนต์ซ้อนกันหลายตัวและมีชื่อเดียวกัน XBrevity จะทำการกำหนดค่าตัวเลขให้ตั้งแต่การซ้อนกัน 3 ชั้นขึ้นไป โดยเริ่มใส่ตัวเลข 2 เป็นต้นไป ผลที่ได้เป็นไปตามรูปที่ 6 จะเห็นได้ว่าแม้เอกสารจะมีอิลิเมนต์แบบ mixed-content ก็สามารถบีบอัดด้วย XBrevity ได้เช่นกัน

#### 4. การทดลอง

ในการทดลองการบีบอัดข้อมูลนี้ ได้นำเอกสาร XML ตัวอย่างโดยเป็นเอกสารที่ถูกสร้างขึ้นมาจากที่ต่างๆ ไม่ว่าจะได้จากแหล่งดาวน์โหลดต่างๆ หรืออาจจะใช้ XMark [12] ในการสร้างเอกสาร .xml ขึ้นมาก็ได้ ซึ่งขนาดเอกสารที่นำมาทดสอบจะมีค่าต่างๆกัน และนำเอกสารที่ได้ขึ้นไปทดสอบกับ gzip และ XMill เพื่อเปรียบเทียบประสิทธิภาพการบีบอัดข้อมูล และพัฒนาระบบการบีบอัด โดยการทดลองนี้ได้ทำบน Intel Pentium4 1.80GHz หน่วยความจำ 256 MB ระบบปฏิบัติการ Windows XP Professional with Service Pack 2 และใช้ภาษาจาวา (JDK 5.0 Update 4) ในการสร้างและพัฒนาระบบการบีบอัด แต่ในการทดลองนี้ไม่สามารถทดสอบร่วมกับ XGrind, XPRESS และ XPACK เนื่องจาก XGrind เป็นโปรแกรมที่ใช้ทดสอบบนระบบปฏิบัติการ Linux หรือบนระบบ Unix เท่านั้น ส่วน XPRESS และ XPACK ไม่มีโปรแกรมให้ดาวน์โหลดเพื่อใช้ทดสอบ จึงไม่สามารถนำมาใช้ทดสอบเพื่อเปรียบเทียบประสิทธิภาพกับตัวบีบอัด (Compressor) ตัวอื่นๆ ได้ ดังนั้นในการทดลองนี้จึงทำการเปรียบเทียบการบีบอัดด้วย gzip, XMill และ XBrevity

##### 4.1 การวัดประสิทธิภาพของการบีบอัด

ประสิทธิภาพของการบีบอัดวัดจากอัตราส่วนในการบีบอัดซึ่งหาได้จากสมการที่ (1)

$$\text{อัตราส่วน} = 1 - \frac{\text{ขนาดของเอกสารที่บีบอัดแล้ว}}{\text{ขนาดของเอกสารต้นฉบับ}} \quad (1)$$

ในการทดลองนี้ได้ใช้ XMark สร้างเอกสาร XML ขึ้นมาทั้งหมด 5 เอกสาร โดยมีขนาดของเอกสารที่แตกต่างกันขึ้นอยู่กับค่าแฟกเตอร์ (XMark factor) ที่ใช้ในการกำหนดขนาดของเอกสารที่จะสร้างขึ้นมา ซึ่งแสดงคุณลักษณะของเอกสารดังตารางที่ 1

ตารางที่ 1 คุณลักษณะของเอกสารที่ใช้ในการทดสอบ

XML Files	Size (Bytes)	Depth	Elem	Attr	XMark factor
a.xml	27,233	8	58	3	0.0001
b.xml	118,274	9	72	4	0.001
c.xml	1,182,547	10	72	5	0.01
d.xml	11,875,066	10	72	5	0.1
e.xml	118,552,713	10	72	5	1

จากข้อมูลในตารางที่ 1 a.xml, b.xml, c.xml, d.xml และ e.xml เป็นชื่ออ้างอิงถึงเอกสาร XML ที่ใช้ในการทดสอบ นอกจากนี้ในตารางที่ 1 ยังมีข้อมูลซึ่งบ่งบอกชนิดของข้อมูลซึ่งได้แก่

*Size* คือขนาดของเอกสาร XML มีหน่วยเป็นไบต์ (bytes)

*Depth* คือความลึกหรือจำนวนชั้นที่ซ้อนกันสูงสุดที่เกิดขึ้น

ในเอกสาร

*Elem* คือจำนวนอิลิเมนต์ (Element types) ในเอกสาร

*Attr* คือจำนวนแอตทริบิวต์ (Attribute types) ในเอกสาร

*XMark factor* คือค่าแฟกเตอร์ที่ใช้กำหนดขนาดในการสร้างเอกสารตัวอย่าง

##### 4.2 ผลการทดลอง

เมื่อทำการบีบอัดในแต่ละวิธีแล้ว ผลที่ได้ปรากฏอยู่ในตารางที่ 2 โดยวัดจากขนาดที่บีบอัดได้ซึ่งมีหน่วยเป็นกิโลไบต์ (Kbyte) ขนาดที่บีบอัดด้วย gzip และ XMill มีขนาดเล็กกว่าเมื่อเทียบกับ XBrevity เพียงอย่างเดียวเนื่องจากว่า gzip และ XMill เป็นการบีบอัดในรูปแบบของไบนารี แต่ XBrevity ทำการบีบอัดเฉพาะส่วนของแท็กเท่านั้น ซึ่งผู้วิจัยกำลังพัฒนาเพิ่มเติมเพื่อให้

บีบอัดส่วนที่เป็นข้อความได้ด้วย แต่อย่างไรก็ตามเมื่อนำเอา XBrevity ที่ได้ทำการจัดเรียงข้อมูลแบบกระชับแล้ว มาใช้ร่วมกับ gzip จะได้ผลการทดลองที่แตกต่างออกไป ดังแสดงในตารางที่ 2

ตารางที่ 2 การเปรียบเทียบการบีบอัด

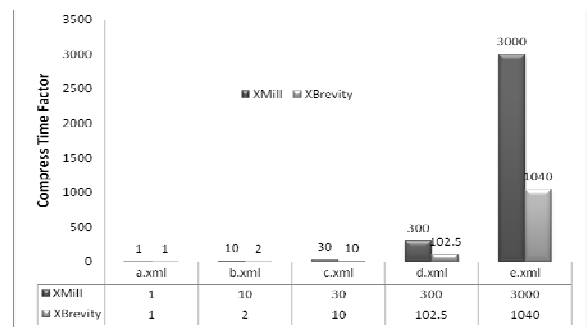
XML Files	Compressed File Size (KBytes)			
	gzip	XMill	XBrevity	XBrevity+gzip
a.xml	10.25	10.15	25.96	10.55
b.xml	39.79	37.91	113.3	40.56
c.xml	371.5	338.1	1,130	374.9
d.xml	3,732	3,320	11,359	3,760
e.xml	37,358	32,996	112,989	37,949

จากการนำผลการบีบอัดที่ได้จาก XBrevity มารวมกับ gzip ทำให้ประสิทธิภาพในการบีบอัดเอกสาร ทำได้ดีขึ้นกว่าการใช้ XBrevity บีบอัดเพียงอย่างเดียว เนื่องจากว่าเมื่อ เอกสารที่ผ่านการจัดเรียงแบบกระชับด้วย XBrevity แล้วนั้นขนาดของเอกสารจะเล็กกว่าต้นฉบับเดิมและเมื่อรวมกับ gzip จึงทำให้ขนาดที่บีบอัดได้ใกล้เคียงกับการบีบอัดด้วย gzip เพียงอย่างเดียว แต่เมื่อเทียบกับ XMill แล้วเอกสารยังมีขนาดใหญ่กว่าเล็กน้อย เมื่อขนาดของ XBrevity ที่รวมกับ gzip มีขนาดเล็กลงกว่าการใช้ XBrevity เพียงอย่างเดียวทำให้บีบอัดได้มากขึ้นตามไปด้วย อย่างไรก็ตามการรวมกันดังกล่าวไม่สามารถทำการ Query ได้ เนื่องจาก XBrevity ที่รวมกับ gzip จะมีคุณสมบัติเหมือน gzip นั่นเอง

เนื่องจากในการพัฒนาโปรแกรม XBrevity ได้ใช้ภาษาจาวา (Java language) เป็นเครื่องมือในการสร้างโปรแกรมเพื่อทดสอบเทคนิคนี้ โดยมีขั้นตอนของการอ่านไฟล์และแยกส่วนของข้อมูลออกเป็นอีลิเมนต์ แอตทริบิวต์ และเนื้อหาของเอกสาร เมื่ออ่านเอกสารในแต่ละส่วนแล้วจะต้องเก็บค่าลงใน Hashtable ก่อนแล้วจึงเรียกข้อมูลมาทำการเขียนลงไปบนไฟล์ โดยในการเขียนลงไฟล์นั้นต้องทำการตรวจสอบให้เป็นไปตามแบบฟอร์มของภาษา XML อีกด้วย ซึ่งกระบวนการต่างๆ ดังกล่าวมีหลายขั้นตอน ทำให้เวลาที่ใช้บีบอัดมากตามไปด้วย แต่ XMill ใช้ภาษาซีพลัสพลัส (C++ language) ในการพัฒนา

โดยเมื่ออ่านเอกสารและทำการคัดแยกแล้วจะทำการเข้ารหัสซึ่งเป็นแบบไบนารีไปจนเสร็จกระบวนการ และ gzip ใช้ภาษาซี (C language) ในการพัฒนาโดยอ่านไฟล์โดยไม่มีการแยกอีลิเมนต์ใดๆแล้วเข้ารหัสแบบไบนารีไปจนเสร็จกระบวนการ ซึ่งไฟล์ที่ได้จากการบีบอัดของ XBrevity เป็นรูปแบบที่แตกต่างกับ XMill และ gzip กล่าวคือ XBrevity ยังเป็นไฟล์ XML อยู่สามารถนำไปใช้ได้โดยไม่ต้องขยาย ส่วน XMill และ gzip ต้องขยายการบีบอัดก่อนจึงจะสามารถใช้งานเอกสารนั้นได้

ด้วยความแตกต่างของโปรแกรมภาษาที่ใช้ในการพัฒนา จึงเปรียบเทียบเวลาในการบีบอัดด้วยการเทียบกับไฟล์อ้างอิง ซึ่งในที่นี้ได้ใช้ a.xml เป็นไฟล์อ้างอิงกับไฟล์อื่นๆ เมื่อทำการเปรียบเทียบประสิทธิภาพในด้านของเวลา ปรากฏว่า XBrevity ใช้เวลาได้น้อยกว่า XMill เมื่อเทียบกับด้วยไฟล์อ้างอิง โดยการทดลองนี้จะไม่ใช้กับ gzip เนื่องจากไม่ได้ใช้บีบอัด ไฟล์ที่เป็น XML โดยเฉพาะ ดังผลที่แสดงไว้ในรูปที่ 7



รูปที่ 7 กราฟแสดงการเทียบเวลาที่ใช้ในการบีบอัด

ในกระบวนการขยายการบีบอัดนั้น จะนำไฟล์ที่ถูกบีบอัดมาทำการขยายออกเป็นเอกสารต้นฉบับ ผลที่ได้เป็นเอกสารต้นฉบับที่มีความสมบูรณ์และถูกต้องตามหลักของภาษา XML และเป็นไปในทิศทางเดียวกันทุกเอกสารที่มีการบีบอัด แต่การขยายเอกสารจะทำได้เร็วกว่าการบีบอัด และในส่วน XBrevity ที่ใช้การบีบอัดรวมกับ gzip มีคุณสมบัติเดียวกันกับ gzip และจะต้องทำการ gunzip ก่อนจึงจะได้เอกสารในรูปแบบที่ถูกบีบอัดด้วย XBrevity

จากผลการทดลองที่ได้และแสดงไว้ในตารางที่ 2 นั้น จะเห็นได้ว่าเปรียบเทียบอัตราส่วนของการบีบอัดแล้ว XMill และ gzip มีค่าค่อนข้างสูงเมื่อเทียบกับการใช้ XBrevity เพียงอย่างเดียวเพราะ XBrevity บีบอัดเฉพาะส่วนของแท็กเท่านั้น แต่



อย่างไรก็ตามถ้า XBrevery ที่ได้ทำการบีบอัดแล้วมาใช้ร่วมกับ gzip จะเห็นว่ามีความยาวของข้อมูลที่ลดลงมาก่อนข้างมาก แต่ไม่สามารถ Query ได้เพราะใช้ gzip ในการบีบอัดในขั้นสุดท้ายจึงมีคุณสมบัติของไฟล์เป็น gzip และผลที่ได้มีความใกล้เคียงกับการใช้ gzip และ XMill โดยเวลาที่ใช้ในการขยายไฟล์ออกเป็นเอกสารต้นฉบับเดิมนั้น น้อยกว่าการบีบอัดเอกสาร ไม่ว่าจะใช้วิธีการใด แต่เนื่องด้วยความแตกต่างทางโปรแกรมภาษา ทำให้ XBrevery เสียเปรียบในเรื่องนี้ แต่เมื่อเปรียบเทียบเวลาที่ใช้ในการบีบอัด โดยเทียบจากไฟล์อ้างอิงแล้วทำการวัดเวลาการบีบอัดด้วยโปรแกรมภาษานั้นๆ ปรากฏว่า XBrevery สามารถทำได้ดีกว่า XMill เนื่องจาก XBrevery ทำการบีบอัดเฉพาะในส่วนของแต่ละ แต่ XMill บีบอัดในส่วนเนื้อหาของอิลิเมนต์ (element content) ด้วย ซึ่งผู้ใช้ปลายทางต้องคลายการบีบอัดของเนื้อหาของอิลิเมนต์ และเป็นเหตุให้ต้องใช้เวลาในการขยายข้อมูลเพิ่มขึ้นตามไปด้วย

## 5. สรุป

การบีบอัดข้อมูล XML มีความสำคัญเพราะ XML เป็นภาษาที่มีข้อดีอยู่หลายประการ ซึ่งปัจจุบันได้มีการใช้กันอย่างแพร่หลาย แต่ข้อจำกัดของ XML ก็มีเช่นกัน โดยปกติแล้วเอกสารภาษา XML จะมีขนาดใหญ่กว่าเอกสารของเท็กซ์ไฟล์ (Text File) ทั่วไป เนื่องจากเอกสารภาษา XML มักจะมีการใช้แท็กกำกับความหมายของข้อมูล การส่งข้อมูลโดยใช้ภาษา XML ทำให้เกิดความต้องการเน็ตเวิร์กแบนด์วิดท์ (Network Bandwidth) ขนาดใหญ่เนื่องจากขนาดของไฟล์ วิธีหนึ่งที่จะช่วยลดเน็ตเวิร์กแบนด์วิดท์ในการส่งข้อมูล ก็คือการบีบอัดข้อมูลให้มีขนาดเล็กลง อีกทั้งยังช่วยให้ใช้พื้นที่ในการเก็บข้อมูลลดลงตามไปด้วย จากงานวิจัยที่ผ่านมาเกี่ยวกับการบีบอัดข้อมูล XML นั้น ได้ทำการบีบอัดข้อมูลให้มีขนาดเล็กลง ซึ่งในบางวิธีต้องใช้โครงสร้างของภาษา (XML Schema) เป็นตัวกำหนดไวยากรณ์เป็นของตัวเอง แต่ว่าเอกสารเอ็กซ์เอ็มแอลบางเอกสารก็ไม่มีเอกสารอธิบายโครงสร้างของเอกสาร ซึ่งหากใช้วิธีนั้น จะต้องมีการสร้างเอกสารอธิบายโครงสร้างของเอกสาร ทำให้ยุ่งยากต่อการนำไปใช้งานต่างๆไป

เทคนิคที่ได้นำเสนอคือการบีบอัดข้อมูล XML โดยใช้วิธีจัดเรียงข้อมูลแบบกระชับ (XBrevery) ซึ่งเป็นเทคนิคการ

จัดเรียงแบบกระชับ เพื่อให้เอกสารมีขนาดเล็กลงจากเดิม และยังสามารถเข้าใจความหมายเอกสารที่ถูกบีบอัดได้ง่าย โดยไม่ต้องมีการขยายเอกสารให้เป็นต้นฉบับเดิม เพราะอยู่ในรูปแบบของภาษา XML อยู่แล้ว ทำให้ยังคงข้อดีที่มีอยู่ในภาษา XML อีกทั้งยังง่ายต่อการนำไปใช้งานต่างๆไป ซึ่งไม่ต้องมีตัวกำหนดไวยากรณ์แบบใดแบบหนึ่งโดยเฉพาะ เพื่อที่จะได้ไม่เสียเวลาในการขยายข้อมูลอีก หรือหากต้องการขยายข้อมูลก็สามารถทำได้เช่นกัน

## 6. แนวทางการพัฒนาในอนาคต

นอกจากงานวิจัย XBrevery ที่ได้นำเสนอไป ซึ่งเน้นถึงความสำคัญในการบีบอัดข้อมูลในส่วนโครงสร้างแท็ก แต่ตัวอักษรซึ่งเป็นข้อมูลภายในแท็ก (content) ยังไม่ได้รับการบีบอัด ดังนั้นจึงยังสามารถพัฒนาประสิทธิภาพของเทคนิคการบีบอัดได้มากขึ้นไปอีกโดยการบีบอัดอักขระภายในแท็ก ซึ่งจะมีผลให้ขนาดของเอกสารลดลงได้อีก นอกจากนี้แนวทางและวิธีการนำเอาเอกสารที่ถูกบีบอัดแล้วซึ่งอยู่ในรูปแบบ XML ไปใช้เพื่อการค้นหาข้อมูลก็เป็นงานวิจัยที่น่าสนใจ รวมถึงการพัฒนาให้โปรแกรมสามารถบีบอัดเพิ่มที่มีเอกสาร XML ได้ทั้งเพิ่มแทนที่จะบีบอัดที่ละไฟล์ และผลการทดลองจากการบีบอัดข้อมูล XML หากสามารถนำไปประยุกต์ใช้งานบนเว็บเซอร์วิส (Web Services) จะเป็นอีกงานวิจัยหนึ่งที่น่าสนใจ และเกิดประโยชน์ในอนาคต ในส่วนของการค้นหา (Query) ข้อมูลที่ถูกบีบอัดนั้นน่าจะสามารถทำได้ เนื่องจากข้อมูลที่ผ่านการบีบอัดแล้วนั้นยังอยู่ในรูปแบบของภาษา XML อยู่ เพียงแต่มีรูปแบบที่ต่างกันออกไป ซึ่งอาจจะต้องทำการประยุกต์ใช้กับ XQuery [17] ซึ่งเป็นข้อกำหนดในการค้นหาข้อมูลในภาษา XML

## เอกสารอ้างอิง

- [1] Amazon. "Amazon Web Services", Available at <http://www.amazon.com/gp/aws/landing.html>
- [2] BEA Systems, IBM, Microsoft, SAP AG and Siebel Systems. "Business Process Execution Language for Web Services", Available at <http://www-128.ibm.com/developerworks/library/specification/ws-bpel/>

- [3] eBay. “eBay API”, Available at <http://developer.ebay.com/common/api>
- [4] J. Ferraiolo, J. Fujisawa, D. Jackson. “Scalable Vector Graphics (SVG) 1.1 Specification”, Available at <http://www.w3.org/TR/SVG>
- [5] J.L. Gailly and M. Adler, “gzip : The compressor data”, Available at <http://www.gzip.org/>
- [6] Google. Google Web APIs (beta), Available at <http://www.google.com/apis/>
- [7] H. Liefke and D. Suciu. “XMill: an Efficient Compressor for XML Data.”, In *Proceeding of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 153-164, May 2000.
- [8] K. Mairiang, and C. Pluempitwiriyawej. “XPack: A Grammar-based XML Document Compression”, In *Proceeding of NCSEC2003 the 7<sup>th</sup> National Computer Science and Engineering Conference*, Oct 28-30, 2003.
- [9] J.-K. Min, M.-J. Park, and C.-W. Chung. “XPRESS: A Queriable Compression for XML Data.” In *Proceeding of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 122-133, June 9-12, 2003.
- [10] P. Murray-Rust and H. Rzepa. “Chemical Markup Language (CML)”, Available at <http://www.xml-cml.org>
- [11] K. Runapongsa and J. M. Patel, “Storing and Querying XML Data in Object-Relational DBMSs”, *EDBT Workshops 2002*: 266-285
- [12] A. R. Schmidt, F. Waas, M. L. Kersten, M. J. Carey, I. Manolescu, and R. Busse. “XMark: A Benchmark for XML Data Management”, *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pp. 974-985, Hong Kong, China, August 2002, Available at <http://monetdb.cwi.nl/xml/index.html>
- [13] P. M. Tolani and J. R. Haritsa. “XGRIND: A Query-friendly XML Compressor.” In *Proceedings of 18th International Conference on Databases Engineering*, Feb 2002.
- [14] UN/CEFACT and OASIS. “ebXML: Enabling a Global Electronic Market”, Available at <http://www.ebxml.org/>
- [15] W3C. “Extensible Markup Language (XML) 1.0 (Third Edition)”, Feb 4, 2004, Available at <http://www.w3.org/TR/2004/REC-xml-20040204/>.
- [16] W3C. “Web Services”, Available at <http://www.w3.org/2002/ws/>
- [17] W3C. “XQuery : An XML Query Language”, Available at <http://www.w3.org/XML/Query>
- [18] Yahoo. “Yahoo! Search Web Services”, Available at <http://developer.yahoo.net/>



**ประพันธ์ เลขาโสดณ** สำเร็จการศึกษา ระดับปริญญาตรี สาขาวิศวกรรมไฟฟ้า จากมหาวิทยาลัยเทคโนโลยีสุรนารีและ สำเร็จศึกษาในระดับปริญญาโท สาขา วิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น มีความสนใจ งานวิจัยด้าน XML Data Compression, Web Services, Computer Network and Security, and Parallel Computing.



**กานดา รุณนะพงศา** สำเร็จการศึกษา ระดับปริญญาเอก สาขาวิศวกรรมคอมพิวเตอร์และวิทยาการคอมพิวเตอร์ จาก The University of Michigan at Ann Arbor ประเทศสหรัฐอเมริกา ปัจจุบันเป็นอาจารย์ประจำภาควิชา วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ และเป็นรอง ผู้อำนวยการศูนย์คอมพิวเตอร์ มหาวิทยาลัยขอนแก่น มีความสนใจงานวิจัยด้าน XML Query Processing, Database Management Systems, Web Services, and Web Semantics.