

การออกแบบเมตาดาต้าสำหรับจัดการเนื้อหาเว็บไซต์  
WEB METADATA DESIGN FOR WEB CONTENT MANAGEMENT

นัทธร สีบุญ  
NATTHEPEON SUEB-AM

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาศาสตรมหาบัณฑิต

คณะวิทยาศาสตรมหาบัณฑิต

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2552

KMITL 2009-EN-M-260-088

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การออกแบบเว็บเมตาเดต้าสำหรับการบริหารจัดการเว็บไซต์

WEB METADATA DESIGN FOR WEB CONTENT MANAGEMENT

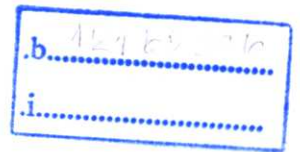


T105053

ณัฐพล สืบอ่ำ

NATTHAPHON SUEB-AM

เลขหมู่.....  
เลขทะเบียน..... 105053  
วัน,เดือน,ปี..... 1 2 พ.ย. 2552



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2552

KMITL-2009-EN-M-230-088

**WEB METADATA DESIGN FOR WEB CONTENT MANAGEMENT**

**NATTHAPHON SUEB-AM**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
MASTER OF ENGINEERING IN INFORMATION ENGINEERING  
FACULTY OF ENGINEERING  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG  
2009  
KMITL-2009-EN-M-230-088**


**COPYRIGHT 2009**

**FACULTY OF ENGINEERING**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การออกแบบเวปเมตาเดต้าสำหรับการบริหารจัดการเวปไซด์  
Thesis Title Web Metadata Design for Web Content Management  
นักศึกษา นายณัฐพล สืบอ่ำ  
รหัสประจำตัว 47061122  
ปริญญา วิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชา วิศวกรรมสารสนเทศ  
อาจารย์ที่ปรึกษาวิทยานิพนธ์ ผศ.มยุรี เลิศเวชกุล  
หมายเลขวิทยานิพนธ์ KMITL-2009-EN-M-230-088

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.ดร.พิทักษ์	ธรรมวาริน	
รศ.ดร.ชวลิต	เบญจางคประเสริฐ	
รศ.ดร.ปิติเขต	สุรักษา	
ดร.วีระพล	โมนยะกุล	
ผศ.มยุรี	เลิศเวชกุล	

วัน / เดือน / ปี ที่สอบ วันพฤหัสบดีที่ 28 พฤษภาคม พ.ศ. 2552 เวลา 12.30-14.30 น.

สถานที่สอบ ณ อาคาร A ชั้น 3 ห้องประชุม 2

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

คณะวิศวกรรมศาสตร์ รับรองแล้ว



(รองศาสตราจารย์ ดร.กอบชัย เดชหาญ)

คณบดี คณะวิศวกรรมศาสตร์

วันที่ 28 พฤษภาคม พ.ศ. 2552

หัวข้อวิทยานิพนธ์	การออกแบบเวปเมตาเดต้าสำหรับการบริหารจัดการเวปไซต์
นักศึกษา	นายณัฐพล สืบอ่ำ
รหัสนักศึกษา	47061122
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
พ.ศ.	2552
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ.มยุรี เลิศเวชกุล

### บทคัดย่อ

ในปัจจุบันเวปไซต์ถือเป็นสื่อที่ได้รับความนิยมอย่างสูงและมีผู้ใช้เป็นจำนวนมาก จากวัตถุประสงค์การใช้งานที่แตกต่างกันไปทำให้เกิดรูปแบบของการนำเสนอข้อมูลบนเวปไซต์ที่หลากหลาย ไม่ว่าจะเป็น ข่าวสาร รูปภาพ ภาพเคลื่อนไหว หรือ วิดีโอ โดยที่ส่วนใหญ่ผู้ดูแลเวปไซต์นั้นมักจะใช้โปรแกรมสำหรับบริหารจัดการเวปไซต์สำเร็จรูปมาช่วยในการจัดการข้อมูล ซึ่งมีข้อจำกัดในการใช้งานทั้งในด้านการออกแบบเวปไซต์และการบริหารจัดการ ไม่ว่าจะเป็นการกำหนดเวลาเผยแพร่ข้อมูลบนเวปไซต์ในเวลาที่ต้องการ หรือในการตรวจสอบการแสดงผลบนเวปไซต์ในช่วงเวลาที่ต้องการตรวจสอบว่ามีการนำเสนอข้อมูลเป็นอย่างไร ผู้จัดทำจึงได้เสนอแนวทางในการพัฒนาระบบบริหารจัดการเวปไซต์โดยการออกแบบฐานข้อมูลเชิงเวลาในรูปเวปเมตาเดต้าเพื่อจัดเก็บ โครงสร้างของเวปไซต์และข้อมูลที่เสนอ จากการศึกษาความสัมพันธ์ของโครงสร้างทางภาษา HTML มาใช้ในการออกแบบระบบบริหารจัดการเวปไซต์ เพื่อช่วยในการบำรุงรักษาทั้งโครงสร้างและข้อมูลต่างๆของเวปไซต์ ซึ่งจะช่วยแก้ปัญหาในการตรวจสอบ การออกแบบ และการบริหารจัดการเชิงเวลาทั้งตั้งเวลาล่วงหน้า และย้อนกลับมาคูในช่วงเวลาที่ต้องการ ทั้งนี้เพื่ออำนวยความสะดวกให้ผู้ดูแลเวปไซต์ทำงานได้อย่างมีประสิทธิภาพ

<b>Thesis Title</b>	Web Metadata Design for Web Content Management
<b>Student</b>	Mr. Natthaphon Sueb-Am
<b>Student ID.</b>	47061122
<b>Degree</b>	Master of Engineering
<b>Program</b>	Information Engineering
<b>Year</b>	2009
<b>Thesis Advisor</b>	Asst.Prof.Mayuree Lertwatechakul

## **ABSTRACT**

Nowadays, website is a popular media for representation of a variety of news, photos, video or animation. Because the information and the way to represent content on websites is usually updated dynamically, webmasters prefer to have a tool to manage their content and representing patterns. Most recent Web content management systems provide facility for web designing and current content management but not in term of timing control, which is important for today content management. In order to provide most features required for webmasters, we designed a web metadata data that capable to store both of content and web structure in form of relations of HTML structure. The web metadata is developed by using TNIAM. (Temporal Database using Nijssen's Information Analysis Methodology) This makes the web content management system to have ability of time scheduling control to present web content and web structure at the desired time. The developed web content management system is equipped with webpage analyzing engine that can import, parser and store the analyzed output into web metadata. By applying the analyzed information stored in web metadata, we can manage web content and its representation in advance. Moreover, the system is also useful to view the history of any websites captured into the web metadata.

## กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา ศศ. มยุรี เลิศเวชกุล ที่ให้ความช่วยเหลือ ให้คำชี้แนะช่วยแก้ปัญหาตลอดจนให้ความรู้และประสบการณ์ที่ดีแก่ข้าพเจ้า บริษัท พร็อพเพอร์ตี้ เพอร์เฟก (มหาชน) จำกัด และ บริษัท ทรอมัส ไอเดีย จำกัด ที่ให้โอกาสในการนำเสนอแนวคิดของวิทยานิพนธ์ชิ้นนี้ไปประยุกต์ใช้งานจริง ตลอดจนเพื่อนๆร่วมรุ่นที่ทำให้กำลังใจแล้วช่วยเหลือกันมาตลอด จนงานนี้ลุล่วงมาด้วยดี

สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดา มารดา น้องชาย และญาติที่เคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ และถ่ายทอดประสบการณ์ที่ดีแก่ข้าพเจ้า และผู้มีส่วนช่วยเหลืองานในครั้งนี้ทุกคน

ณัฐพล สืบอ่ำ

# สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	IX
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	2
1.3 สมมติฐานของการศึกษา.....	2
1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย.....	3
1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน.....	3
1.6 ขอบเขตการวิจัย.....	3
1.7 ขั้นตอนการศึกษา.....	3
บทที่ 2 ทฤษฎีพื้นฐานที่ใช้ในการวิจัย.....	5
2.1 รูปแบบการบริหารจัดการเว็บไซต์ในปัจจุบัน.....	5
2.1.1 การบริหารจัดการเว็บไซต์โดยทั่วไป.....	5
2.1.2 การพัฒนาเว็บไซต์ผ่านระบบการจัดการข้อมูล (Content Management System : CMS).....	6
2.1.2.1 osCommerce.....	8
2.1.2.2 XOOPS.....	9
2.1.2.3 MAMBO.....	10
2.1.2.4 DOTNETNUKE.....	11
2.2 งานวิจัยอื่นที่เกี่ยวข้อง.....	13
2.3 โครงสร้างทางภาษา HTML.....	16
2.3.1 ลักษณะทางภาษา HTML.....	16
2.3.2 โครงสร้างพื้นฐานของ HTML.....	17
2.3.2.1 ส่วนของ HEAD.....	17
2.3.2.2 ส่วนของ BODY.....	18

# สารบัญ (ต่อ)

หน้า

2.4 แบบจำลองข้อมูลแบบโนแอม (NIAM) และฐานข้อมูลเชิงเวลา TNIAM .....	18
2.4.1 ส่วนประกอบของโนแอมสกีมา (NIAM Schema) .....	18
2.4.2 ฐานข้อมูลเชิงเวลา TNIAMและการออกแบบ .....	20
2.4.2.1 Lifespan Time .....	20
2.4.2.2 Valid Time .....	21
บทที่ 3 การออกแบบฐานข้อมูลสำหรับเว็บเมตาเคด้าด้วย โมเดลโนแอม .....	22
3.1 สร้างเอนทิตีที่เกี่ยวข้อง .....	22
3.2 สร้างความสัมพันธ์ด้วย Element Fact Type .....	24
3.3 ออกแบบความสัมพันธ์ของโครงสร้างเว็บจากภาษา HTML ด้วยวิธีโนแอม .....	27
3.4 ทำการแมปปีงจากแบบจำลองโครงสร้างฐานข้อมูล .....	28
3.5 การแมปปีงฐานข้อมูลมาเป็นตาราง .....	29
3.6 ออกแบบโมเดลความสัมพันธ์ของโครงสร้างเว็บเมตาเคด้าเชิงเวลาด้วย TNIAM .....	31
3.7 ทำการแมปปีงแบบจำลองโครงสร้างฐานข้อมูลเชิงเวลาที่ปรับปรุงด้วย TNIAM .....	32
3.8 การแมปปีงฐานข้อมูลเชิงเวลาที่ปรับปรุงมาเป็นตาราง .....	33
บทที่ 4 การพัฒนาเว็บเมตาเคด้าเพื่อใช้ในการบริหารจัดการเว็บไซต์ .....	36
4.1 หลักการพัฒนาและออกแบบโปรแกรมประยุกต์ .....	36
4.1.1 ส่วนการนำเข้าข้อมูล (Import Function) .....	37
4.1.2 ส่วนการตัดคำ (Parser Function) .....	38
4.1.3 ส่วนการควบคุม (Control Function) .....	39
4.1.4 ส่วนการแสดงผล (Display Function) .....	39
4.1.5 การเตรียมฐานข้อมูลเพื่อรองรับระบบ .....	39
4.2 การเตรียมข้อมูลและทดลองการใช้งาน .....	40
4.3 การนำระบบมาประยุกต์ใช้งานจริง .....	41
บทที่ 5 ผลการทดลอง .....	46
5.1 การทดสอบความสามารถของฐานข้อมูลที่ออกแบบ .....	46
5.1.1 การบันทึก Source Code จากเว็บเพจลงฐานข้อมูล .....	46
5.1.2 ทดลองปรับเปลี่ยนค่าที่บันทึกในฐานข้อมูล .....	47
5.2 ทดสอบความสามารถของข้อมูลเมื่อเพิ่มความสามารถเชิงเวลา .....	50
5.2.1 ทดสอบการบันทึกฐานข้อมูลในรูปแบบเชิงเวลาในช่วงเวลาต่างๆ .....	50

# สารบัญ (ต่อ)

	หน้า
5.3 การทดลองการแสดงผลจากการใช้โปรแกรมต้นแบบ.....	52
5.3.1 การนำเวปเพจลงฐานข้อมูล.....	52
5.3.2 การนำเวปเพจต่างช่วงเวลาลงฐานข้อมูล.....	53
5.3.3 การแสดงผลเวปไซด์ผ่านจากโปรแกรมต้นแบบ.....	54
5.4 การนำฐานข้อมูลเชิงเวลาไปประยุกต์ใช้.....	57
5.4.1 การทดลองการแสดงผลในรูปแบบกำหนดการแสดงผลล่วงหน้า.....	57
5.4.2 การทดลองการแสดงผลในรูปแบบการดูย้อนหลังในช่วงเวลาที่ต้องการ.....	60
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ .....	62
บรรณานุกรม.....	64
ภาคผนวก.....	66
ภาคผนวก ก. ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่ .....	67
ภาคผนวก ข. มาตรฐาน HTML (HTML 4.01 Specification) .....	73
ประวัติผู้เขียน .....	93

# สารบัญตาราง

ตารางที่	หน้า
2.1 การเปรียบเทียบข้อดีข้อเสียของการสร้างเว็บไซต์โดยการจัดการในลักษณะต่าง ๆ .....	12
3.1 ค่าคุณสมบัติเอนติตี้ที่เกี่ยวข้องทั้งหมดในการออกแบบ.....	22
3.2 ค่า Elementary Fact Type แสดงความสัมพันธ์กันในแต่ละอีลีเมนต์ .....	24
3.3 URL .....	29
3.4 Content.....	29
3.5 Element_type - Attribute_Properties .....	29
3.6 Element - Attribute_Value.....	30
3.7 Element Relation .....	30
3.8 Element Type Relation .....	30
3.9 Element Properties.....	30
3.10 URL .....	33
3.11 Element_Type_Relation .....	33
3.12 Content.....	33
3.13 Element_type - Attribute_Properties .....	33
3.14 Element - Attribute_Value.....	34
3.14 Element Relation .....	34
3.14 Element Properties.....	34
4.1 การเตรียมข้อมูลพื้นฐานเพื่อนำเข้าข้อมูลในตาราง Element_Type_Relation .....	39
4.2 การเตรียมข้อมูลพื้นฐานเพื่อนำเข้าข้อมูลในตาราง Element_Attribute_Properties .....	39
4.3 การเตรียมข้อมูลพื้นฐานเพื่อนำเข้าข้อมูลในตาราง Element_Properties.....	39
5.1 ตัวอย่างการบันทึกข้อมูลลงในฐานข้อมูลที่ออกแบบไว้ใน ตาราง URL .....	45
5.2 ตัวอย่างการบันทึกข้อมูลลงในฐานข้อมูลที่ออกแบบไว้ใน ตาราง CONTENT .....	46
5.3 ตัวอย่างการบันทึกข้อมูลในฐานข้อมูลที่ออกแบบในตาราง Element_Attribute_Value .....	46
5.4 ข้อมูลที่บันทึกในฐานข้อมูลก่อนเปลี่ยนแปลงในตาราง Content.....	47
5.5 ข้อมูลที่บันทึกในฐานข้อมูลก่อนเปลี่ยนแปลงในตาราง Element_Attribute_Value .....	47
5.6 ข้อมูลที่ทำการทดลองเปลี่ยนแปลงในตาราง Content.....	48
5.7 ข้อมูลที่ทำการทดลองเปลี่ยนแปลงในตาราง Element_Attribute_Value .....	49
5.8 การบันทึกในฐานข้อมูลในตาราง Content ในรูปเชิงเวลา.....	50
5.9 การบันทึกในฐานข้อมูลในตาราง Element_Attribute_Value ในรูปเชิงเวลา .....	50

## สารบัญตาราง(ต่อ)

ตารางที่	หน้า
5.10 การบันทึกในฐานข้อมูลในตาราง Content ในรูปเชิงเวลาจากโปรแกรมต้นแบบ.....	52
5.11 การบันทึกในฐานข้อมูลในตาราง Element_Attribute_Value ในรูปเชิงเวลาจากโปรแกรมต้นแบบ .....	52
5.12 การบันทึกในฐานข้อมูลในตาราง Content ในรูปเชิงเวลาในเวลาต่างๆ .....	53
5.13 การบันทึกในฐานข้อมูลในตาราง Element_Attribute_Value ในรูปเชิงเวลาในเวลาต่างๆ.....	53

# สารบัญรูป

รูปที่	หน้า
2.1 ตัวอย่างเว็บไซต์แบบในกรณีแบบ Static.....	6
2.2 หลักการโดยทั่วไปของการจัดการเว็บไซต์ของ Web Content Management.....	7
2.3 ตัวอย่างเว็บไซต์ที่ใช้การบริหารจัดการเว็บไซต์ด้วย osCommerce .....	8
2.4 ตัวอย่างเว็บไซต์ที่ใช้การบริหารจัดการเว็บไซต์ด้วย XOOPS .....	9
2.5 ตัวอย่างเว็บไซต์ที่ใช้การบริหารจัดการเว็บไซต์ด้วย MAMBO.....	10
2.6 ตัวอย่างเว็บไซต์ที่ใช้การบริหารจัดการเว็บไซต์ด้วย DOTNETNUKE .....	11
2.7 ระบบที่ทำการแปลง XML มาใช้งานเป็น UI ในรูป HTML Form.....	14
2.8 การจัดเก็บเพื่อการแสดงผล UI ในรูป XML .....	15
2.9 โครงสร้างพื้นฐานทางภาษา HTML.....	17
2.10 สัญลักษณ์พื้นฐานของในแอมสกีมา .....	18
2.11 ความสัมพันธ์ของออปเจกต์ .....	19
2.12 ความสัมพันธ์แบบ 1:1, 1:N และ M : N.....	20
2.13 ตัวอย่างสัญลักษณ์ที่ใช้การออกแบบฐานข้อมูล แบบ LifeSpan Time .....	20
2.14 ตัวอย่างสัญลักษณ์ที่ใช้การออกแบบฐานข้อมูล แบบ Valid Time .....	21
3.1 ความสัมพันธ์ที่ออกแบบด้วยวิธีในแอม .....	27
3.2 ความสัมพันธ์ของฐานข้อมูลที่ออกแบบ .....	28
3.3 ความสัมพันธ์ของเว็บเมตาเดต้าในรูปแบบฐานข้อมูลเชิงเวลา TNIAM.....	31
3.4 ความสัมพันธ์ของฐานข้อมูลที่ได้จากรูปแบบเชิงเวลาของเว็บเมตาเดต้า .....	32
4.1 โปรแกรม Microsoft Access สำหรับบันทึก Source code ลงฐานข้อมูล.....	35
4.2 หลักการทำงานของระบบการบริหารจัดการเว็บไซต์จากเว็บเมตาเดต้า.....	36
4.3 หน้าจอสำหรับการทำงาน .....	36
4.4 หน้าจอสำหรับโปรแกรมในส่วนการนำเข้าข้อมูล .....	36
4.5 โฟลชาร์ตแสดงการทำงานของ Parser Function .....	37
4.6 หน้าจอสำหรับโปรแกรมในส่วนการแสดงผลข้อมูล .....	38
4.7 ความสัมพันธ์ของโครงสร้างทั้งหมดของเว็บไซต์ที่ทำมาปรับปรุง.....	40
4.8 หน้าเว็บไซต์ของบริษัทแห่งหนึ่งที่น่ามาใช้ในการปรับปรุง .....	41
4.9 การนำเสนอข้อมูลข่าวสาร และ โปร โโมชั่นในหน้าแรกของเว็บไซต์ที่ทำการศึกษา.....	41

## สารบัญรูป(ต่อ)

รูปที่	หน้า
4.10 โพล์ชาร์ตแสดงการทำงานการนำเสนอข้อมูลข่าวสารและ โปร โมชันในหน้าแรกของเวป ไซด์.....	42
4.11 หน้าจอส่วนโปรแกรมการจัดการกลุ่มข่าวสารและ โปร โมชันที่ต้องการแสดงผล .....	43
4.12 หน้าจอแสดงส่วนสถานะในการนำเสนอของกลุ่มข้อมูลข่าวสารและ โปร โมชัน .....	43
5.1 การเปรียบเทียบจากเวปต้นฉบับ และการสร้างเวปเพจจากฐานข้อมูล .....	45
5.2 การเปรียบเทียบจากเวปต้นฉบับ และการสร้างเวปเพจจากฐานข้อมูล .....	49
5.3 เวปเพจที่เกิดจากการสร้างจากฐานข้อมูลในวันที่ 15-06-2006 และวันที่ 20-01-2007 .....	50
5.4 เวปเพจที่เกิดจากการสร้างจากฐานข้อมูลในวันที่ 12-08-2006 .....	51
5.5 เวปเพจที่เกิดจากการสร้างจากฐานข้อมูลในวันที่ 01-01-2007 .....	51
5.6 ผลการนำเข้าไปในกลุ่ม 2009-01-20 Google Now .....	54
5.7 การทำงานเมื่อต้องการเลือกแสดงผลกลุ่มข้อมูลอื่นๆ .....	54
5.8 ผลการนำเข้าไปในกลุ่ม 2009-01-1 Google Newyear09.....	55
5.9 ผลการนำเข้าไปในกลุ่ม 2006-08-12 Google Mother Day.....	55
5.10 หน้าระบบแสดงสถานะของการแสดงผลของกลุ่มข้อมูลที่กำหนดให้แสดงในช่วงวันเวลาต่างๆ ในขณะนี้แสดงผลกลุ่มข้อมูลในวันที่ 22/12/2008 .....	56
5.11 กลุ่ม โปร โมชันที่ต้องการให้แสดงตั้งแต่วันที่ 22/12/2008.....	57
5.12 หน้าแรกของเวปไซด์เมื่อเวลาของเซิร์ฟเวอร์เป็นวันที่ 22/12/2008.....	57
5.13 กลุ่ม โปร โมชันที่ต้องการให้แสดงตั้งแต่วันที่ 1/2/2009.....	58
5.14 หน้าแรกของเวปไซด์เมื่อเวลาของเซิร์ฟเวอร์เป็นวันที่ 1/2/2009.....	58
5.15 หน้าระบบแสดงสถานะของการแสดงผลเมื่อเวลาของเซิร์ฟเวอร์แสดงผลวันที่ 1/2/2009 .....	59
5.16 หน้าแรกของเวปไซด์เมื่อเวลาของเซิร์ฟเวอร์เป็นวันที่ 29/1/2009.....	59
5.17 หน้าระบบแสดงสถานะของการแสดงผลเมื่อเวลาของเซิร์ฟเวอร์แสดงผลวันที่ 29/1/2009 .....	60
5.18 กลุ่ม โปร โมชันที่ต้องการให้แสดงตั้งแต่วันที่ 1/2/2009.....	60

ด้วยเหตุนี้จึงเกิดแนวคิดที่ว่าควรจะมีการพัฒนาโครงสร้างของการจัดการเว็บไซต์ เพื่อช่วยให้สามารถนำมาบริหารจัดการเว็บไซต์ได้ในหลากหลายประเภทภายใต้การจัดการโครงสร้างลักษณะเดียวกันและลดปัญหาอันเกิดมาจากความยุ่งยากในการบริหารจัดการ สังเกตเห็นว่าการแสดงผลของเว็บไซต์ต่าง ๆ จะถูกแสดงผลผ่านบราวเซอร์ในรูปแบบการแสดงผลผ่านภาษา HTML จึงเกิดแนวความคิดที่ว่าน่าจะนำโครงสร้างทางภาษาของ HTML มาจัดเก็บในรูปแบบเวปเมตาเดต้า ซึ่งสามารถจัดเก็บได้ทั้งส่วนของโครงสร้างเว็บไซต์ และส่วนของข้อมูลที่แสดงบนเว็บไซต์ แล้วนำโครงสร้างของเวปในรูปแบบเวปเมตาเดต้ามาประยุกต์ใช้ในการบริหารจัดการเว็บไซต์ รวมไปถึงการเพิ่มความสามารถต่าง ๆ เพื่อความสะดวกในการใช้งานในรูปแบบเชิงเวลา ตัวอย่างเช่น การย้อนกลับไปดูข้อมูลหรือหน้าเว็บไซต์ที่เราบริหารจัดการในช่วงเวลาที่ต้องการ หรือการวางแผนล่วงหน้าในการนำเสนอโปรโมชัน หรือข่าวสารต่าง ๆ ในเว็บไซต์ ซึ่งทางผู้จัดทำเห็นว่าจะจะเป็นประโยชน์สำหรับการบริหารเว็บไซต์เป็นอย่างมาก

ที่กล่าวมาข้างต้นคือที่มาและหลักการในการศึกษาการออกแบบเว็บไซต์ด้วยเมตาเดต้า ตามขอบเขตการศึกษาของวิทยานิพนธ์นี้ ซึ่งหวังว่าจะเป็นแนวคิดที่เป็นประโยชน์และเป็นแนวทางในการพัฒนาการสร้างระบบที่มีความสามารถในการบริหารจัดการเว็บไซต์ในอนาคตต่อไป ซึ่งนับเป็นสิ่งที่ทำหายอย่างยิ่ง

## 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์ฉบับนี้มุ่งหวังเพื่อศึกษาหาแนวทางการปรับปรุงการบริหารจัดการเว็บไซต์ เพื่อแก้ปัญหาในการจัดเก็บข้อมูลของเว็บไซต์ ที่ในปัจจุบันจัดเก็บฐานข้อมูลเฉพาะส่วนของข้อมูลที่อยู่ในเว็บไซต์แต่ไม่ได้จัดเก็บโครงสร้างของเว็บที่มี ทำให้ไม่สามารถนำหน้าเว็บในอดีตมาเปรียบเทียบกับปัจจุบันได้ รวมถึงการจัดการเว็บไซต์ในเชิงเวลาเพื่อการวางแผนในการกำหนดวันเผยแพร่ข้อมูล โดยอัตโนมัติ ดังนั้นในวิทยานิพนธ์นี้จึงเสนอการออกแบบเวปเมตาเดต้าสำหรับบริหารจัดการเว็บไซต์ โดยการออกแบบโครงสร้างฐานข้อมูลให้รองรับเวปเมตาเดต้าและโครงสร้างทางภาษา HTML เพื่อนำเสนอการบริหารเว็บไซต์ด้วยเวปเมตาเดต้า และประยุกต์การใช้งานในการจัดการเชิงเวลาของเว็บไซต์ทั้งในการย้อนกลับเพื่อดูข้อมูลในอดีตหรือวางแผนเพื่อกำหนดเวลาแสดงผลในอนาคต

## 1.3 สมมติฐานของการศึกษา

เว็บไซต์ต่าง ๆ เมื่อถูกนำเสนอผ่านเว็บเบราว์เซอร์จะถูกจัดการแสดงผลในรูปแบบภาษา HTML แสดงว่าเราสามารถนำโครงสร้างของ HTML มาออกแบบฐานข้อมูลเป็นเวปเมตาเดต้าเพื่อออกแบบระบบสำหรับการบริหารจัดการเว็บไซต์ได้ และจากปัญหาความต้องการวางแผนการนำเสนอข้อมูลผ่านทางเว็บไซต์ที่ต้องการความแน่นอนของเวลาในการเผยแพร่ รวมไปถึงการ

ย้อนไปดูข้อมูลในอดีตนั้น หากสามารถปรับฐานข้อมูลเชิงเวลามาประยุกต์ใช้น่าจะช่วยลดปัญหาที่ก่อให้เกิดความยุ่งยากในการจัดการ และรองรับการใช้งานในวัตถุประสงค์ดังกล่าวได้

#### 1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

ทฤษฎีหรือแนวคิดที่นำมาใช้ในการวิจัยวิทยานิพนธ์ฉบับนี้ คือทฤษฎีหรือแนวคิดที่เกี่ยวกับการนำฐานข้อมูลเชิงเวลามาประยุกต์ใช้ร่วมกับเมตาเดต้า และ โครงสร้างทางภาษา HTML ที่ใช้สำหรับการสร้างเว็บไซต์ มาออกแบบฐานข้อมูลเป็นเว็บเมตาเดต้าและพัฒนาร่วมกันเป็นระบบบริหารจัดการเว็บไซต์ที่มีประสิทธิภาพได้

#### 1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน

การออกแบบเว็บเมตาเดต้าสำหรับการบริหารจัดการเว็บไซต์นี้ เมื่อเปรียบเทียบหลักการกับวิธีพื้นฐานแล้วในส่วนของการจัดเก็บข้อมูลในฐานข้อมูลเพื่อการใช้งาน ในรูปแบบพื้นฐานจะจัดเก็บเฉพาะส่วนข้อมูลซึ่งแตกต่างจากวิธีที่นำเสนอ เนื่องจากจะจัดเก็บโครงสร้างของเว็บไซต์ในรูปแบบภาษา HTML ด้วย นอกจากนั้นยังมีความสามารถในการเชิงเวลาที่ไม่มีในรูปแบบพื้นฐานทั่วไป เพื่อเพิ่มประสิทธิภาพและลดปัญหาในการบริหารจัดการและเพิ่มความสามารถในการใช้งานให้มากขึ้น

#### 1.6 ขอบเขตการวิจัย

ในวิทยานิพนธ์ฉบับนี้ได้นำเสนอวิธีการออกแบบฐานข้อมูลให้อยู่ในรูปของเว็บเมตาเดต้า เพื่อรองรับการจัดเก็บข้อมูลในเว็บไซต์และ โครงสร้างของเว็บไซต์ ตามรูปแบบทางภาษา HTML และได้พัฒนารูปแบบของฐานข้อมูลให้อยู่ในรูปแบบของฐานข้อมูลเชิงเวลา เพื่อพัฒนาความสามารถของฐานข้อมูลให้รองรับการจัดเก็บข้อมูลและ โครงสร้างของเว็บไซต์ในรูปแบบภาษา HTML ในต่างเวลากันเพื่อนำมาประยุกต์ใช้งานเชิงเวลาทั้งการวางแผนล่วงหน้าในการใช้งาน และการย้อนกลับของข้อมูล ณ ช่วงเวลาที่ต้องการได้

#### 1.7 ขั้นตอนของการศึกษา

วิทยานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 6 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความจำเป็นมาของงานวิจัย ความมุ่งหมายและวัตถุประสงค์ สมมติฐาน ทฤษฎีที่ใช้ ขอบเขตของการวิจัย และขั้นตอนการศึกษา

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในการวิจัย และงานวิจัยที่เกี่ยวข้อง เวปเมตาเดต้า  
ฐานข้อมูลเชิงเวลาและการบริหารจัดการเวปในรูปแบบต่าง ๆ

บทที่ 3 กล่าวถึงการออกแบบฐานข้อมูลของเวปเมตาเดต้าด้วยวิธีในแอม

บทที่ 4 กล่าวถึงการพัฒนาเวปเมตาเดต้ามาใช้ในการบริหารจัดการเวปไซด์

บทที่ 5 แสดงผลการทดลองทั้งหมดของการศึกษานี้

บทที่ 6 เป็นบทสรุปผลการวิจัยและข้อเสนอแนะ

## บทที่ 2

# ทฤษฎีพื้นฐานที่ใช้ในการวิจัย

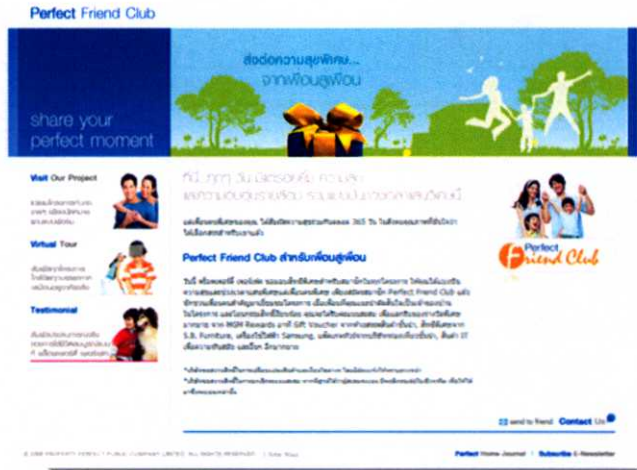
ในหัวข้อนี้จะกล่าวถึงงานวิจัยอื่นที่เกี่ยวข้อง รูปแบบของการบริหารจัดการเว็บไซต์ โดยทั่วไป ทฤษฎีพื้นฐานต่าง ๆ ที่เกี่ยวข้องในการวิจัย และรูปแบบฐานข้อมูลเชิงเวลาที่นำมาใช้ในการออกแบบ ซึ่งเนื้อหาทั้งหมดนี้จำเป็นสำหรับการศึกษา และนำมาใช้ในการออกแบบเวปเมตาเดต้าเพื่อนำมาใช้ในการบริหารเว็บไซต์ต่อไป

### 2.1 รูปแบบการบริหารจัดการเว็บไซต์ในปัจจุบัน

ในปัจจุบันวิธีการการออกแบบและบริหารจัดการเว็บไซต์มีอยู่มากมายขึ้นอยู่กับความถนัดของผู้ดูแลและตามประเภทรูปแบบของเว็บไซต์นั้น ๆ ซึ่งในที่นี้ขอจำแนกการบริหารจัดการเว็บไซต์ออกเป็น 2 ประเภทใหญ่ ๆ คือ การบริหารจัดการเว็บไซต์ทั่วไปและการบริหารจัดการเวปผ่านระบบระบบการจัดการข้อมูล

#### 2.1.1 การบริหารจัดการเว็บไซต์โดยทั่วไป

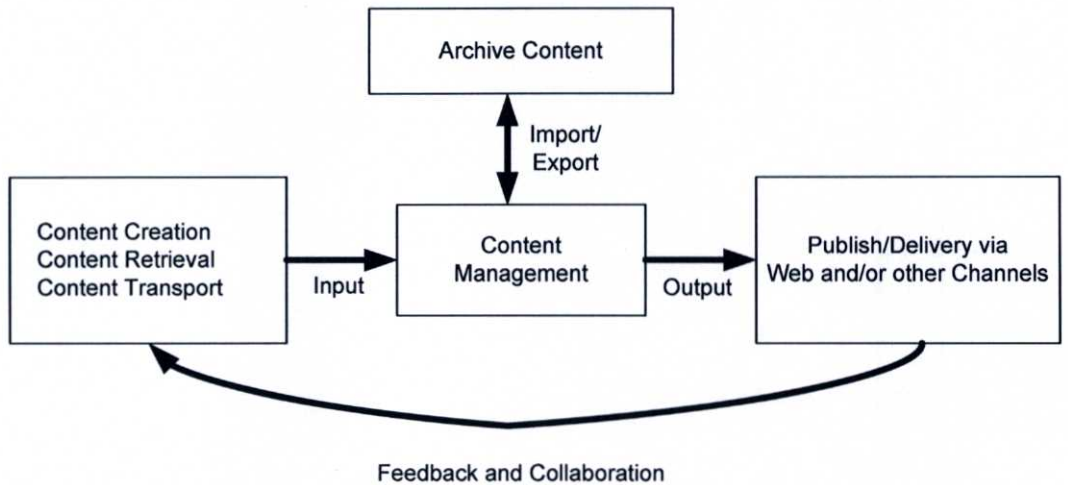
ซึ่งกรณีการพัฒนาเว็บไซต์แบบปกตินั้น สามารถเรียกเวปเพจประเภทนี้อีกแบบหนึ่งได้ว่า การบริหารจัดการเว็บไซต์ด้วยเวปเพจแบบคงที่ (Static) กล่าวคือจะมีการออกแบบหน้าเวปเพจลักษณะคล้ายการทำเอกสาร โดยสร้างรูปแบบ HTML โดยตรงเพื่อแสดงผลข้อมูลรูปแบบเดียวกับที่ได้ทำการออกแบบไว้ โดยทั่วไปจะเหมาะกับการใช้งานบนเว็บไซต์ที่มีเนื้อหาคงที่ไม่มีเปลี่ยนแปลงบ่อยนัก ไม่มีโครงสร้างยุ่งยากซับซ้อนเกินไป โดยถือว่ามีความสะดวกในการออกแบบ แต่ก็ยังมีปัญหาในการดูแลในภายหลัง กล่าวคือหากมีการปรับเปลี่ยนข้อความหรือมีการปรับปรุงเว็บไซต์ ก็จะต้องมีการเขียนหน้าเวปใหม่เพื่อนำมาแทนเวปเก่าอยู่เสมอ เช่น เมื่อต้องการจะให้หน้าเวปเปลี่ยนข้อมูลต่าง ๆ ไปตามเทศกาลก็ต้องทำการปรับเปลี่ยนหน้าเวปเป็นรายครั้งตามช่วงเวลาเทศกาลนั้น จะต้องทำการสร้างเวปเพจใหม่ขึ้นมา เพื่อนำไปทดแทนหน้าเวปเพจเดิมเอง ทำให้การดูแลเวปไม่ได้รับความสะดวก โดยเฉพาะถ้าผู้ดูแลคนเดียวต้องบริหารจัดการเว็บไซต์จำนวนมาก แต่ข้อดีของเวปประเภทนี้คือความอิสระในการจัดทำในการทำเว็บไซต์ตามที่ต้องการ ตัวอย่างเว็บไซต์ประเภทนี้โดยส่วนใหญ่จะเป็นเวปที่ให้ข้อมูลบริษัทต่าง ๆ ที่มีข้อมูลไม่เปลี่ยนแปลงบ่อยนัก เช่นรายชื่อผู้บริหาร ประวัติของบริษัท เป็นต้น



รูปที่ 2.1 ตัวอย่างเว็บไซต์ในแบบ Static

### 2.1.2 การพัฒนาเว็บไซต์ผ่านระบบการจัดการ (Content Management System : CMS)

ในปัจจุบันได้มีระบบที่เข้ามาช่วยในการบริหารจัดการเว็บไซต์ซึ่งเรียกว่า Content Management System : CMS ซึ่งการบริหารจัดการเว็บไซต์รูปแบบนี้ จะนำเอาการพัฒนาเว็บเพจประเภทปรับเปลี่ยนหรือไดนามิก (Dynamic) มาใช้ในการพัฒนา โดยในการใช้งานนั้นเมื่อผู้ดูแลต้องการเปลี่ยนแปลงเฉพาะข้อมูลในเว็บไซต์ โดยไม่ต้องการเปลี่ยนโครงสร้างหน้าตาของเว็บเดิม ผู้ดูแลไม่จำเป็นต้องสร้างเว็บเพจใหม่ไปแทน แต่เป็นการบริหารจัดการผ่านระบบโปรแกรมที่ใช้เพื่อปรับเปลี่ยนข้อมูลเฉพาะข้อมูลที่ต้องการเปลี่ยนเท่านั้น ซึ่งโปรแกรมบริหารจัดการเว็บไซต์ในปัจจุบันมีอยู่หลากหลายค่ายผู้ให้พัฒนา เช่น MAMBO, osCommerce, DOTNETNUKE, XOOPS เป็นต้น ซึ่งในแต่ละผู้พัฒนาก็จะมีความแตกต่างกันในเรื่องของรูปแบบการใช้งาน ภาษาที่ใช้ในการพัฒนา และจุดประสงค์การใช้งานเว็บไซต์ที่แตกต่างกัน แต่ในทุกๆระบบบริหารจัดการนั้นมีจุดประสงค์เดียวกันคือเพื่อปรับปรุงข้อมูลบนเว็บไซต์ ซึ่งสามารถแสดงหลักการทำงานของระบบการบริหารจัดการเว็บไซต์ได้จากรูปที่ 2.2



รูปที่ 2.2 หลักการโดยทั่วไปของการจัดการเว็บไซต์ของ Web Content Management [1]

ในการบริหารจัดการเว็บไซต์ด้วยการบริหารผ่านระบบการจัดการข้อมูลโดยการใช้รูปแบบการบริหารจัดการเว็บไซต์สำเร็จรูปนี้นิยมใช้กันเป็นส่วนใหญ่ เนื่องจากการพัฒนาเว็บไซต์ในรูปแบบนี้ผู้ใช้ไม่จำเป็นต้องลงมือเขียนสคริปต์ในการสร้างเว็บไซต์เองแต่อย่างใด ทำให้ไม่ต้องมีความรู้ในการสร้างเว็บไซต์ผ่านการเขียนเว็บด้วยภาษาต่าง ๆ เพียงแต่สามารถปรับเปลี่ยนเนื้อหาของและการจัดวางรูปแบบของข้อมูลในเว็บไซต์ตามที่ต้องการภายใต้ขอบเขตความสามารถของแต่ละโปรแกรมบริหารจัดการเว็บไซต์นั้น ๆ หรืออาจจะนำมาใช้สำหรับให้ลูกค้าสอบถามข้อมูลจากการเลือกรับข้อมูลในเว็บไซต์ เช่น การขอเรียกดูสินค้าในระบบซื้อขาย เป็นต้น ทั้งนี้หลักการโดยทั่วไปของระบบบริหารจัดการเว็บไซต์ทำได้ด้วยวิธีการจัดเก็บในฐานข้อมูลแล้วนำมาแสดงผล

อย่างไรก็ดีในปัจจุบันเว็บไซต์ส่วนใหญ่ที่ใช้การบริหารจัดการเว็บไซต์ผ่านระบบการบริหารจัดการในรูปแบบนี้ มักจะมีข้อจำกัดในบางประการ เช่น ความสวยงาม ความคล่องตัวในการปรับเปลี่ยน รูปแบบการใช้งาน เนื่องมาจากความไม่ยืดหยุ่นของระบบ เพราะมีความเป็นระบบสำเร็จรูปมากเกินไป ทำให้ผู้ใช้ต้องประยุกต์การใช้งานแบบผสมผสาน เช่น ข้อมูลใด ๆ ที่ต้องการความสวยงามและไม่มีปรับเปลี่ยนบ่อย ๆ ก็จะนิยมใช้หน้านั้นเป็นแบบคงที่ และในทางเดียวกันข้อมูลใด ๆ ที่มีการใช้งานแสดงผลในโครงสร้างเดิมแต่ต้องปรับเปลี่ยนข้อมูลได้หลากหลายจะใช้ระบบระบบบริหารจัดการเว็บไซต์มาเสริมแทน หรือในบางบริษัทพบว่าในมีการพัฒนาระบบบริหารจัดการเว็บไซต์มาใช้เอง เนื่องจากมีจุดประสงค์การใช้งานที่เป็นเอกลักษณ์เฉพาะของแต่ละบริษัท และให้การใช้งานตรงใจผู้ใช้งานมากที่สุด

ในการศึกษาครั้งนี้จะขอยกตัวอย่างของระบบบริหารจัดการเว็บไซต์ที่เป็นที่นิยมและเป็นที่ยอมรับโดยทั่วไปพอสังเขป โดยผู้อ่านอาจสามารถศึกษาเพิ่มเติมได้เองจากแหล่งค้นคว้าอื่น ๆ ต่อไป โดยในที่นี้จะกล่าวไว้เพื่อให้มีความเข้าใจเบื้องต้นในแต่ละประเภทดังต่อไปนี้

### 2.1.2.1 osCommerce [10],[11]



เป็นซอฟต์แวร์โอเพนซอร์ซที่ได้รับการพัฒนาจาก Harald Ponce de Leon และกลุ่มผู้พัฒนาชาวเยอรมันและอเมริกา โปรแกรมนี้ใช้เป็นเครื่องมือสำเร็จรูปในการจัดการและบริหารเว็บไซต์ เพื่อช่วยให้ผู้ที่ไม่มีความรู้ในการเขียนเว็บไซต์ประโยชน์ในการสร้างเว็บไซต์ได้ง่าย พัฒนาด้วยภาษา PHP ในรูปแบบ Object oriented backend (MS3) หรือการเขียนโปรแกรมเชิงวัตถุ โดยทำงานร่วมกับฐานข้อมูลแบบ MYSQL มีลักษณะเฉพาะในการใช้งาน ซึ่งสามารถนำมาใช้ประโยชน์ได้เพียงอย่างเดียวคือใช้ประโยชน์ในการทำเว็บไซต์ที่ต้องการมีระบบสำหรับการซื้อขายสินค้า แบ่งการใช้งานออกเป็น 2 ส่วนคือ หน้าเว็บที่เกิดขึ้นจริง (Front End) สำหรับผู้สนใจทั่วไปและส่วนของการบริหารจัดการ (Back End) สำหรับผู้ดูแลระบบ (ตัวอย่างในรูปที่ 2.3) นอกจากนี้ยังมีระบบรักษาความปลอดภัยแบบ SSL เพิ่มขึ้น และมีข้อจำกัดในการปรับเปลี่ยนรูปแบบโครงสร้างข้อมูลของเว็บไซต์ เนื่องจากต้องใช้รูปแบบโครงสร้างข้อมูลสำเร็จรูปของที่ทางเจ้าของโปรแกรมพัฒนามาเป็นทางเลือกให้ ทำให้ไม่สามารถปรับเปลี่ยนรูปแบบของเว็บไซต์ได้โดยอิสระ นอกจากนี้ยังสามารถเพิ่มเติมความสามารถต่าง ๆ ได้ในอนาคตในลักษณะของการเพิ่มโมดูลพิเศษต่าง ๆ เข้าไปอีกด้วย



หน้าเว็บ



ระบบจัดการ

รูปที่ 2.3 ตัวอย่างเว็บไซต์ที่ใช้การบริหารจัดการเว็บไซต์ด้วย osCommerce

### 2.1.2.2 XOOPS [10], [13], [16]



XOOPS เป็นโปรแกรมบริหารจัดการเว็บไซต์สำเร็จรูป ที่พัฒนาขึ้นมาจากภาษา PHP ในรูปแบบการพัฒนาแบบการเขียนโปรแกรมเชิงวัตถุซึ่งมีข้อดีในการประมวลผลที่รวดเร็วและทำงานได้พร้อมกันหลายอย่าง โดยทำงานร่วมกับฐานข้อมูลแบบ MySQL เช่นเดียวกับระบบอื่น ๆ นับเป็นโปรแกรมหนึ่งที่กำลังได้รับความนิยมมากขึ้นเรื่อย ๆ เนื่องจากมีความรวดเร็วในการใช้งาน และมีระบบจัดการที่ออกแบบหน้าจอบริการควบคุมที่เข้าใจได้ง่าย โปรแกรม XOOPS นี้ได้รับการพัฒนาจากทีมงาน XOOPS Foundation

การใช้งานของโปรแกรม XOOPS นั้นมีจุดประสงค์ในการใช้งานในลักษณะทั่วไป ไม่เฉพาะทางใดทางหนึ่งแล้วแต่ผู้ใช้จะเลือกโมดูลต่าง ๆ เข้าไปติดตั้งเพิ่มเติม เช่น กระดานข่าวสารโพล หรืออื่น ๆ ซึ่งสามารถเพิ่มเติมความสามารถของเว็บไซต์ได้ในอนาคตจากการติดตั้งโมดูลเพิ่มเติม การบริหารจัดการของ XOOPS ก็มีลักษณะเช่นเดียวกันกับระบบบริหารจัดการเว็บไซต์อื่นทั่วไป คือมีส่วนของหน้าเว็บที่เกิดขึ้นจริงและส่วนของการบริหารจัดการ (ตัวอย่างในรูปที่ 2.4) แต่ยังไม่สามารถปรับเปลี่ยนหน้าตาของเว็บไซต์ได้อิสระ ยังต้องใช้การสร้างโครงสร้างเว็บเพจและใช้งานผ่านโครงสร้างการจัดวางข้อมูลสำเร็จรูปที่นำมาติดตั้งเพิ่มเติม

การใช้งานในส่วนใหญ่ที่เหมาะสมกับระบบบริหารจัดการเว็บไซต์ในรูปแบบนี้ เหมาะสำหรับการนำมาใช้สำหรับเว็บไซต์ประเภทให้ข้อมูลทั่วไป หรือที่เรียกว่า Portal Web



หน้าเว็บ



ระบบจัดการ

รูปที่ 2.4 ตัวอย่างเว็บไซต์ที่ใช้การบริหารจัดการเว็บไซต์ด้วย XOOPS

### 2.1.2.3 MAMBO [10], [12], [15]



MAMBO เป็นโปรแกรมประเภทระบบสร้างและบริหารเว็บไซต์สำเร็จรูป พัฒนาจากภาษา PHP ซึ่งพัฒนาในรูปแบบโปรแกรมเชิงวัตถุ เช่นกันกับ XOOPS เป็นอีกระบบหนึ่งที่มีความนิยม โดยมีกลุ่มผู้ใช้ใกล้เคียงกับ XOOPS ลักษณะพิเศษของ MAMBO นั้นจะเน้นไปที่รูปแบบระบบการจัดการที่ใช้งานได้ง่าย และความสวยงามของระบบ โปรแกรม MAMBO นี้เดิมพัฒนาโดยบริษัท Micro Corporation ประเทศออสเตรเลีย ต่อมาได้รับการพัฒนาจากทีมงาน MAMBO Foundation แทน

รูปแบบการใช้งานของโปรแกรม MAMBO นั้นสามารถรองรับงานการบริหารจัดการเว็บไซต์ในทั่วไปได้อย่างหลากหลายตามโมดูลที่ผู้ดูแลเว็บไซต์ต้องการติดตั้ง และยังสามารถเลือกโมดูลต่าง ๆ ติดตั้งเพิ่มเติมได้ในอนาคต เช่น กระดานข่าวสาร โพล หรืออื่น ๆ และทำงานร่วมกับฐานข้อมูลแบบ MYSQL

นอกจากนี้การบริหารจัดการเว็บไซต์ของ MAMBO ก็มีลักษณะเช่นเดียวกันกับระบบบริหารจัดการเว็บไซต์ทั่วไปคือทำการปรับเปลี่ยนหน้าตาของเว็บไซต์อย่างจำกัด โดยต้องเลือกใช้โครงสร้างการจัดวางข้อมูลสำเร็จรูปจากการผู้พัฒนา และแสดงผลการทำงาน 2 ส่วน ในส่วนของหน้าเว็บที่เกิดขึ้นจริงและส่วนของการบริหารจัดการ เช่นเดียวกัน (ตัวอย่างในรูปที่ 2.5)



หน้าเว็บ



ระบบจัดการ

รูปที่ 2.5 ตัวอย่างเว็บไซต์ที่ใช้การบริหารจัดการเว็บไซต์ด้วย MAMBO

### 2.1.2.4 DOTNETNUKE [14]



DOTNETNUKE เป็นระบบบริหารจัดการเว็บไซต์อีกตัวหนึ่งที่น่าสนใจ โดยเฉพาะจุดเด่นปรับแต่งรูปแบบโครงสร้างของเว็บไซต์ และการเพิ่มเติมแก้ไขข้อมูลที่ทำให้ได้โดยง่าย โดยที่ส่วนของหน้าเว็บที่เกิดขึ้นจริงที่ใช้ในการแสดงผลจะมีรูปแบบที่ใกล้เคียงกับส่วนของการบริหารจัดการ (ตัวอย่างในรูปที่ 2.6) โดยสามารถทำการปรับปรุงข้อมูลในลักษณะแบบ In-Line Edit ไม่จำเป็นต้องใช้รูปแบบโครงสร้างของเว็บไซต์สำเร็จรูปในการปรับเปลี่ยน แต่ที่รองรับการใช้งานในส่วนนี้เช่นกัน นอกจากนี้ยังมีรูปแบบของการพัฒนาที่แตกต่างไปจากค่ายอื่น ๆ เนื่องจากพัฒนาในรูปแบบภาษา VB.Net แล้วมาแสดงผลใน ASP.net Framework โดยทีมพัฒนามาจาก DotNetNuke Corporation

เป็นที่น่าเสียดายที่ DOTNETNUKE ไม่เป็นที่นิยมเท่าที่ควรเนื่องมาจากการติดตั้งใช้งานจริงนั้นต้องอาศัยเทคโนโลยีของ Microsoft เป็นส่วนใหญ่ทำให้มีค่าใช้จ่ายในการใช้งานที่อาจไม่เกิดขึ้นหากเลือกใช้โปรแกรมบริหารจัดการเว็บไซต์ค่ายอื่น อันเนื่องมาจากส่วนของค่าลิขสิทธิ์ของ Operation System Windows ที่มารองรับติดตั้ง หรือฐานข้อมูล SQL Server ที่นำมาใช้ใน

นอกจากนี้ยังสามารถเพิ่มความสามารถต่าง ๆ ได้โดยง่ายจากการติดตั้งโมดูลที่ต้องการเพิ่มเติมเข้าไป เช่น กระดานข่าวสาร โพล หรืออื่น ๆ รวมไปถึงระบบซื้อขาย บนเว็บไซต์ก็สามารถทำได้เช่นเดียวกัน



หน้าเว็บ



ระบบจัดการ

รูปที่ 2.6 เว็บไซต์ที่ใช้การบริหารจัดการเว็บไซต์ด้วย DOTNETNUKE

โดยสรุปแล้วจะเห็นได้ว่าการพัฒนาเว็บไซต์มีอยู่หลายวิธีขึ้นอยู่กับความถนัด และ ชื่นชอบของแต่ละบุคคล ซึ่งก็มีข้อดีข้อเสียแตกต่างกันไป ไม่ว่าจะเป็นลักษณะการเขียนสร้างเว็บไซต์ในแบบการสร้างเว็บเพจเองหรือการใช้ระบบการบริหารจัดการเว็บไซต์แบบสำเร็จรูปเข้ามาเข้ามาช่วยในการจัดการ ซึ่งในแต่ละระบบการบริหารจัดการเว็บไซต์นั้น ก็มีรูปแบบที่ต่างกันออกไปขึ้นอยู่กับลักษณะการใช้งาน ซึ่งจะแสดงให้เห็นได้จากการจัดแบ่งระบบบริหารจัดการเว็บไซต์ต่าง ๆ ตามตารางที่ 2.1 แต่จะเห็นได้ว่าระบบบริหารจัดการข้อมูลในลักษณะนี้ จะมีรูปแบบที่เหมือนกันในส่วนของรูปแบบของระบบการบริหารจัดการเว็บไซต์ โดยประกอบด้วย 2 ส่วน คือ ส่วนของหน้าเว็บที่เกิดขึ้นจริง และส่วนของการบริหารจัดการ โดยทั้งหมดจะถูกจัดเก็บไว้ในฐานข้อมูลในรูปแบบการออกแบบที่แตกต่างกันออกไป แต่ใช้หลักการคล้าย ๆ กันในการออกแบบฐานข้อมูลคือมี 1 ตารางเพื่อจัดเก็บข้อมูลของแต่ละโมดูล และมีตารางที่ใช้ในการจัดการในเรื่องลำดับหรือจัดตำแหน่งในหน้าเว็บ เช่น ทำโพล ทำป้ายโฆษณา ทำกระดานข่าว เป็นต้น นอกจากนี้ยังมีการพัฒนาระบบเสริมในลักษณะโมดูลต่าง ๆ จากหลาย ๆ ช่องทางเพื่อเข้ามาช่วยในการบริหารหน้าเว็บให้ตรงกับความต้องการมากยิ่งขึ้น รวมไปถึงเพื่อเพิ่มสีสันและลูกเล่นต่าง ๆ ภายในเว็บ นอกจากนี้ยังสามารถปรับเปลี่ยนรูปแบบ โครงสร้างการจัดวางข้อมูลของเว็บไซต์ได้หลายวิธี และมี รูปแบบโครงสร้างการจัดวางมาตรฐานที่แต่ละผู้พัฒนาจัดเตรียมไว้ให้ เพื่อใช้หารดัดตั้งเพิ่มเติมในภายหลังมาใช้เพื่อปรับเปลี่ยนให้สวยงามยิ่งขึ้นอีกด้วย

ตารางที่ 2.1 เปรียบเทียบข้อดีข้อเสียของการสร้างเว็บไซต์โดยการจัดการในลักษณะต่าง ๆ

	Manual	osCommerce	XOOPS	MAMBO	DOTNETNUKE
รูปแบบการของโปรแกรม	HTML	PHP/Object oriented	PHP/Object oriented	PHP/Object oriented	VB.net/.Net Framework
ประเภทการใช้งาน	ทั่วไป	E-Commerce	Portal	Portal	Portal
การออกแบบเว็บไซต์	●	◐	◑	◑	◑
ระบบการจัดการเว็บ (ADMIN)		●	●	●	●
การปรับเปลี่ยนขนาด	●	◑	◑	◑	◑
การสำรองข้อมูล		●	●	●	●
การจัดเก็บในฐานข้อมูล		●	●	●	●
ระบบรายงานเชิงสถิติ		◑	◑	◑	◑
ระบบรักษาความปลอดภัย		●	◑	◑	◑
การวางแผนระบบในอนาคต					
การย้อนกลับข้อมูล					

จะเห็นได้จากการศึกษาระบบบริหารจัดการเว็บไซต์ในทั่วไปนั้นยังไม่สามารถแก้ไข ปัญหาในการจัดเก็บรูปแบบเว็บไซต์ทั้งในส่วน โครงสร้างของเว็บไซต์และข้อมูลของเว็บไซต์ ในการนำกลับมาแสดงผลต่างเวลากัน รวมไปถึงยังขาดการจัดการเชิงเวลาในการวางแผนล่วงหน้า เพื่อส่งข่าวสารต่าง ๆ โดยอัตโนมัติ และการย้อนกลับข้อมูลเพื่อดูการแสดงผลของเว็บไซต์ในอดีต ได้

ทางผู้วิจัยจึงทำการหาแนวทางในการแก้ปัญหาดังกล่าวต่อไป โดยการศึกษางานวิจัยของ ผู้อื่น เพื่อหาแนวทางที่น่าสนใจในการจัดการ ดังจะกล่าวถึงในหัวข้อถัดไป

## 2.2 งานวิจัยอื่นที่เกี่ยวข้อง

นอกจากการบริหารจัดการเว็บไซต์ด้วยวิธีการดูแลด้วยคนหรือการใช้โปรแกรมบริหารจัดการเว็บไซต์สำเร็จรูปที่กล่าวมาข้างต้นแล้วนั้น จากการค้นคว้าและศึกษาหาข้อมูลในงานวิจัย ต่าง ๆ พบว่ามีการนำเสนองานวิจัยเพื่อช่วยแก้ไขปัญหาดังกล่าวต่าง ๆ ในการจัดการบริหารเว็บไซต์ หลากหลายวิธีและมีจุดประสงค์ในการนำเสนอวิธีการแก้ปัญหาค้นคว้าที่แตกต่างกัน ในการค้นคว้าครั้งนี้ พบว่ามีงานวิจัยที่น่าสนใจและนำมาเป็นแนวทางในการศึกษาและพัฒนางานวิจัยต่อไปคืองานวิจัย ของ Mihaela Sokic นักวิจัยชาวโครเอเชีย จาก Croatian Telecom, Viktor Matic และ Alen Bazant จากคณะวิศวกรรมอิเล็กทรอนิกส์และคอมพิวเตอร์ มหาวิทยาลัย Zagreb ในปี 2003 [7] ได้นำเสนองานวิจัยเรื่องการบริหารจัดการเว็บไซต์โดยการนำ Native XML database [8] มาใช้ ร่วมกับฐานข้อมูล โดยจุดประสงค์เพื่อแก้ไขปัญหาค้นคว้าเรื่องความยุ่งยากในการบริหารจัดการในการดูแลเว็บไซต์ ในงานวิจัยได้นำเสนอวิธีการจัดการเว็บไซต์ในส่วนของคุณสมบัติโดยใช้เทคโนโลยี XML และภาษา JAVA ในการออกแบบงานวิจัยครั้งนี้

คณะผู้จัดทำได้นำเสนอฟังก์ชันการทำงานของงานวิจัยออกเป็น 4 ส่วนหลัก ๆ ดังต่อไปนี้

1. ส่วนการจัดการ โครงสร้างข้อมูล (Data Structuring)
2. ส่วนหน้าระบบจัดการเว็บไซต์ (UI)
3. ส่วนการจัดเก็บข้อมูล (Storing Data)
4. ส่วนการค้นหาค้นคว้าและการนำกลับมาแสดงผล (Search and retrieval )

กล่าวโดยรวมในการทำงานของฟังก์ชันต่าง ๆ ที่คณะผู้วิจัยได้นำเสนอในส่วนการจัดการ โครงสร้างข้อมูลถือเป็นกลไกในการจัดการโครงสร้างข้อมูล ซึ่งนับเป็นส่วนสำคัญสำหรับการ จัดการ ในงานวิจัยนี้ทางผู้ดูแลเว็บไซต์สามารถทำการสร้าง หรือแก้ไขโครงสร้างของการจัดวาง ในเว็บเพจได้เอง โดยการจัดการโครงสร้างของการจัดวางนั้นจะใช้หน้าระบบจัดการเว็บไซต์ในการจัดการ ซึ่งหน้าระบบจัดการเว็บไซต์นี้จะถูกสร้างโดยอ้างอิงมาจากโครงสร้างของการจัดวาง ข้อมูลในเว็บเพจนั้น ๆ โดยอัตโนมัติ กล่าวคือหากมีการเปลี่ยนแปลงโครงสร้างของข้อมูลในหน้า เว็บ หน้าระบบจัดการเว็บไซต์จะปรับเปลี่ยนไปโดยอัตโนมัติ รวมไปถึงโครงสร้างของฐานข้อมูล

ที่ทำการบันทึกข้อมูลผ่านหน้าระบบจัดการเว็บไซต์นั้น จะถูกรับปรุงให้รองรับโครงสร้างของข้อมูลที่ได้ปรับเปลี่ยนไปด้วย นอกจากนี้ก็จะมีระบบในการค้นหาข้อมูลและกลไกในการแสดงผลข้อมูลออกมา

ด้วยระบบนี้ทางคณะผู้วิจัยจึงนำเอาความสามารถของ XML เข้ามาใช้ในการแก้ปัญหาดังกล่าวโดยมองจุดประสงค์ว่าจะช่วยแก้ปัญหาการจัดการให้แก่ผู้ดูแลได้สะดวกมากยิ่งขึ้น

ทางคณะผู้วิจัยได้กล่าวไว้ว่าเนื่องจากการพัฒนาความสามารถของ XML จากเดิมต้องแปลงข้อมูลจากฐานข้อมูลมาอยู่ในรูปแบบของ XML จากนั้นจึงจะสามารถนำมาแสดงผลในรูปแบบ HTML ได้ แต่ในปัจจุบัน XML ได้พัฒนาไปสู่รูปแบบของ Native XML Database ทำให้สามารถลดความยุ่งยากในการออกแบบฐานข้อมูล และลดขั้นตอนการทำงานโดยสามารถจัดเก็บข้อมูลในรูปแบบ XML ได้โดยตรง ทำให้ลดขั้นตอนในการแปลงรูปแบบจากฐานข้อมูลมาในรูปแบบ XML ได้ขั้นตอนหนึ่ง

แต่อย่างไรก็ดีหากต้องการนำเอาความสามารถของ XML มาใช้ร่วมกับการบริหารจัดการเว็บไซต์และโปรแกรมการจัดการเดิมในรูปแบบฐานข้อมูลที่มีอยู่แล้ว หรือต้องการเพิ่มความสามารถในการบริหารจัดการ การใช้รูปแบบของ Native XML Database มาปรับใช้คงไม่ใช่ทางออกที่ดีนัก เนื่องจากต้องแปลงความสัมพันธ์ของฐานข้อมูลทั้งหมดมาอยู่ในรูปแบบของ XML รวมไปถึงการแสดงผล การจัดเก็บ หรือฟังก์ชันในการจัดการต่าง ๆ จะก่อให้เกิดความยุ่งยากในการจัดการเป็นอย่างมาก ทางคณะผู้จัดทำจึงนำเสนอวิธีให้ปรับปรุงโดยใช้การทำงานร่วมกันทั้งในรูปแบบของฐานข้อมูลและ XML โดยในส่วนของการค้นหาข้อมูลและการนำข้อมูลกลับมาแสดงผลนั้นใช้กลไกของ SQL ในการใช้ฐานข้อมูลมาใช้ในการทำงาน ส่วนในการสร้างระบบ HTML UI อัตโนมัตินั้นสร้างจากการใช้ XML โดยใช้ XSLT (eXtensible Stylesheet Language) มาใช้ในการแปลงการจัดเก็บในรูปแบบ XML มาเป็น HTML รวมไปถึงการจัดเก็บข้อมูลที่ใช้วิธีการจัดเก็บในรูปแบบ XML ภายใต้อุปกรณ์ Native XML Database

### HTML form Book

ID 22

Name Applying Use Cases

Cover type  Soft  
 Hard

Description

A practical guide to UML

Change Book

รูปที่ 2.7 ระบบที่ทำการแปลง XML มาใช้งานเป็น UI ในรูปแบบ HTML Form

```

<XMLForm>
  <XMLInstance>
    <VarContent>
      <Book>
        <bookID>22</bookID>
        <bookName>Applying Usr Cases</bookName>
        <cover>Soft</cover>
        <description>A practical guide to UML</description>
      </Book>
    </VarContent>
  </XMLInstance>
<Form>
  <Display key="bookID" label = "ID"/>
  <Entry key = "bookName" label = "Name"/>
  <RadioGroup key="cover" label = "Cover type">
    <Option value = "S" valueTitle = "Soft" />
    <Option value = "H" valueTitle = "Hard" />
  </RadioGroup>
  <LargeEntry key = "description" label = "Description" />
</Form>
</XMLForm>

```

### รูปที่ 2.8 การจัดเก็บเพื่อการแสดงผล UI ในรูป XML

จากงานวิจัยของกลุ่มนักวิจัยชาวโครเอเชียซึ่งนี้ทางผู้จัดทำเห็นว่าน่าจะเป็นแนวทางในการพัฒนาการบริหารจัดการเว็บไซต์ในงานวิจัยซึ่งนี้ได้ โดยนำเอาหลักการจัดการโดยการนำการจัดเก็บข้อมูลและโครงสร้างในรูปแบบ XML Native Database ร่วมกับนำข้อดีมาปรับปรุงโดยยึดหลักการเก็บฐานข้อมูลในรูปแบบเวปเมตาเดต้าของเว็บไซต์แทน เนื่องจากมีข้อดีในหลายประการ อาทิ สามารถจัดเก็บโครงสร้างของเว็บไซต์และเนื้อหาต่าง ๆ บนเว็บไซต์ได้ในครั้งเดียว ในขณะที่โปรแกรมบริหารจัดการเว็บไซต์ในปัจจุบัน มุ่งเน้นแต่การจัดการบริหารในส่วนของเนื้อหาข้อมูลเพียงอย่างเดียว แต่ไม่สนใจในการจัดเก็บโครงสร้างของเว็บไซต์ในขณะนั้นว่ามีการเปลี่ยนแปลงหรือมีลักษณะเป็นอย่างไร อีกทั้งหากนำความสามารถของฐานข้อมูลเชิงเวลาจากการศึกษางานวิจัยเรื่องออกแบบและพัฒนาฐานข้อมูลเชิงเวลาด้วยวิธีในแอม ของมุสตี พรผล มาประยุกต์ใช้ในรูปแบบเวปเมตาเดต้าแล้วนั้น น่าจะทำให้มีประโยชน์และช่วยแก้ปัญหาในการบริหารจัดการเชิงเวลาได้เป็นอย่างดี

ซึ่งต่อไปจะกล่าวถึงข้อมูลและเนื้อหาต่าง ๆ ที่จำเป็นต่อการนำมาใช้ในการออกแบบงานวิจัยเรื่องการบริหารจัดการเว็บไซต์ในรูปแบบเวปเมตาเดต้าซึ่งนี้

## 2.3 โครงสร้างของภาษา HTML

HTML ย่อมาจากคำว่า Hypertext Markup Language เป็นภาษาคอมพิวเตอร์รูปแบบหนึ่ง ซึ่งเป็นภาษาหลักที่ใช้ในการสร้างไฟล์เว็บเพจขึ้นมา ทั้งนี้เป็นภาษาที่ถูกพัฒนามาจากภาษา SGML (Standard Generalized Markup Language) ที่มีลักษณะมีโครงสร้างทางภาษาโดยอาศัยตัวกำกับเรียกว่าอิลิเมนต์ (Element) เพื่อควบคุมการแสดงผลของข้อความ รูปภาพ หรือวัตถุอื่น ๆ ผ่านโปรแกรมบราวเซอร์ (Browser) ที่ใช้สำหรับการแสดงผลเว็บเพจ และในอิลิเมนต์นั้นสามารถที่จะมีแอททริบิวต์ (Attributes) เป็นส่วนขยายคุณลักษณะหรือคุณสมบัติของอิลิเมนต์ เพื่อแสดงผลในรูปแบบที่แตกต่างกันไป ตัวอย่างเช่น กำหนดขนาดตัวอักษร สี ความกว้าง ฯลฯ ในปัจจุบันมีการพัฒนารูปแบบมาตรฐานจากองค์กรต่าง ๆ จนเป็นที่แพร่หลาย และมีความสามารถในการแสดงผลที่เพิ่มขึ้นมาอย่างไม่หยุดยั้ง ภายใต้อุปกรณ์ทางภาษา HTML โดยสามารถศึกษาเพิ่มเติมได้จาก HTML 4.01 Specification ของ W3C <http://www.w3.org/TR/1999/REC-html401-19991224> [2] ทั้งนี้แม้ว่าปัจจุบันภาษา HTML ได้หยุดการพัฒนาลง เนื่องจากองค์กรต่าง ๆ ได้หันไปพัฒนามาตรฐานใหม่แทนเช่น XHTML, PHP หรือ Dot Net ด้วยเหตุผลที่ว่าต้องการให้เว็บมีความสามารถรองรับความต้องการในการตอบสนองกับผู้ใช้ และสามารถนำไปใช้กับอุปกรณ์ต่าง ๆ นอกจากคอมพิวเตอร์ได้มากขึ้น แต่อย่างไรก็ดีไม่ว่าจะมีการพัฒนาด้วยภาษาต่าง ๆ อย่างไร ก็ยังอยู่บนพื้นฐานของการแสดงผลในรูปแบบของ HTML อันจะเห็นได้จากการแสดงผลบนบราวเซอร์ที่ ยังมีการแสดงผลเว็บไซต์ต่าง ๆ ในรูปแบบของภาษา HTML เช่นเดิม

### 2.3.1 ลักษณะทางภาษา HTML

ลักษณะทางภาษา HTML นั้นสามารถแบ่งออกได้เป็น 2 ส่วนด้วยกัน คือ ส่วนที่เป็นข้อมูล โดยทั่วไป (Content) ซึ่งรวมไปถึงรูปภาพ วิดีโอ และสื่ออิเล็กทรอนิกส์ต่าง ๆ และส่วนที่ 2 คือส่วนที่เป็นคำสั่งที่ใช้ในการกำหนดรูปแบบของข้อความที่แสดง อาทิ การทำให้ตัวอักษรเป็นตัวหนา ตัวเอียง การวางตำแหน่ง หรืออื่น ๆ ซึ่งเรียกส่วนนี้ว่า แท็ก (Tag) โดยรูปแบบของแท็กเป็นลักษณะเฉพาะของภาษา HTML โดยคำสั่งเพื่อการแสดงผลในรูปแบบภาษา HTML นั้น จะอยู่ในเครื่องหมาย < และ > แท็กมี 2 ลักษณะคือ มีทั้งแท็กเปิด <.> และ แท็กปิด </.> โดยแท็กปิดจะมีเครื่องหมาย / นำหน้าคำสั่งในแท็กนั้น ๆ เช่น <B>...</B>, <P>...</P> เป็นต้น ทั้งนี้คำสั่งต่าง ๆ ที่นำมาเขียนในแท็กนั้นเรียกว่าอิลิเมนต์ ซึ่งใน HTML 4.01 (มาตรฐานปัจจุบัน) นี้มีทั้งหมด 91 ชนิดด้วยกัน

ตัวอย่างเช่น <TITLE>Google</TITLE>

ความหมาย แสดงผลที่ไตเติ้ลบาร์ของบราวเซอร์ให้แสดงคำว่า Google เป็นต้น

นอกจากนี้ยังมีข้อยกเว้นในบางอิลิเมนต์ ไม่จำเป็นจะต้องมีแท็กปิด และยังมีรูปแบบเฉพาะอีกหลายรูปแบบ โดยรายละเอียดข้อจำกัดต่าง ๆ จะกล่าวในภาคผนวก ข. ต่อไป นอกจากนี้

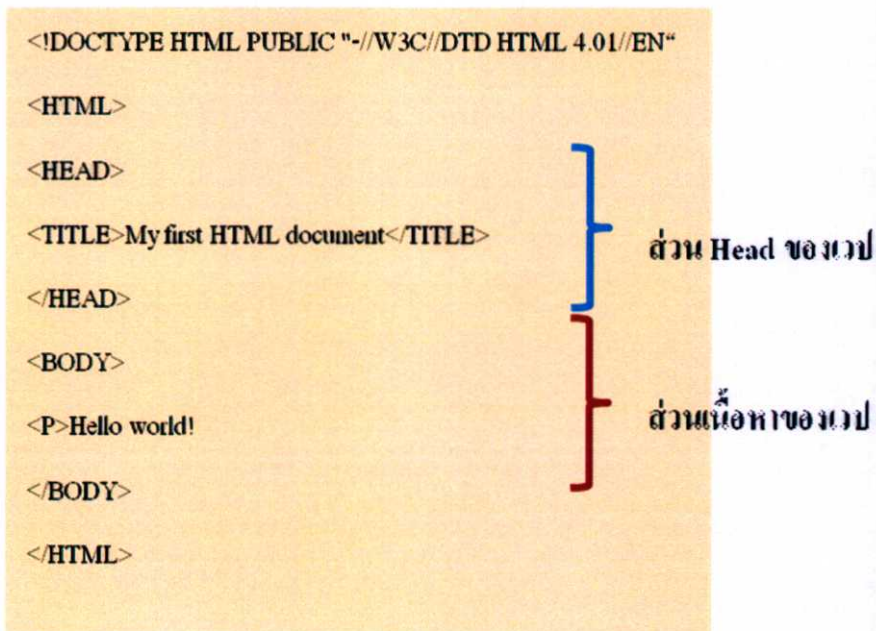
แล้วสิ่งสำคัญอีกส่วนหนึ่งของการใช้แท็กนั้นก็คือส่วนขยายที่สามารถขยายความลักษณะหรือคุณสมบัติในการแสดงผลของอีลิเมนต์ให้มีรายละเอียดมากยิ่งขึ้น เรียกส่วนขยายนี้ว่าแอททริบิวต์ ซึ่งจะต้องใส่ภายในเครื่องหมาย <> ต่อจากส่วนของอีลิเมนต์เฉพาะในส่วนแท็กเปิดของอีลิเมนต์เท่านั้นและมีส่วนของการบอกค่าของคุณลักษณะภายใต้เงื่อนไขของแต่ละแอททริบิวต์ตามข้อกำหนดในแต่ละอีลิเมนต์ แท็กคำสั่งในแต่ละอีลิเมนต์ที่มีแอททริบิวต์ที่อาจจะแตกต่างกันไป และอาจมีจำนวนไม่เท่ากันขึ้นอยู่กับชนิดของอีลิเมนต์ การระบุแอททริบิวต์ที่มากกว่า 1 ชนิดนั้นสามารถทำได้โดยใช้ช่องว่างเป็นตัวคั่น

ตัวอย่างเช่น <FONT color="red" size="+3">Thailand</FONT>

ความหมาย ให้คำว่า Thailand ที่อยู่ภายใต้อีลิเมนต์ FONT นี้แสดงผลเป็นสีแดงและมีขนาด +3

### 2.3.2 โครงสร้างพื้นฐานของ HTML

โครงสร้างพื้นฐานทางภาษาของ HTML นั้น สามารถแบ่งออกเป็น 2 ส่วนหลักดังนี้



รูปที่ 2.9 โครงสร้างพื้นฐานทางภาษา HTML

#### 2.3.2.1 ส่วนของ HEAD

สำหรับข้อมูลในส่วนหัวของ HTML เช่น ข้อความบนไตเติลบาร์ เป็นต้น ซึ่งอาจจะใช้ในการกำหนดข้อมูลต่าง ๆ ของเว็บเพจนั้น ๆ เช่น ข้อมูลเพื่อการค้นหา กำหนดรูปแบบข้อความและรูปแบบการจัดวางของเว็บเพจ รวมไปถึงการแทรก หรือใส่ลูกเล่นของเว็บไซต์จากภาษาอื่นที่นำมาใช้ในการเขียนร่วมกับ HTML อาทิ JavaScript หรือ VBScript เป็นต้น

### 2.3.2.2 ส่วนของ BODY

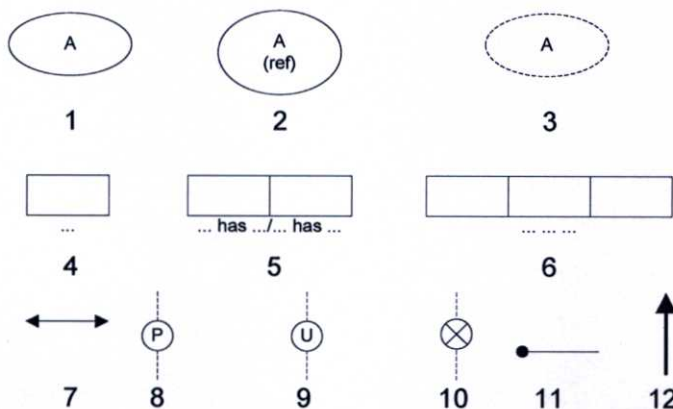
สำหรับการแสดงผลเนื้อหาและข้อมูลเกี่ยวกับภาพหรืออื่น ๆ แสดงออกยังหน้าเอกสารหรือเวปเพจ โดยทั้ง 2 ส่วนประกอบข้างต้น จะถูกกำกับภายใต้แท็กอิลีเมนต์และส่วนการควบคุมต่าง ๆ

## 2.4 แบบจำลองข้อมูลแบบไนแอม (NIAM) และฐานข้อมูลเชิงเวลา TNIAM

แบบจำลองข้อมูลระดับแนวคิดไนแอม (NIAM Conceptual Model หรือ NIAM Schema) นั้น คำว่า NIAM ย่อมาจาก Nijssen's Information Analysis Methodology ซึ่งเป็นรูปแบบหนึ่งของแบบจำลองข้อมูลในระดับแนวคิด ที่ได้มีการคิดค้นมาตั้งแต่ปี ค.ศ. 1977 โดยศาสตราจารย์จีเอ็ม ไนเซน [3] ได้เสนอแบบจำลองข้อมูลไนแอม ซึ่งเป็นแบบจำลองที่มีพื้นฐานมาจากโครงสร้างภาษาธรรมชาติ ที่มีรูปประโยคประกอบด้วย ประธาน กริยา และกรรม เป็นแบบจำลองระดับแนวคิด ที่มีความหมายและเครื่องหมาย แสดงความสัมพันธ์ของข้อมูล และข้อจำกัดของข้อมูลได้อย่างชัดเจน นอกจากนี้แล้วยังสามารถแปลง แบบจำลองดังกล่าว เป็นโครงสร้างของฐานข้อมูลเชิงสัมพันธ์ (Relational Database Schema) ซึ่งผลลัพธ์ที่ได้จากการแปลงจะอยู่ในรูปของ Fifth Normal Form (5NF) ดังนั้น จึงมีการนำแบบจำลองข้อมูลระดับแนวคิดไนแอม มาใช้ช่วยในการออกแบบฐานข้อมูลเชิงสัมพันธ์กันอย่างแพร่หลาย เพราะมีการใช้สัญลักษณ์ที่แสดงความสัมพันธ์ของข้อมูลอย่างชัดเจน และง่ายต่อการเข้าใจ และในปัจจุบันได้มีการนำเอาแบบจำลองข้อมูลแบบไนแอมไปประยุกต์ใช้กับงานด้านอื่น ๆ เช่น งานวิจัยเกี่ยวกับระบบผู้เชี่ยวชาญ งานวิจัยด้านคลังข้อมูล เป็นต้น

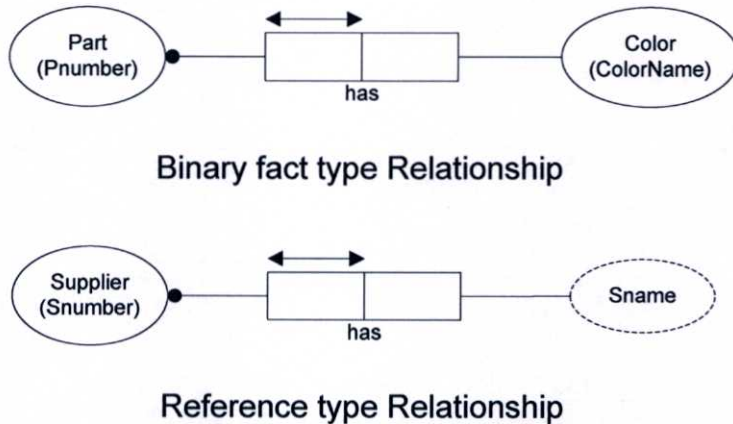
### 2.4.1 ส่วนประกอบของไนแอมสกีมา (NIAM Schema)

แบบจำลองข้อมูลไนแอมประกอบด้วยสัญลักษณ์พื้นฐาน ที่ใช้ในการสื่อความหมายของประโยคภาษาธรรมชาติ ตามที่ผู้สร้างต้องการนำเสนอ สัญลักษณ์ต่าง ๆ แสดงดังรูปที่ 2-10



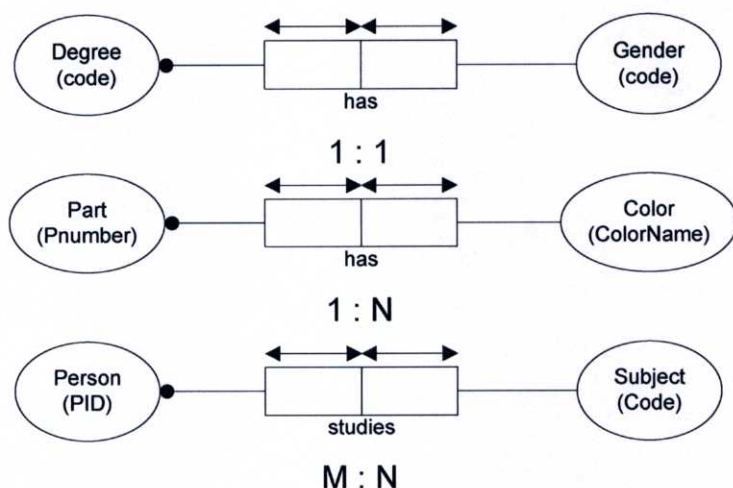
รูปที่ 2.10 สัญลักษณ์พื้นฐานของไนแอมสกีมา

สัญลักษณ์ที่แสดงในรูปที่ 2.10 เป็นสัญลักษณ์พื้นฐานที่นำมาสร้างแบบจำลองข้อมูลในแอม แต่ละสัญลักษณ์มีความหมายต่างกันคือ สัญลักษณ์หมายเลข 1- 3 เป็นสัญลักษณ์ที่แทนออบเจกต์ไพบี (Object Type) ซึ่งออบเจกต์ไพบีแบ่งออกเป็น 2 ประเภทคือ Entity type (หมายเลข 1,2) และ Value or Label Type (หมายเลข 3) หมายเลข 4-6 แสดงถึงสัญลักษณ์ของบทบาทหรือกริยา (Role) ของออบเจกต์ หมายเลข 4 แสดงถึงบทบาทเดียว (Unary role) หมายเลข 5 แสดงสัญลักษณ์ของสองบทบาท (Binary role) และหมายเลข 3 แสดงสัญลักษณ์ของสามบทบาท (Tertiary role) หมายเลข 7 ถึง 12 แสดงถึงคอนสเตรนหรือกฎบังคับความถูกต้องแบบต่าง ๆ คือ หมายเลข 7 แสดง Internal Uniqueness Constraint หมายเลข 8 แสดง External Uniqueness Constraint Primary หมายเลข 9 แสดงสัญลักษณ์ของ External Uniqueness Constraint หมายเลข 10 แสดงสัญลักษณ์ของ Exclusion Constraint หมายเลข 11 แสดงสัญลักษณ์ของ Mandatory role และหมายเลข 12 แสดงสัญลักษณ์ของ Subtype สัญลักษณ์ของโนแอมสกีมาที่กล่าวมานี้เป็นเพียงสัญลักษณ์พื้นฐานที่นำมาใช้ในการวิจัยนี้ แต่ยังมีอีกหลายสัญลักษณ์ที่ไม่ได้กล่าวถึง การสร้างในแอมสกีมาเกิดจากการนำเอาออบเจกต์มาเชื่อมต่อกันหรือมาสัมพันธ์กัน (Relationship) ประเภทของความสัมพันธ์กันระหว่าง 2 ออบเจกต์มี 2 ประเภทคือ Fact type relationship และ Reference type relationship ความสัมพันธ์ทั้งสองแบบแสดงดังรูปที่ 2.11



รูปที่ 2.11 ความสัมพันธ์ของออบเจกต์

โดยที่ Fact type relationship แสดงถึงความสัมพันธ์ระหว่างเอนติตี้ไพบีกับเอนติตี้ไพบี และ Reference type relationship แสดงถึงความสัมพันธ์ระหว่างเอนติตี้ไพบีกับเลเบลไพบี นอกจากนี้ประเภทของความสัมพันธ์ยังสามารถแยกได้อีก 3 ประเภทคือ ความสัมพันธ์แบบ 1 : 1, 1 : N และ M : N การพิจารณาจากจำนวนบทบาทที่ Internal Uniqueness Constraint ครอบคลุมประเภทของความสัมพันธ์ทั้งสามรูปแบบ แสดงดังรูปที่ 2.12



รูปที่ 2.12 ความสัมพันธ์แบบ 1:1, 1:N และ M:N

#### 2.4.2 ฐานข้อมูลเชิงเวลา TNIAM และการออกแบบ [3], [4]

จากการศึกษาฐานข้อมูลเชิงเวลาเรื่องการออกแบบและพัฒนาฐานข้อมูลเชิงเวลาด้วยวิธีในแอม ของผุสดี พรผล สาขาเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ได้นำเสนอแนวคิดในการออกแบบฐานข้อมูลเชิงเวลาด้วยวิธี TNIAM โดยการพัฒนาจากออกแบบฐานข้อมูลด้วยวิธีในแอม โดยการเพิ่มสัญลักษณ์บางอย่างและส่วนของเวลาเข้าไปในโครงสร้างของในแอม เพื่อให้รองรับการออกแบบฐานข้อมูลเชิงเวลาได้ นอกจากนี้ยังได้นำเสนอขั้นตอนต่าง ๆ ในการแปลงโมเดล TNIAM ให้อยู่ในรูป 5NF ได้โดยตรง เพื่อนำมาประยุกต์ใช้กับระบบงานจริงและพัฒนาเป็นโปรแกรมประยุกต์

ในงานการวิจัยเรื่องการออกแบบและพัฒนาฐานข้อมูลเชิงเวลาด้วยวิธีในแอม นี้ ได้เน้นรูปแบบของเวลา 2 รูปแบบคือแบบ Lifespan Time และ Valid Time โดยได้ให้ความหมายของช่วงเวลาต่าง ๆ ดังนี้

##### 2.4.2.1 Lifespan Time

คือระยะเวลาที่เอนติตี้ไทม์นั้นเกิดขึ้นมาจนกระทั่งเอนติตี้ไทม์นั้นจบลง ถ้าเอนติตี้ไทม์ใดมีการระบุค่าเวลาของเอนติตี้ไทม์หมายความว่ารูปแบบเวลาของเอนติตี้ไทม์นั้นจะถูกเก็บลงฐานข้อมูล

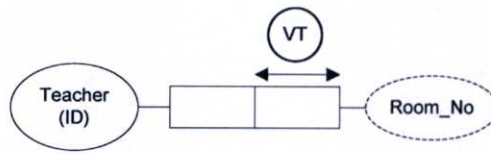


รูปที่ 2.13 ตัวอย่างสัญลักษณ์ที่ใช้การออกแบบฐานข้อมูล แบบ LifeSpan Time

### 2.4.2.2 Valid Time

คือช่วงเวลาที่ข้อมูลเป็นจริงคือเวลาที่ข้อมูลเกิดขึ้นจริง ๆ ซึ่งทุก ๆ ความจริงจะมีค่าเวลาที่ข้อมูลเป็นจริงอยู่ โดยพิจารณาการจับเก็บจาก 3 วิธีคือ

- ทำการจับเก็บช่วงเวลาที่ข้อมูลนั้นเป็นจริง เมื่อสามารถทำการจัดการจับเก็บทุก ๆ ค่าที่เป็นจริงถูกต้อง และคงอยู่ตลอดไปของความจริงของข้อมูลนั้น
- จะไม่ทำการจับเก็บช่วงเวลาที่ข้อมูลนั้นเป็นจริง เมื่อค่าความจริงของข้อมูลไม่มีการเปลี่ยนแปลง
- ไม่ทำการจับเก็บเวลาที่ข้อมูลเป็นจริงของความจริงของข้อมูล เมื่อรู้ค่าที่ถูกต้องแต่ไม่รู้เวลาที่แน่นอน



รูปที่ 2.14 ตัวอย่างสัญลักษณ์ที่ใช้การออกแบบฐานข้อมูล แบบ Valid Time

เพื่อความเหมาะสมของประเภทการใช้งานในฐานข้อมูล งานวิจัยชิ้นนี้จึงใช้วิธีการจัดรูปแบบเวลาแบบ Valid Time โดยจะบันทึกเฉพาะข้อมูลที่เกิดการเปลี่ยนแปลงลงในฐานข้อมูล โดยพิจารณาว่าเอนติตี้ใหม่ ก็เป็นข้อมูลหนึ่งของระบบเช่นเดียวกัน

ในบทที่ 3 ต่อจากนี้จะกล่าวถึงการประยุกต์ใช้เมตาเดต้าของเวปไซต์ในการออกแบบฐานข้อมูลเพื่อนำไปใช้ในการจัดการเวปไซต์ในรูปแบบต่าง ๆ

## บทที่ 3

### การออกแบบฐานข้อมูลสำหรับเว็บเมตาเดต้าด้วยโมเดลในแอม

ในบทความวิจัยนี้ได้ทำการทดลองโดยการใช้ทฤษฎีและหลักการเบื้องต้นจากการศึกษาโครงสร้างของเวปเพจ และความสัมพันธ์ของเว็บเมตาเดต้าตามรูปแบบ HTML 4.01 Specification ที่ W3C ได้กำหนดไว้ ตัวอย่างเช่น การสร้างเวปไซต์มีโครงสร้างมาตรฐานมีลักษณะเป็นอย่างไร ส่วนประกอบของอีลีเมนต์แต่ละตัว เช่น อีลีเมนต์ HEAD มีแอททริบิวต์ใดบ้างที่สามารถมาใช้งานร่วมกันได้ และมีขอบเขตของค่าแอททริบิวต์ที่ใช้กำหนดการแสดงผลของแต่ละอีลีเมนต์ พร้อมทั้งความสัมพันธ์ในรูปแบบการจัดเก็บเพื่อใช้ในการแสดงผลได้อย่างไร รวมไปถึงข้อจำกัดและข้อยกเว้นในค่าต่าง ๆ ของ อีลีเมนต์และแอททริบิวต์ เช่น ค่าสีของอีลีเมนต์ FONT อยู่ในช่วง #000000 – #FFFFFF เป็นต้น

จากนั้นได้นำความสัมพันธ์ที่ได้จากการศึกษามาวิเคราะห์และออกแบบฐานข้อมูลด้วยวิธีในแอม มาทำการแมปปิ้งให้อยู่ในรูปของฐานข้อมูลในรูปแบบเว็บเมตาเดต้าเพื่อนำมาใช้บันทึกจัดเก็บลงในฐานข้อมูลต่อไป ตามขั้นตอนการออกแบบฐานข้อมูลเป็นเว็บเมตาเดต้าด้วยวิธีในแอมตามขั้นตอนต่อไปนี้

#### 3.1 สร้างเอนติตี้ที่เกี่ยวข้อง

ทำการออกแบบความสัมพันธ์ของแต่ละอีลีเมนต์ของเว็บและการแสดงผล โดยกำหนดเอนติตี้พร้อมทั้งระบุชนิดข้อมูลและอื่น ๆ ที่เกี่ยวข้องทั้งหมดของโครงสร้างฐานข้อมูล ซึ่งนำมาจัดมาในรูปแบบ Entity Data Dictionary ได้ตามตารางที่ 3.1

ตารางที่ 3.1 ค่าคุณสมบัติเอนติตี้ที่เกี่ยวข้องทั้งหมดในการออกแบบ

Entity Name	IDentifier	Data type	Description	Example Data
URL_ID	URL_ID	Number	หมายเลขของเวปเพจ	11070011
Domain_Name	URL_ID	Text	ชื่อของโดเมนของเวปไซต์	<a href="http://www.google.co.th">www.google.co.th</a>
Element_ID	Element_ID	Number	หมายเลขของอีลีเมนต์แต่ละแท็ก	12050010
Element_type_Name	Element_ID	Text	ชื่อของอีลีเมนต์แต่ละประเภท	BODY
Order_Number	Element_ID	Number	ลำดับของอีลีเมนต์	165
Content_text	Element_ID	Text	ข้อมูลของแต่ละอีลีเมนต์	ระบบค้นหาข้อมูล
Element_type_Name	Element_type_Name	Number	หมายเลขของอีลีเมนต์แต่ละประเภท	12

ตารางที่ 3.1 (ต่อ) ค่าคุณสมบัติเอนิตี้ที่เกี่ยวข้องทั้งหมดในการออกแบบ

Entity Name	IDentifier	Data type	Description	Example Data
Attrib_type_Name	Attrib_type_Name	Text	ชื่อของแอททริบิวต์แต่ละประเภท	Width
Unit_Type	Unit_Type	Text	ประเภทของข้อมูล (F = Fix Value, R = Relate Value, N = None )	F
Start_Value	Element_type_Name, Attrib_type_Name, Unit_Type	Text	ค่าเริ่มต้นของแต่ละแอททริบิวต์ในแต่ละอีลิเมนต์	#000000
Limit_Value	Element_type_Name, Attrib_type_Name, Unit_Type	Text	ค่าจำกัดของแต่ละแอททริบิวต์ในแต่ละอีลิเมนต์	#FFFFFF
Default_Value	Element_type_Name, Attrib_type_Name, Unit_Type	Text	ค่ามาตรฐานของแต่ละแอททริบิวต์ในแต่ละอีลิเมนต์	#FEFEFE
Element_ID	Element_ID	Number	หมายเลขของอีลิเมนต์แต่ละ Tag	12050010
Attrib_type_Name	Attrib_type_Name	Text	ชื่อของแอททริบิวต์แต่ละประเภท	Width
Unit_Type	Unit_Type	Text	ประเภทของข้อมูล (F = Fix Value, R = Relate Value, N = None )	F
Value	Element_ID, Attrib_type_Name, Unit_Type	Text	ค่าของแต่ละแอททริบิวต์ของอีลิเมนต์ในแต่ละแท็ก	30%
Element_ID	Element_ID	Number	หมายเลขของอีลิเมนต์แต่ละประเภท	12
Sub_Element_ID	Sub_Element_ID	Number	หมายเลขของอีลิเมนต์ที่สัมพันธ์กับประเภทของอีลิเมนต์	13
Element_type_Name	Element_type_Name	Text	ชื่อของอีลิเมนต์แต่ละประเภท	BODY
Sub_Element_type_Name	Sub_Element_type_Name	Text	ชื่อของอีลิเมนต์แต่ละประเภท	TABLE
Qty	Element_type_Name, Sub_Element_type_Name	Text	บอกรูปแบบของความสัมพันธ์ (1 = One , N = Many)	N

ตารางที่ 3.1 (ต่อ) ค่าคุณสมบัติเอนติตี้ที่เกี่ยวข้องทั้งหมดในการออกแบบ

Entity Name	Identifier	Data type	Description	Example Data
Element_type_Name	Element_type_Name	Text	ชื่อของอีลิเมนต์แต่ละประเภท	TABLE
Start_Tag	Element_type_Name	Text	ต้องมีแท็กเริ่มหรือไม่ (Y = Must ,O = Option , F = Forbidden)	Y
End_Tag	Element_type_Name	Text	ต้องมีแท็กจบหรือไม่ (Y = Must ,O = Option , F = Forbidden)	F
Empty	Element_type_Name	Text	ในการใช้เป็นอีลิเมนต์ย่อยไม่มีแท็กนี้ได้หรือไม่ (E = ไม่มีได้ ,N = ต้องมี)	E

### 3.2 สร้างความสัมพันธ์ด้วย Elementary Fact Type

เขียน Elementary Fact Type เพื่อแสดงความสัมพันธ์ในแต่เอนติตี้ซึ่งจะบอกได้ว่า ความสัมพันธ์ระหว่างเอนติตี้เป็นความสัมพันธ์รูปแบบใดจากการทดลองเขียนประโยค

ตารางที่ 3.2 ค่า Elementary Fact Type แสดงความสัมพันธ์กันในแต่ละอีลิเมนต์

Relationship	Cardinality	Description	Example
URL (ID) has Domain name	N : 1	Each URL(ID) has most 1 URL Name ,but each Domain name may be duplicated	URL หมายเลข 20 คือ www.google.co.th
URL has Element	1 : N	Each URL(ID) has many Elements ,but each Element has most 1 URL	URL หมายเลข 20 ประกอบด้วยอีลิเมนต์หมายเลข 15
Element (ID) is Element Name	N : 1	Each Element(ID) has most 1 Element Name ,but each Element Name may be duplicated	อีลิเมนต์ ID ที่ 20 คืออีลิเมนต์ประเภท BODY
Element (ID) has order number	N : 1	Each Element(ID) has most 1 Order_Number ,but each Order_Number may be duplicated	อีลิเมนต์ ID ที่ 20 อยู่ในลำดับที่ 19
Element (ID) has content	N : 1	Each Element(ID) has most 1 content ,but each Content may be duplicated	อีลิเมนต์ ID ที่ 20 มีข้อมูลว่า "Google"
Element has Attribute and type of value	M : N	Each Element has many Attribute and many type of value ,and each Element may be duplicated	อีลิเมนต์ประเภท TABLE มีแอททริบิวต์ width ที่ประเภทข้อมูลเป็นแบบ Fix

ตารางที่ 3.2 (ต่อ) ค่า Elementary Fact Type แสดงความสัมพันธ์กันในแต่ละอีลีเมนต์

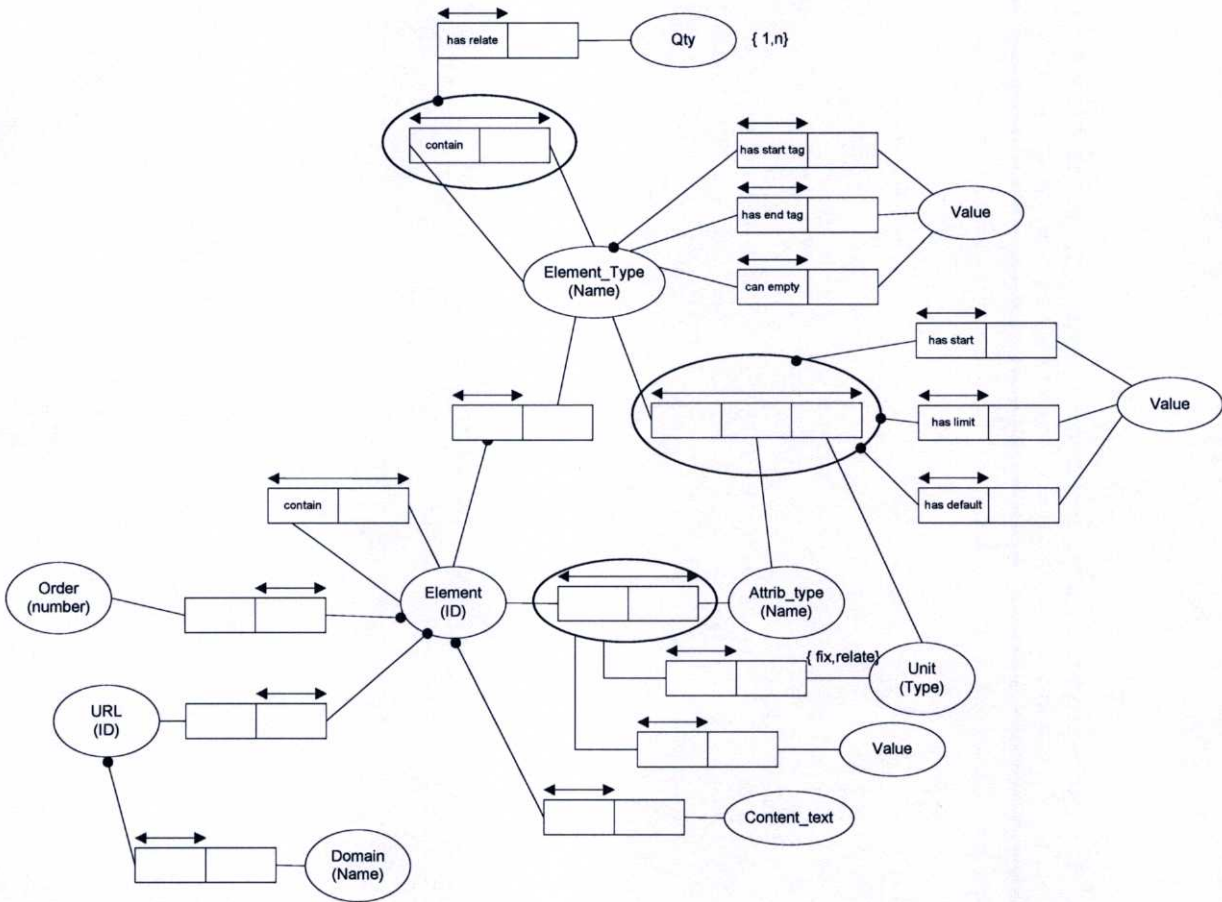
Relationship	Cardinality	Description	Example
Element and Attribute and type of value has Start Value	N : 1	Each Element , Attribute and Type of value has 1 Start Value ,but each Start Value may be duplicated	อีลีเมนต์ประเภท TABLE มีแอททริบิวท์ width ที่ประเภทข้อมูลเป็นแบบ Fix มีค่าไม่ต่ำกว่า 30
Element and Attribute and type of vaule has limit Value	N : 1	Each Element Type , Attribute and Type of vaule has 1 Limit Value ,and each Limit Value may be duplicated	อีลีเมนต์ประเภท TABLE มีแอททริบิวท์ width ที่ประเภทข้อมูลเป็นแบบ Fix มีค่าไม่เกินกว่า 100
Element and Attribute and type of vaule has Defalut Value	N : 1	Each Element Type , Attribute and Type of vaule has 1 default Value ,but each Default Value may be duplicated	อีลีเมนต์ประเภท BODY มีแอททริบิวท์ bgcolor ที่ประเภทข้อมูลเป็นแบบ Fix หากไม่ระบุค่าจะมีค่าเป็น "#FFFFFF"
Element(ID) has and attribute and type of vaule	M : N	Each Element(ID) has many Attribute and Type of vaule ,and each Element(ID) may be duplicated	อีลีเมนต์ ID ที่ 20 มีแอททริบิวท์ width ที่ประเภทข้อมูลเป็นแบบ Fix
Element(ID) and attribute and type of value has value	N : 1	Each Element(ID), Attribute and Type of vaule has 1 Value ,but each Value may be duplicated	อีลีเมนต์ ID ที่ 20 มีแอททริบิวท์ width ที่ประเภทข้อมูลเป็นแบบ Fix มีค่าเท่ากับ 30
Element (ID) contain Element(ID)	M : N	Each Element(ID)has contain many Sub Element(ID),and each Sub Element(ID) may be duplicated	อีลีเมนต์ ID ที่ 20 มี อีลีเมนต์ ID ที่ 21 เป็นส่วนประกอบ
Element relate Sub Element	M : N	Each Element has relate many Sub Element ,and each Sub Element type may be duplicated	อีลีเมนต์ประเภท TR เกี่ยวข้องกับอีลีเมนต์ประเภท TD
Element and Sub Element has relation (Qty)	N : 1	Each Relation of Element and Sub Element has 1 Relation ,but each Relation may be duplicated	อีลีเมนต์ประเภท TR เกี่ยวข้องกับอีลีเมนต์ประเภท TD ในแบบ N (Many)

ตารางที่ 3.2 (ต่อ) ค่า Elementary Fact Type แสดงความสัมพันธ์กันในแต่ละอีลีเมนต์

Relationship	Cardinality	Description	Example
Element has Start tag condition	N : 1	Each Element has most 1 condition of Start tag ,and each condition of Start tag may be duplicated	อีลีเมนต์ประเภท TR การใช้ต้องมีแท็กเริ่ม (Y)
Element has End tag condition	N : 1	Each Element has most 1 condition of End tag ,and each condition of End tag may be duplicated	อีลีเมนต์ประเภท BODY การจะมีแท็กจบหรือไม่ก็ได้ (O)
Element has empty tag condition	N : 1	Each Element has most 1 condition of Empty tag ,and each condition of End tag may be duplicated	อีลีเมนต์ประเภท FRAME ไม่จำเป็นต้องมีแท็ก (E)

### 3.3 ออกแบบความสัมพันธ์ของโครงสร้างของเว็บจากภาษา HTML ด้วยวิธีในแอม

จากการศึกษาความสัมพันธ์ของ HTML ทำให้สามารถออกแบบไดอะแกรมความสัมพันธ์ด้วย ในแอม โมเดล ได้ดังนี้



รูปที่ 3.1 ความสัมพันธ์ที่ออกแบบด้วยวิธีในแอม

### 3.4 ทำการแมปปีงจากแบบจำลองโครงสร้างฐานข้อมูล

จากความสัมพันธ์ที่ได้ทำการออกแบบไว้ สามารถเขียนเป็นตารางของความสัมพันธ์ต่าง ๆ ได้ดังนี้

ตาราง URL

URL_ID	Domain_Name
--------	-------------

ตาราง Content

Element_ID	Element_type_Name	URL_ID	Order_Number	Content_text
------------	-------------------	--------	--------------	--------------

ตาราง Element\_Type\_Attribute\_Properties

Element_type_Name	Attrib_type_Name	Unit_Type	Start_Value	Limit_Value	Default_Value
-------------------	------------------	-----------	-------------	-------------	---------------

ตาราง Element\_Attribute\_Value

Element_ID	Attrib_type_Name	Unit_Type	Value
------------	------------------	-----------	-------

ตาราง Element\_Relation

Element_ID	Sub_Element_ID
------------	----------------

ตาราง Element\_Type\_Relation

Element_Type_Name	Sub_Element_Type_Name	Qty
-------------------	-----------------------	-----

ตาราง Element\_Properties

Element_type_Name	Start_tag	End_tag	Empty
-------------------	-----------	---------	-------

รูปที่ 3.2 ความสัมพันธ์ของฐานข้อมูลที่ออกแบบ

### 3.5 การแมปพื้นฐานข้อมูลมาเป็นตาราง

สามารถแมปพื้นฐานข้อมูลจากการออกแบบในรูปแบบของตารางความสัมพันธ์ได้เป็น 7 ตารางดังต่อไปนี้

#### ตารางที่ 3.3 URL

Name	Type	Key	Meaning	Example
URL_ID	Number	Primary Key	หมายเลขของเวปเพจ	11070011
Domain_Name	Text	-	ชื่อโดเมนของเวปไซด์	www.google.co.th

#### ตารางที่ 3.4 Content

Name	Type	Key	Meaning	Example
Element_ID	Number	Primary Key	หมายเลขของอีลิเมนต์แต่ละแท็ก	12050010
Element_type_Name	Text	-	ชื่อของอีลิเมนต์แต่ละประเภท	BODY
URL_ID	Number	-	หมายเลขของเวปเพจ	11070011
Order_Number	Number	-	ลำดับของอีลิเมนต์	165
Content_text	Text	-	ข้อมูลของแต่ละอีลิเมนต์	การทดลอง

#### ตารางที่ 3.5 Element\_type - Attribute\_Properties

Name	Type	Key	Meaning	Example
Element_type_Name	Text	Composite Key	ชื่อของอีลิเมนต์แต่ละประเภท	BODY
Attrib_type_Name	Text	Composite Key	ชื่อของแอททริบิวต์แต่ละประเภท	Width
Unit_Type	Text	Composite Key	ประเภทของข้อมูล (F = Fix Value, R = Relate Value, N = None )	F
Start_Value	Text	-	ค่าเริ่มต้นของแต่ละแอททริบิวต์ในแต่ละอีลิเมนต์	#000000
Limit_Value	Text	-	ค่าจำกัดของแต่ละแอททริบิวต์ในแต่ละอีลิเมนต์	#FFFFFF
Default_Value	Text	-	ค่ามาตรฐานของแต่ละแอททริบิวต์ในแต่ละอีลิเมนต์	#FEFEFE

ตารางที่ 3.6 Element - Attribute\_Value

Name	Type	Key	Meaning	Example
Element_ID	Number	Composite Key	หมายเลขของอีลิเมนต์แต่ละแท็ก	12050010
Attrib_type_Name	Text	Composite Key	ชื่อของแอททริบิวต์แต่ละประเภท	Width
Unit_Type	Text	-	ประเภทของข้อมูล (F = Fix Value, R = Relate Value, N = None )	F
Value	Text	-	ค่าของแต่ละแอททริบิวต์ของอีลิเมนต์ในแต่ละแท็ก	30%

ตารางที่ 3.7 Element Relation

Name	Type	Key	Meaning	Example
Element_ID	Number	Composite Key	ID ของอีลิเมนต์แต่ละตัว	12
Sub_Element_ID	Number	Composite Key	ID ของอีลิเมนต์แต่ละตัว	13

ตารางที่ 3.8 Element Type Relation

Name	Type	Key	Meaning	Example
Element_type_Name	Text	Composite Key	ชื่อของอีลิเมนต์แต่ละประเภท	BODY
Sub_Element_type_Name	Text	Composite Key	ชื่อของอีลิเมนต์แต่ละประเภท	TABLE
Qty	Text	-	บอกรูปแบบของความสัมพันธ์ (1 = One , N = Many)	N

ตารางที่ 3.9 Element Properties

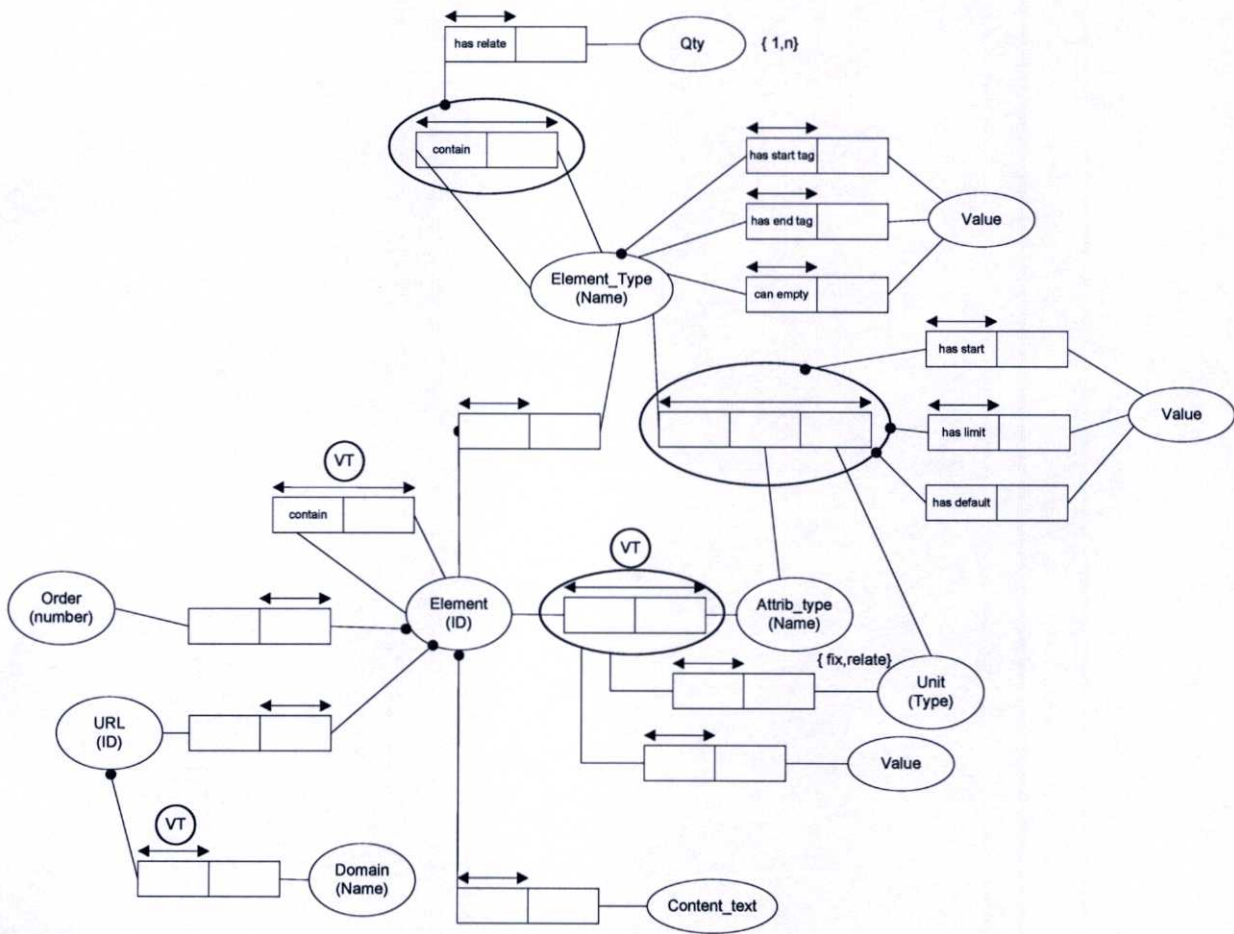
Name	Type	Key	Meaning	Example
Element_type_Name	Text	Primary Key	ชื่อของอีลิเมนต์แต่ละประเภท	TABLE
Start_Tag	Text	-	ต้องมีแท็กเริ่มหรือไม่ (Y = Must ,O = Option , F = Forbidden)	Y
End_Tag	Text	-	ต้องมีแท็กจบหรือไม่ (Y = Must ,O = Option , F = Forbidden)	F
Empty	Text	-	ในการใช้เป็นอีลิเมนต์ย่อยไม่มีแท็ก นี้ได้หรือไม่ (E = ไม่มีได้ ,N = ต้องมี )	E

เมื่อทดสอบฐานข้อมูลว่าสามารถรองรับโครงสร้างทั่วไปของเว็บไซต์ต่าง ๆ ได้แล้ว จึงมาทำการออกแบบและนำฐานข้อมูลที่ได้ทำการทดสอบมาแล้วนั้นมาทำการปรับปรุง ในรูปแบบเชิงเวลาแบบที่อ้างอิงเวลาที่เป็นจริง (Valid Time) เพื่อทดสอบนำความสามารถที่เพิ่มขึ้นในเรื่องของความสามารถเชิงเวลามาประยุกต์ใช้ในงานต่าง ๆ เช่นการย้อนกลับมาดูเว็บไซต์ในช่วงเวลาที่แตกต่างกัน ยกตัวอย่างในกรณีเมื่อต้องการย้อนกลับไปดูหน้าเว็บเมื่อ 1 เดือนที่แล้วว่ามีเป็น

อย่างไรเปลี่ยนแปลงไปจากเดิมอย่างไร หรือการวางแผนล่วงหน้าสำหรับการปรับเปลี่ยนเนื้อหาเว็บไซต์โดยอัตโนมัติ ตัวอย่างเช่น การปรับเปลี่ยนโปรโมชั่นสินค้าตามเทศกาล ซึ่งจะเห็นได้จากหัวข้อถัดไป

### 3.6 ออกแบบโมเดลความสัมพันธ์ของโครงสร้างเว็บเมตาเดต้าเชิงเวลาด้วย TNIAM

ทำการออกแบบการจัดเก็บเว็บเพจในรูปของฐานข้อมูลเชิงเวลา ด้วยวิธี TNIAM โดยการปรับปรุงจากการออกแบบฐานข้อมูลให้อยู่ในรูปแบบเชิงเวลาได้ดังนี้



รูปที่ 3.3 ความสัมพันธ์ของเว็บเมตาเดต้าในรูปแบบฐานข้อมูลเชิงเวลา TNIAM

### 3.7 การแมปโปงแบบจำลองโครงสร้างฐานข้อมูลเชิงเวลาที่ปรับปรุงด้วย TNIAM

จากโมเดลโนแอมสามารถเขียนเป็นตารางของความสัมพันธ์ต่าง ๆ ได้ดังนี้

ตาราง URL

<u>URL_ID</u>	Vts	Domain_Name	Vte
---------------	-----	-------------	-----

ตาราง Element\_Type\_Relation

Element_Type_Name	Sub_Element_Type_Name	Qty
-------------------	-----------------------	-----

ตาราง Content

<u>Element_ID</u>	Vts	Element_type_Name	URL_ID	Order_Number	Content_text	VTe
-------------------	-----	-------------------	--------	--------------	--------------	-----

ตาราง Element\_Attribute\_Properties

Element_type_Name	Attrib_type_Name	Unit_Type	Start_Value	Limit_Value	Default_Value
-------------------	------------------	-----------	-------------	-------------	---------------

ตาราง Element\_Attribute\_Value

<u>Element_ID</u>	<u>Attrib_type_Name</u>	Vts	<u>Unit_Type</u>	VTe	Value
-------------------	-------------------------	-----	------------------	-----	-------

ตาราง Element\_Relation

<u>Element_ID</u>	<u>Sub_Element_ID</u>	Vts	VTe
-------------------	-----------------------	-----	-----

ตาราง Element\_Properties

Element_type_Name	Start_tag	End_tag	Empty
-------------------	-----------	---------	-------

รูปที่ 3.4 ความสัมพันธ์ของฐานข้อมูลที่ได้จากรูปแบบเชิงเวลาของเวปเมตาเดต้า

### 3.8 การแมปโปงฐานข้อมูลเชิงเวลาที่ปรับปรุงมาเป็นตาราง

สามารถแมปโปงฐานข้อมูลจากการออกแบบในรูปของของตารางความสัมพันธ์ จากฐานข้อมูลเวปเมตาเดต้าเชิงเวลา ได้เป็น 7 ตารางดังต่อไปนี้

ตารางที่ 3.10 URL

Name	Type	Key	Meaning	Example
URL_ID	Number	Composite Key	หมายเลขของเวปเพจ	11070011
Domain_Name	Text	-	ชื่อโดเมนของเวปไซด์	www.google.co.th
Vts	Date-time	Composite Key	เวลาที่นำข้อมูลเข้าระบบ	2006-08-12
Vte	Date-time	-	เวลาที่สุดท้ายที่ข้อมูลคงอยู่	2007-12-31

ตารางที่ 3.11 Element\_Type\_Relation

Name	Type	Key	Meaning	Example
Element_type_Name	Text	Composite Key	ชื่อของอีลิเมนต์แต่ละประเภท	BODY
Sub_Element_type_Name	Text	Composite Key	ชื่อของอีลิเมนต์แต่ละประเภท	TABLE
Qty	Text	-	บอกรูปแบบของความสัมพันธ์ (1 = One , N = Many)	N

ตารางที่ 3.12 Content

Name	Type	Key	Meaning	Example
Element_ID	Number	Composite Key	หมายเลขของอีลิเมนต์แต่ละแท็ก	12050010
Element_type_Name	Text	-	ชื่อของอีลิเมนต์แต่ละประเภท	BODY
URL_ID	Number	-	หมายเลขของเวปเพจ	11070011
Order_Number	Number	-	ลำดับของอีลิเมนต์	165
Content_text	Text	-	ข้อมูลของแต่ละอีลิเมนต์	การทดลอง
Vts	Date-time	Composite Key	เวลาที่นำข้อมูลเข้าระบบ	2006-08-12
Vte	Date-time	-	เวลาที่สุดท้ายที่ข้อมูลคงอยู่	2007-12-31

ตารางที่ 3.13 Element\_type - Attribute\_Properties

Name	Type	Key	Meaning	Example
Element_type_Name	Text	Composite Key	ชื่อของอีลิเมนต์แต่ละประเภท	BODY
Attrib_type_Name	Text	Composite Key	ชื่อของแอททริบิวต์แต่ละประเภท	Width
Unit_Type	Text	Composite Key	ประเภทของข้อมูล (F = Fix Value, R = Relate Value, N = None )	F
Start_Value	Text	-	ค่าเริ่มต้นของแต่ละแอททริบิวต์ในแต่ละอี ลิเมนต์	#000000
Limit_Value	Text	-	ค่าจำกัดของแต่ละแอททริบิวต์ในแต่ละอี ลิเมนต์	#FFFFFF
Default_Value	Text	-	ค่ามาตรฐานของแต่ละแอททริบิวต์ในแต ละอีลิเมนต์	#FEFEFE

ตารางที่ 3.14 Element - Attribute\_Value

Name	Type	Key	Meaning	Example
Element_ID	Number	Composite Key	หมายเลขของอีลิเมนต์แต่ละแท็ก	12050010
Attrib_type_Name	Text	Composite Key	ชื่อของแอททริบิวต์แต่ละประเภท	Width
Unit_Type	Text	-	ประเภทของข้อมูล (F = Fix Value, R = Relate Value, N = None )	F
Value	Text	-	ค่าของแต่ละแอททริบิวต์ของอีลิเมนต์ในแต่ละแท็ก	30%
Vts	Date-time	Composite Key	เวลาที่นำข้อมูลเข้าระบบ	2006-08-12
Vte	Date-time	-	เวลาที่สุดท้ายที่ข้อมูลคงอยู่	2007-12-31

ตารางที่ 3.15 Element Relation

Name	Type	Key	Meaning	Example
Element_ID	Number	Composite Key	ID ของอีลิเมนต์แต่ละตัว	12
Sub_Element_ID	Number	Composite Key	ID ของอีลิเมนต์แต่ละตัว	13
Vts	Date-time	Composite Key	เวลาที่นำข้อมูลเข้าระบบ	2006-08-12
Vte	Date-time	-	เวลาที่สุดท้ายที่ข้อมูลคงอยู่	2007-12-31

ตารางที่ 3.16 Element Properties

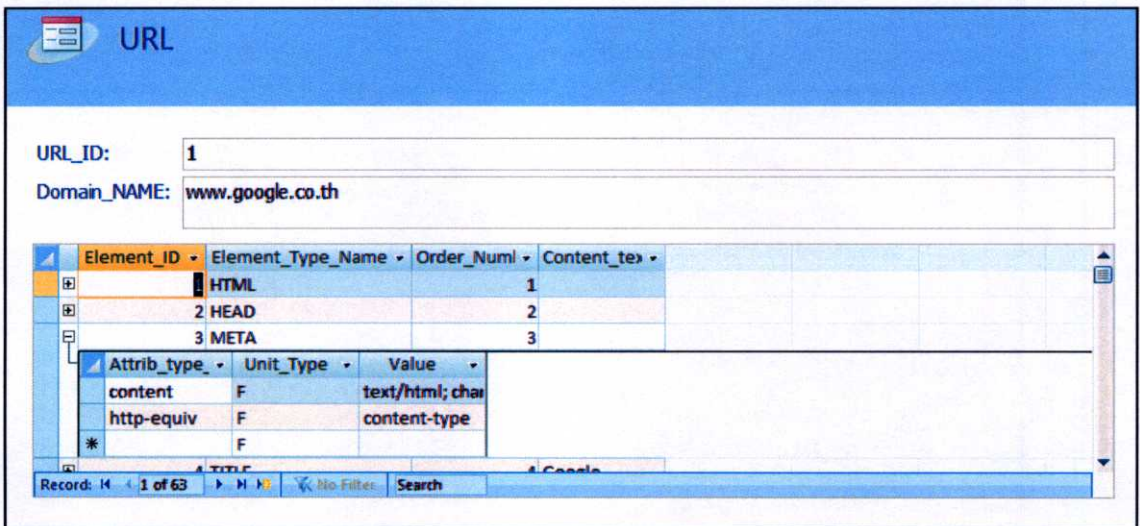
Name	Type	Key	Meaning	Example
Element_type_Name	Text	Primary Key	ชื่อของอีลิเมนต์แต่ละประเภท	TABLE
Start_Tag	Text	-	ต้องมีแท็กเริ่มหรือไม่ (Y = Must ,O = Option , F = Forbidden)	Y
End_Tag	Text	-	ต้องมีแท็กจบหรือไม่ (Y = Must ,O = Option , F = Forbidden)	F
Empty	Text	-	ในการใช้เป็นอีลิเมนต์ย่อยไม่มีแท็กนี้ได้ไหม (E = ไม่มีได้ ,N = ต้องมี )	E

## บทที่ 4

# การพัฒนาเว็บเมตาเดต้าเพื่อใช้ในการบริหารจัดการเว็บไซต์

### 4.1 หลักการพัฒนาและออกแบบโปรแกรมประยุกต์

จากการออกแบบฐานข้อมูลในบทที่ 3 ทางผู้จัดทำได้นำฐานข้อมูลที่ได้มาจากการออกแบบจากความสัมพันธ์จากการศึกษาเว็บเมตาเดต้ามาทดสอบ โดยการนำตัวอย่างข้อมูลที่เตรียมไว้ (ในการทดลองนี้เลือกใช้หน้าเว็บ [www.google.co.th](http://www.google.co.th) มาเป็นตัวอย่างในการทดลอง) มาจัดเก็บลงฐานข้อมูลผ่านโปรแกรม Microsoft Access เพื่อทดสอบความสามารถในการจัดเก็บเว็บเพจ ลงในฐานข้อมูลที่ได้ออกแบบไว้



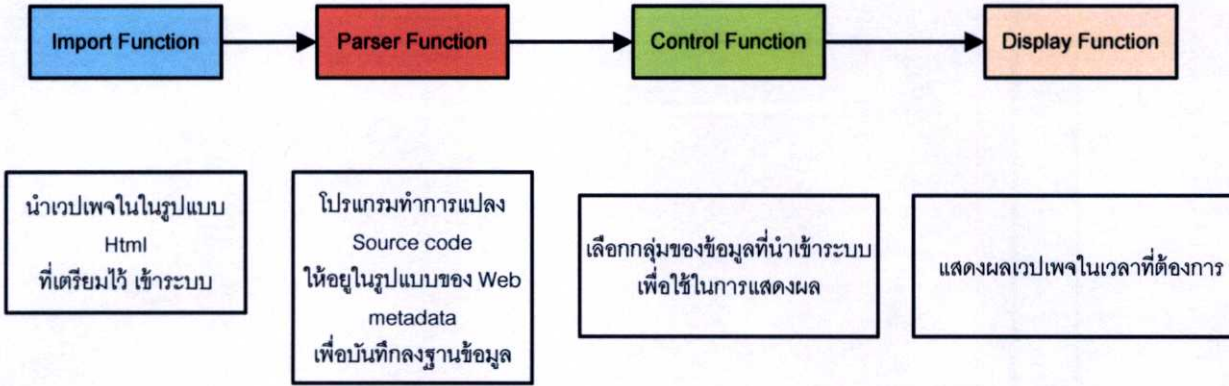
Element_ID	Element_Type_Name	Order_Num1	Content_tes
1	HTML	1	
2	HEAD	2	
3	META	3	

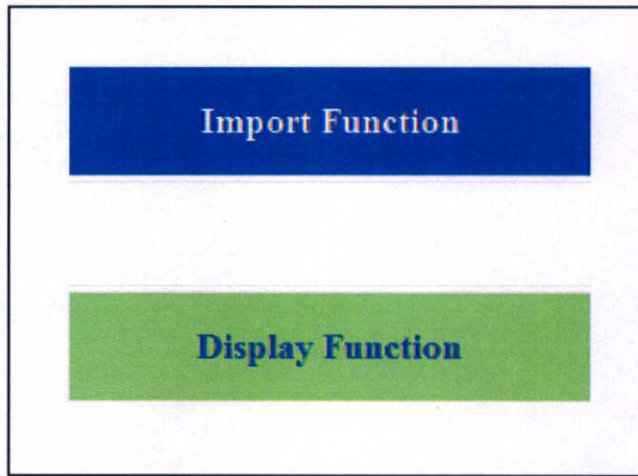
Attrib_type	Unit_Type	Value
content	F	text/html; charset=utf-8
http-equiv	F	content-type
*	F	

รูปที่ 4.1 โปรแกรม Microsoft Access สำหรับบันทึกโครงสร้าง HTML และข้อมูลในเว็บไซด์ลงฐานข้อมูล

หลังจากนั้นได้ทำการศึกษาหลักการและทำงานของการบริหารจัดการเว็บไซต์ในทั่วไป ทางผู้จัดทำจึงได้ออกแบบหลักการทำงานของ โปรแกรมที่นำมาเป็นต้นแบบในการทดลองในการบริหารจัดการเว็บไซต์จากเว็บเมตาเดต้าได้ดังต่อไปนี้ (ปรากฏตามรูปที่ 4.2)



รูปที่ 4.2 หลักการทำงานของระบบการบริหารจัดการเวปไซต์จากเวปเมตาเดต้า



รูปที่ 4.3 หน้าจอสำหรับการทำงาน

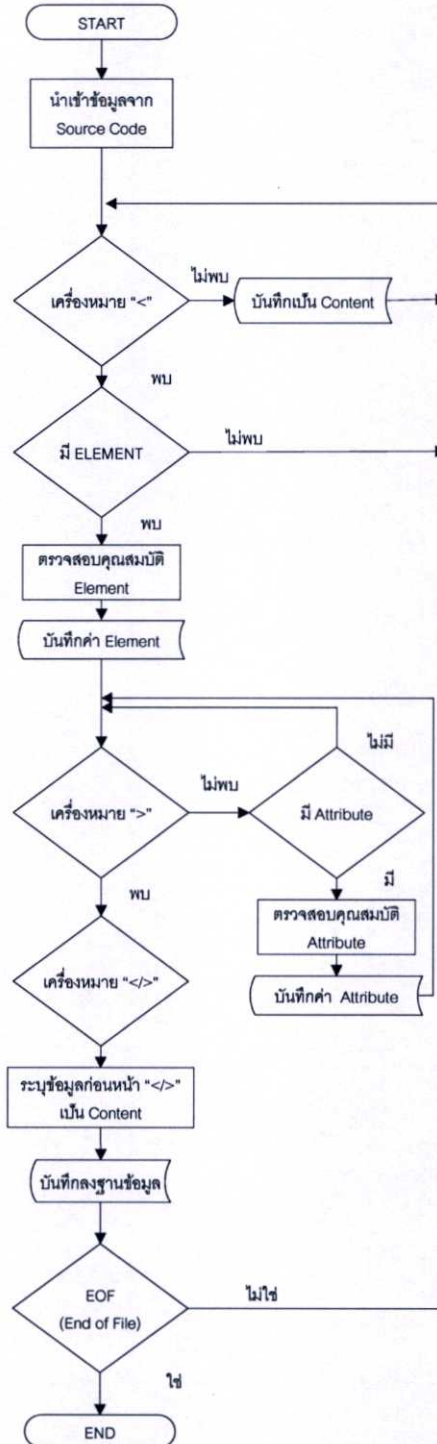
#### 4.1.1 ส่วนการนำเข้าข้อมูล (Import Function)

ส่วนนี้จะทำหน้าที่ในการนำเข้าข้อมูลเพื่อเป็นอินเทอร์เน็ตเฟสสำหรับส่งต่อไปให้ระบบตัดคำ (Parser function) โดยผู้ใช้งานต้องตั้งชื่อกลุ่มข้อมูลที่นำเข้าในรูปแบบไฟล์ HTML

รูปที่ 4.4 หน้าจอสำหรับโปรแกรมในส่วนการนำเข้าข้อมูล

#### 4.1.2 ส่วนการตัดคำ (Parser Function)

ในส่วนนี้จะทำหน้าที่ในการตัดคำ แยกคำในไฟล์ที่ได้มาให้อยู่ในรูปของอีลีเมนต์ แอททริบิวต์ ข้อมูล และค่าต่าง ๆ บันทึกลงในฐานข้อมูลตามที่ยกแบบไว้ โดยจะนำเอาแท็กที่พบในการนำเข้ามาตรวจสอบในฐานข้อมูลว่าเป็นแท็กประเภทไหน มีค่าเป็นอย่างไร มีข้อมูลเป็นอย่างไร จากนั้นก็บันทึกลงฐานข้อมูลตามที่ได้กำหนดไว้ สามารถอธิบายให้เห็นถึงการทำงานตามรูปแบบดังต่อไปนี้



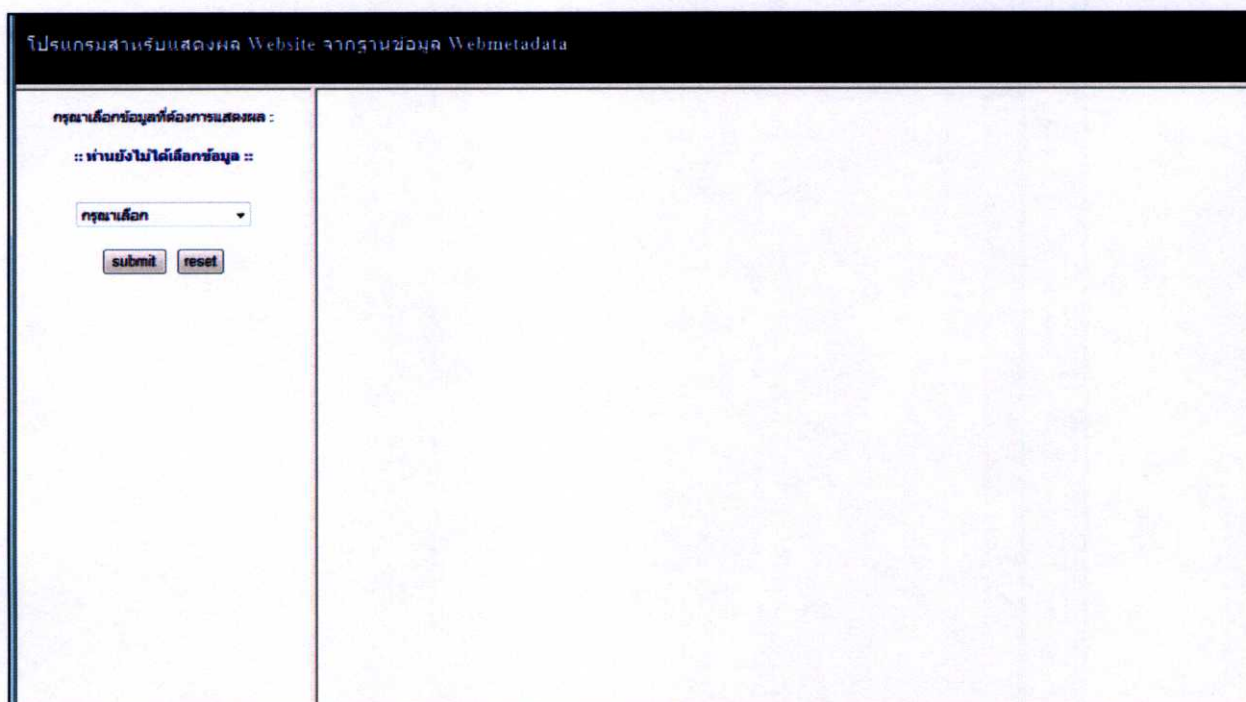
รูปที่ 4.5 โฟลว์ชาร์ตการทำงานของ Parser Function

### 4.1.3 ส่วนการควบคุม (Control Function)

ในส่วนการควบคุมนั้น จะมีหน้าที่ในการกำหนดกลุ่มของข้อมูลที่ต้องการส่งต่อไปยัง ส่วนการแสดงผล โดยในส่วนนี้จะทำหน้าที่คัดเลือกและทำการจัดกลุ่มของข้อมูลในช่วงเวลา หรือกลุ่มเวลาที่ต้องการ

### 4.1.4 ส่วนการแสดงผล (Display Function)

ในส่วนการแสดงผลนั้น มีหน้าที่ในการดึงข้อมูลในฐานข้อมูลตามที่ได้เลือกมาจากส่วน การควบคุม เพื่อนำมาแสดงผลจากการ Re-Generate เป็นเอกสาร HTML เพื่อมาใช้ในการ แสดงผลให้ได้จากเว็บเพจในช่วงเวลาที่กำหนด



รูปที่ 4.6 หน้าจอสำหรับ โปรแกรมในส่วนการแสดงผลข้อมูล

### 4.1.5 การเตรียมฐานข้อมูลเพื่อรองรับระบบ

จากฐานข้อมูลที่ได้ทำการออกแบบไว้ในบทที่ 3 ต้องมีการเตรียมฐานข้อมูลในส่วน ความสัมพันธ์ของเว็บเมตาเคด้าเพื่อประกอบการใช้งานระบบ เช่น การเตรียมข้อมูลในเรื่อง ความสัมพันธ์ระหว่างของอิลิเมนต์ ในตาราง ELEMENT\_TYPE\_RELATION คุณสมบัติของอิลิเมนต์ในตาราง ELEMENT\_PROPERTIES และคุณสมบัติของแอททริบิวท์ในแต่ละอิลิเมนต์ใน ตาราง ELEMENT\_ATTRIBUTE\_PROPERTIES โดยทั้งหมดมาจากการศึกษาความสัมพันธ์ตาม มาตรฐาน HTML 4.01 เพื่อกำหนดเป็นค่าเริ่มต้นเพื่อใช้ในการประมวลผล นำข้อมูลจากเว็บไซต์ เข้าระบบต่อไป ดังจะเห็นตัวอย่างได้จากตัวอย่างการเตรียมข้อมูลพื้นฐาน (ส่วนหนึ่ง) ในตาราง

Element\_Type\_Relation ตาราง Element\_Attribute\_Properties และตาราง Element\_Properties ในตารางที่ 4.1 -4.3 ตามลำดับ

ตารางที่ 4.1 การเตรียมข้อมูลพื้นฐาน เพื่อนำเข้าข้อมูลในตาราง Element\_Type\_Relation

ELEMENT_TYPE_NAME	SUB_ELEMENT_TYPE_NAME	QTY
ADDRESS	BODY	1
APPLET	BODY	1
BODY	ADDRESS	N

ตารางที่ 4.2 การเตรียมข้อมูลพื้นฐาน เพื่อนำเข้าข้อมูลในตาราง Element\_Attribute\_Properties

ELEMENT_TYPE_NAME	ATTRIB_TYPE_NAME	UNIT_TYPE	START_VALUE	LIMIT_VALUE	DEFAULT_VALUE
BODY	bgcolor	F	#000000	#####	#####
TABLE	high	R	1%	100%	100%
TABLE	width	R	1%	100%	100%

ตารางที่ 4.3 การเตรียมข้อมูลพื้นฐาน เพื่อนำเข้าข้อมูลในตาราง Element\_Properties

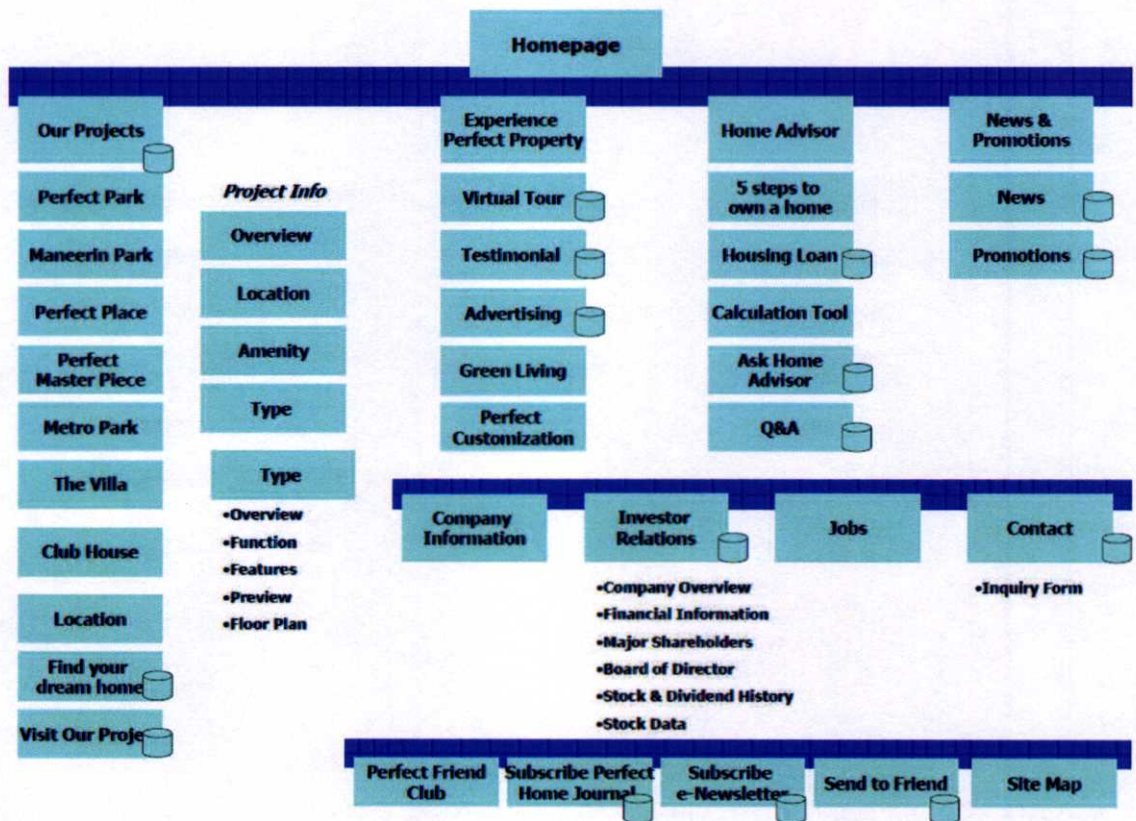
ELEMENT_TYPE_NAME	START_TAG	END_TAG	EMPTY
BODY	O	O	N
AREA	Y	F	E
TABLE	Y	Y	N

## 4.2 การเตรียมข้อมูลและทดลองการใช้งาน

จากที่กล่าวมาข้างต้น ทางผู้จัดทำได้ทำการนำตัวอย่างเว็บไซต์ [www.google.co.th](http://www.google.co.th) มาจำลองใช้ในการทดลองครั้งนี้ เนื่องจากสามารถทำให้เห็นภาพในการแสดงผลอย่างชัดเจน เพราะเว็บไซต์ [www.google.co.th](http://www.google.co.th) มีการเปลี่ยนแปลงข้อมูลในเว็บไซต์ที่เห็นได้ชัด โดยการเปลี่ยนรูปโลโก้ของ Google ตามวันสำคัญต่าง ๆ ในช่วงเวลานั้น เช่น วันทั่วไปเป็นรูปโลโก้ปกติ วันปีใหม่เป็นโลโก้ช่วงปีใหม่ เป็นต้น ด้วยเหตุนี้ทางผู้จัดทำจึงได้นำเอา source code ของเว็บไซต์ google ในช่วงเวลาที่แตกต่างกันมาใช้ในการทดลอง ซึ่งจะแสดงผลการทดลองที่เกิดขึ้นในบทที่ 5 ต่อไป

### 4.3 การนำระบบมาประยุกต์ใช้งานจริง

จากปัญหาในการบริหารจัดการเว็บไซต์ตามที่ได้กล่าวมาข้างต้น ซึ่งในการศึกษาและการออกแบบตามงานวิจัยนี้ได้ทดสอบนำมาพัฒนาใช้กับเว็บไซต์ของบริษัทแห่งหนึ่ง โดยนำผลที่ได้จากการศึกษาการออกแบบเว็บเมตาเดต้าสำหรับการบริหารจัดการเว็บไซต์ มาประยุกต์ใช้เพื่อช่วยแก้ปัญหาในการกำหนดช่วงเวลาในการเผยแพร่โปรโมชันการขายและข่าวสารของบริษัท โดยรองรับการวางแผนล่วงหน้าอัตโนมัติผ่านการจัดการด้วยเว็บเมตาเดต้า รวมไปถึงการแก้ไขปัญหาในการย้อนกลับมาดูเว็บไซต์ในอดีตในช่วงเวลาที่ต้องการได้ โดยทางผู้จัดทำได้ศึกษาข้อมูลโครงสร้างของเว็บไซต์ที่นำมาปรับปรุง ดังแสดงผลในรูปที่ 4.7



รูปที่ 4.7 ความสัมพันธ์ของโครงสร้างทั้งหมดของเว็บไซต์ที่ทำมาปรับปรุง

จากการปรับปรุงในครั้งนี้จะทำการปรับปรุงเฉพาะในหน้าแรก (Homepage) โดยจุดประสงค์เพื่อทำการควบคุมการแสดงผลของข่าวสารและโปรโมชันในหน้าแรก แทนการปรับปรุงข้อมูลโดยการใช้คนในการดูแล โดยได้ทำการศึกษารูปแบบของเว็บไซต์ดังจะเห็นได้จากรูปที่ 4.8

PROPERTY PERFECT

Company Information | Investor Relations | Job | Contact | Tel. 1375

Find Your Dream Home

PERFECT LIVING

Our Projects | Experience Property Perfect | Home Advisor | News & Promotions | Perfect Friend Club

experience property perfect

Perfect Masterpiece | Perfect Place | Perfect Park | The Villa | Metro Park | The Metro

Hot Promotion **ดูทั้งหมด**

What's New **ดูทั้งหมด**

Perfect Home Journal

Visit Our Project

© 2008 PROPERTY PERFECT PUBLIC COMPANY LIMITED. ALL RIGHTS RESERVED. | Site Map

Subscribe E-Newsletter

รูปที่ 4.8 หน้าเว็บไซต์ของบริษัทแห่งหนึ่งที่นำมาใช้ในการปรับปรุง

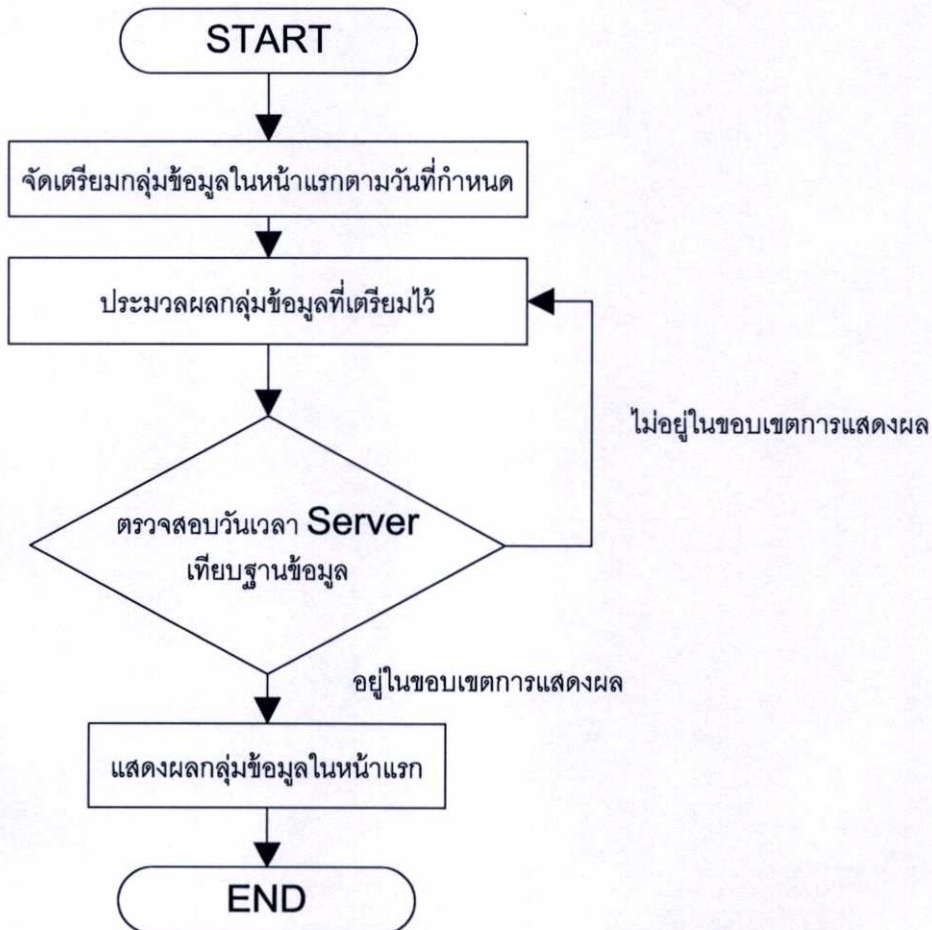


รูปที่ 4.9 โครงสร้างการนำเสนอข้อมูลข่าวสาร และ โปรโมชันในหน้าแรกของเว็บไซต์ที่  
ทำการศึกษา

จากการศึกษาโครงสร้างของหน้าเวปเพจของเดิมพบว่ามีการจัดวาง โพร โมชั่น และ ข่าวสารตามรูปแบบของหน้าเวปที่ได้จำลองออกมาตามรูปที่ 4.9 โดยการปรับปรุงครั้งนี้ทางผู้จัดทำจะทำการแปลงเวปเพจในหน้าแรกในรูปแบบเวปเมตาเคต้าตามการศึกษาตามผลวิจัย โดยมีขั้นตอนการปรับปรุงดังต่อไปนี้

1. ทำการแปลงหน้าแรกของเวปไซค์ให้อยู่ในรูปแบบของเวปเมตาเคต้าเชิงเวลา
2. ทำการสร้างระบบการบริหารจัดการหน้าแรกในรูปแบบเชิงเวลาเพื่อช่วยจัดการ โพร โมชั่น และข่าวสารเพื่อช่วยแสดงผลอัตโนมัติตามโครงสร้างในรูปที่ 4.9

แนวทางในการบริหารจัดการ โพร โมชั่นและข่าวสารนั้นของเวปไซค์นี้ ทางผู้จัดทำได้ทำการออกแบบระบบภายใต้รูปแบบของฐานข้อมูลเวปเมตาเคต้าเชิงเวลา โดยสร้างระบบจัดการจากการอ้างอิงวันที่ที่ต้องการนำเสนอ โพร โมชั่นหรือข่าวสาร เปรียบเทียบกับเวลาปัจจุบัน (เทียบจากเวลาของ เซิร์ฟเวอร์) ซึ่งสามารถอธิบายได้ตามโฟลชาร์ตตามรูปที่ 4.10 ได้ดังต่อไปนี้



รูปที่ 4.10 โฟลว์ชาร์ตแสดงการนำเสนอข้อมูลข่าวสารและ โพร โมชั่นในหน้าแรกของเวปไซค์


Homepage


Home > Homepage Management > Edit


Homepage Information


\* Publish Date: วันจันทร์  
22/12/2008


**Promotion Item**

\* Promotion 1: โปรโมชั่น 1  
 จัสมิ โปรโมชั่นลดเงิน 0%  
 Please select a promotion

Promotion 2: โปรโมชั่น 2  
 โปรโมชั่น ดอกเบี้ย 2% บน 2 ปี  
 Please select a promotion

Promotion 3: โปรโมชั่น 3  
  
 Please select a promotion


Promotion 4: โปรโมชั่น 4  
  
 Please select a promotion


Promotion 5: โปรโมชั่น 5  
  
 Please select a promotion

Display (Thai): แสดงผลหรือไม่ (ไทย)  
 Yes  No

\* Required fields

**News Item**

\* News 1: ข่าว 1  
 เปิดโครงการใหม่ล่าสุด กับ ราคาพิเศษ  
 Please select a news

\* News 2: ข่าว 2  
 Opening @ Beach Resort  
 Please select a news

**Flash Animation**

Flash File:   
  
 (File [\*.swf] only, Size 990 x 620 px, Max. 3 MB)

รูปที่ 4.11 หน้าจอส่วนโปรแกรมการจัดการกลุ่มข่าวสารและโปรโมชันที่ต้องการแสดงผล

Homepage

Home > Homepage Management > Search

Publish Date:  to

Search Result 4 item(s)

Display # 10 Rows   Page 1 of 1

<input type="checkbox"/>	Publish Date	Current	Create Date
<input checked="" type="checkbox"/>	27/02/2009		2/26/2009 10:53:36 PM
<input type="checkbox"/>	01/02/2009		2/26/2009 10:52:23 PM
<input checked="" type="checkbox"/>	29/01/2009		2/17/2009 11:16:01 PM
<input type="checkbox"/>	22/12/2008	Show Now	10/13/2008 4:21:08 PM

รูปที่ 4.12 หน้าจอแสดงส่วนสถานะในการนำเสนอของกลุ่มข้อมูลข่าวสารและโปรโมชัน

การทำงานของระบบนั้น ยึดหลักจากการอ้างอิงวันที่ของเซิร์ฟเวอร์เทียบกับกลุ่มของข้อมูล ทั้งข่าวสารและโปรมโซชันที่ได้ทำการเตรียมข้อมูลไว้แล้ว โดยกลุ่มของข้อมูลนั้นจะทำการระบุวันที่ที่ต้องการให้นำกลุ่มข้อมูลนั้นมาแสดงผล โดยยึดหลักว่าจะแสดงเฉพาะกลุ่มข้อมูลที่อยู่ในขอบเขตของการแสดงผลเท่านั้น เช่น หากเป็นวันที่ยังมาไม่ถึงก็จะยังไม่แสดงผล หากไม่มีกลุ่มข้อมูลใดมาแทนให้นำเสนอกลุ่มข้อมูลล่าสุด ซึ่งสามารถแสดงให้เห็นหน้าจอการทำงานในส่วน ของโปรแกรมเพื่อบันทึกสำหรับจัดการกลุ่มข่าวสารและโปรมโซชันที่ต้องการแสดงผลได้ตามรูป ที่ 4.11 และส่วนโปรแกรมแสดงสถานะในการนำเสนอของกลุ่มข้อมูลข่าวสารและโปรมโซชัน 4.12 ในส่วนของผลการดำเนินการที่ได้จากการทดลองในระบบนี้นั้นจะกล่าวถึงในบทที่ 5 ต่อไป

# บทที่ 5

## ผลการทดลอง

ในบทที่ 5 นี้จะเป็นการนำฐานข้อมูลที่ออกแบบมาทดลองการบันทึกข้อมูลจากเว็บไซต์ลงฐานข้อมูลเว็บเมตาเดต้าที่ได้ออกแบบไว้ และแสดงการผลทดลองในขั้นตอนต่าง ๆ เพื่อให้เห็นกระบวนการและแนวทางในการนำไปใช้ต่อไปในอนาคต รวมไปถึงการนำมาประยุกต์ใช้ร่วมกับโปรแกรมต้นแบบที่ได้ออกแบบไว้ ดังนี้

### 5.1 การทดสอบความสามารถของฐานข้อมูลที่ออกแบบ

การทดลองในครั้งนี้ได้นำ Source Code ของหน้าแรกเว็บไซต์จาก www.google.co.th มาใช้ในการทดลอง โดยการบันทึกข้อมูลลงฐานข้อมูลตามที่ได้ออกแบบไว้ข้างต้นผ่านโปรแกรม Microsoft Access แล้วนำเอาค่าที่บันทึกได้ในฐานข้อมูลกลับมาสร้างเว็บเพจใหม่

#### 5.1.1 การบันทึก source Code จากเว็บเพจลงฐานข้อมูล

- สามารถบันทึก Source Code ของเว็บเพจลงฐานข้อมูลได้
- สามารถนำค่าที่ได้จากการบันทึกมา กลับมาสร้างเว็บเพจได้ โดยไม่ผิดไปจากเดิม



ต้นฉบับ



สร้างจากฐานข้อมูล

รูปที่ 5.1 การเปรียบเทียบจากเว็บต้นฉบับ และนำกลับมาสร้างเว็บเพจจากฐานข้อมูล

ตารางที่ 5.1 ตัวอย่างการบันทึกข้อมูลลงฐานข้อมูลที่ออกแบบไว้ใน ตาราง URL

URL_ID	DOMAIN_NAME
1	www.google.co.th

ตารางที่ 5.2 ตัวอย่างการบันทึกข้อมูลลงในฐานข้อมูลที่ออกแบบไว้ในตาราง CONTENT

ELEMENT_ID	URL_ID	ELEMENT_TYPE_NAME	ORDER_NUMBER	CONTENT_TEXT
1	1	HTML	1	
2	1	HEAD	2	
3	1	META	3	
4	1	TITLE	4	Google
...	...	...	...	...
63	1	FONT	63	&copy;2007 Google

ตารางที่ 5.3 ตัวอย่างการบันทึกข้อมูลในฐานข้อมูลที่ออกแบบในตาราง Element\_attribute\_value

ELEMENT_ID	ATTRIB_TYPE_NAME	UNIT_TYPE	VALUE
3	content	F	text/html; charset=UTF-8
3	http-equiv	F	content-type
7	bgcolor	F	#ffffff
...	...	...	...
9	style	F	background:url(/intl/en_com/images/ logo_plain.png) no-repeat; height:110px;width:276px
...	...	...	...
63	size	F	-2

### 5.1.2 ทดลองปรับเปลี่ยนค่าที่บันทึกในฐานข้อมูล

ปรับเปลี่ยนค่าข้อมูลที่บันทึกไว้ในฐานข้อมูลโดยตรง เช่น เปลี่ยนแปลงข้อมูล สลับตำแหน่ง เพิ่มข้อมูล และลบข้อมูล แล้วนำฐานข้อมูลที่ได้รับการแก้ไขข้อมูลแล้วแสดงผลกลับมาเป็นเวปเพจ

- บันทึกข้อมูลในตารางต่าง ๆ ที่นำเสนอในที่นี้ได้ยกตัวอย่างตารางที่เกี่ยวข้องที่มีผลต่อการเปลี่ยนรายละเอียดของเวปโดยตรง ตาราง Content และ ตาราง Element\_Attribute\_Value แสดงเฉพาะส่วนที่เกี่ยวข้องกับการทดลองเปลี่ยนแปลงข้อมูล

- แปลงฐานข้อมูลที่ได้ออกมาในรูปของเวปเพจได้ดังรูปที่ 5.2

- สามารถรองรับการเปลี่ยนแปลงข้อมูล สลับตำแหน่ง เพิ่มข้อมูล และลบข้อมูลในฐานข้อมูล แล้วนำมาแสดงผลจากสร้างเวปเพจจากฐานข้อมูลได้ โดยทำการแก้ไขฐานข้อมูลใน

ตาราง Content (ตารางที่ 5.4) และตาราง Element\_Attribute\_Value (ตารางที่ 5.4) เห็นการเปลี่ยนแปลงได้จากตารางที่ 5.6 และตารางที่ 5.7

ตารางที่ 5.4 ข้อมูลที่บันทึกในฐานข้อมูลก่อนเปลี่ยนแปลงในตาราง Content

Content				
Element_ID	URL_ID	Element_Type_Name	Order_Number	Content_text
4	1	TITLE	4	Google
10	1	DIV	10	ประเทศไทย
32	1	INPUT	32	
33	1	INPUT	33	
61	1	SCRIPT	61	<pre> &lt;!--if (!hp.isHomePage('http ://www.google.co.th') ) {document.write("&lt;p&gt; &lt;a href='/mgyp.html'\" onClick='\"style.behav ior='url(#default#hom epage)';setHomePage( 'http://www.google.co .th/');\"&gt;ตั้ง Google เป็นหน้าแรกของคุณ &lt;/a&gt;");}</pre>

ตารางที่ 5.5 ข้อมูลที่บันทึกในฐานข้อมูลก่อนเปลี่ยนแปลงในตาราง Element\_Attribute\_Value

Element_Attribute_Value			
Element_ID	Attrib_type_Name	Unit_Type	Value
9	Style	F	background:url(/intl/en_com/images/logo_plain.g no-repeat;height:110px;width:276px

ตารางที่ 5.6 ข้อมูลที่ทำการทดลองเปลี่ยนแปลงในตาราง Content

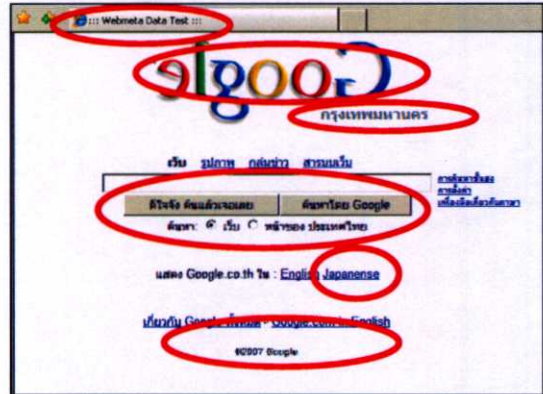
Content					สิ่งที่แก้ไข
Element_ID	URL_ID	Element_Type_Name	Order_Number	Content_text	
4	1	TITLE	4	::: Webmeta Data Test :::	เปลี่ยนจาก Google
10	1	DIV	10	กรุงเทพมหานคร	เปลี่ยนจาก ประเทศไทย
32	1	INPUT	33		สลับตำแหน่ง
33	1	INPUT	32		สลับตำแหน่ง
64	1	A	53	Japanese	เพิ่ม record + ลำดับข้อมูลใหม่
61	1	SCRIPT	61	//<!--if (!hp.isHomePage('http://www.google.co.th/')) {document.write("<p><a href='\"/mgyhp.html\"' onClick='\"style.behavior='url(#default#homepage)';setHomePage('http://www.google.co.th/');\">ตั้ง Google เป็นหน้าแรกของคุณ</a>");}	ลบ Record

ตารางที่ 5.7 แสดงข้อมูลที่ทำการทดลองเปลี่ยนแปลงในตาราง Element\_Attribute\_Value

Element_Attribute_Value				สิ่งที่แก้ไข
Element_ID	Attrib_type_Name	Unit_Type	Value	
9	style	F	background:url(/intl/en_com/imag es/logo_plain2.png) no- repeat;height:110px;width:276px	เปลี่ยน File จาก logo_pain.png เป็น logo_pain2.png
64	href	F	http://www.google.co.jp	เพิ่ม record



ต้นฉบับ



สร้างจากฐานข้อมูล

รูปที่ 5.2 แสดงการเปรียบเทียบจากเว็บต้นฉบับ และการ Re-Generate จากฐานข้อมูล

## 5.2 ทดสอบความสามารถของข้อมูลเมื่อเพิ่มความสามารถเชิงเวลา

### 5.2.1 ทดสอบการบันทึกฐานข้อมูลในรูปแบบเชิงเวลาในช่วงเวลาต่าง ๆ

ปรับเปลี่ยนรูปแบบของฐานข้อมูลเป็นแบบเชิงเวลาตามที่ได้ออกแบบไว้ในบทที่ 3 จากนั้นทดลองบันทึกเว็บเพจลงฐานข้อมูลในเวลาต่าง ๆ กัน ในที่นี้ใช้หน้าแรกของเว็บ [www.google.co.th](http://www.google.co.th) ต่างช่วงเวลา มาใช้ในการทดลอง

ช่วงที่ 1 ในวันที่ 15-06-2006

ช่วงที่ 2 ในวันที่ 12-08-2006

ช่วงที่ 3 ในวันที่ 01-01-2007

ช่วงที่ 4 ในวันที่ 20-01-2007

ทำแปลงฐานข้อมูลที่ได้รับการแก้ไขข้อมูลแล้วออกมาเป็นเว็บเพจในช่วงเวลาต่าง ๆ

ผลการบันทึกข้อมูลในตารางต่าง ๆ ที่นำเสนอในที่นี้ได้ยกตัวอย่างตารางที่เกี่ยวข้องที่มีผลต่อการเปลี่ยนรายละเอียดของเว็บโดยตรง ตาราง Content และ ตาราง Element\_Attribute\_Value แสดงเฉพาะส่วนที่เกี่ยวข้องกับการทดลอง โดยบันทึกเฉพาะส่วนที่มีการเปลี่ยนแปลงตามแนวคิดการบันทึกฐานข้อมูลแบบ Valid Time

ตารางที่ 5.8 การบันทึกในฐานข้อมูลในตาราง Content ในรูปเชิงเวลา

ELEMENT_ID	URL_ID	ELEMENT_TYPE_NAME	ORDER_NUMBER	CONTENT_TEXT	VTs	VTe
1	1	HTML	1		2006-06-15	9999-12-31
2	1	HEAD	2		2006-06-15	9999-12-31
3	1	META	3		2006-06-15	9999-12-31
4	1	TITLE	4	Google	2006-06-15	9999-12-31
...	...	...	...	...	...	...
9	1	DIV	9		2006-06-15	9999-12-31

ตารางที่ 5.9 การบันทึกในฐานข้อมูลในตาราง Element\_Attribute\_Value ในรูปเชิงเวลา

ELEMENT_ID	ATTRIB_TYPE_NAME	UNIT_TYPE	VALUE	VTs	VTe
3	content	F	text/html; charset=UTF-8	2006-06-15	9999-12-31
3	http-equiv	F	content-type	2006-06-15	9999-12-31
7	bgcolor	F	#ffffff	2006-06-15	9999-12-31
...	...	...	...	...	...
9	style	F	background:url(/intl/en_com/images/logo_plain.png) no-repeat; height:110px;width:276px	2006-06-15	9999-12-31
9	style	F	background:url(/intl/en_com/images/mothers_day06.png) no-repeat; height:110px;width:276px	2006-08-12	9999-12-31
9	style	F	background:url(/intl/en_com/images/newyear07.png) no-repeat; height:110px;width:276px	2007-01-01	9999-12-31
9	style	F	background:url(/intl/en_com/images/logo_plain.png) no-repeat; height:110px;width:276px	2007-01-20	9999-12-31

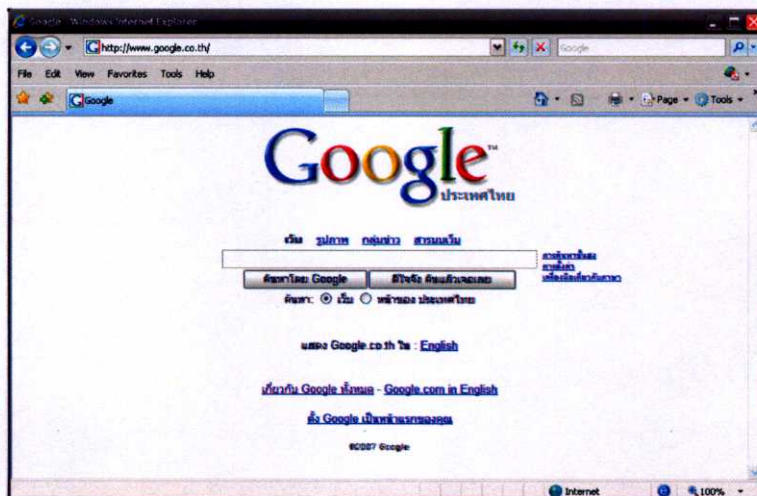
1

2

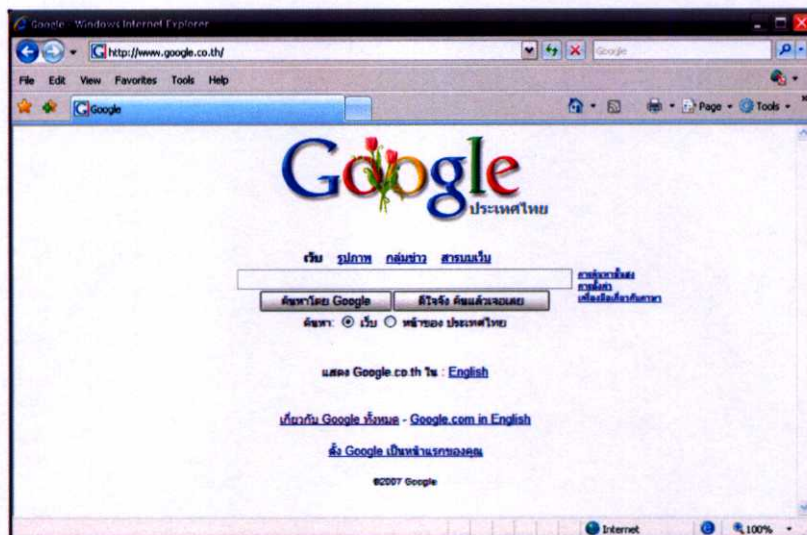
3

4

จากนั้นทำการแปลงฐานข้อมูลที่ได้ออกมาในรูปของเว็บเพจกลับมา ณ ช่วงเวลาต่าง ๆ ได้ดังรูปต่อไปนี้



รูปที่ 5.3 เว็บเพจที่เกิดจากการสร้างจากฐานข้อมูลในวันที่ 15-06-2006 และวันที่ 20-01-2007



รูปที่ 5.4 เว็บเพจที่เกิดจากการสร้างจากฐานข้อมูลในวันที่ 12-08-2006



รูปที่ 5.5 เว็บเพจที่เกิดจากการสร้างจากฐานข้อมูลในวันที่ 01-01-2007

### 5.3 การทดลองการแสดงผลจากการใช้โปรแกรมต้นแบบ

การทดลองในครั้งนี้ได้นำโปรแกรมต้นแบบที่ได้รับการออกแบบมาทดลองใช้แทนการนำเข้าทางฐานข้อมูลโดยตรง ซึ่งได้ผลการทดลองดังต่อไปนี้

#### 5.3.1 การนำเว็บเพจลงฐานข้อมูล

สามารถนำเข้าข้อมูลได้ตามที่ได้ออกแบบรูปแบบในการนำเข้าแบบเชิงเวลาได้ เช่นเดียวกับการบันทึกฐานข้อมูลจากโปรแกรม Microsoft Access

ตารางที่ 5.10 การบันทึกในฐานข้อมูลในตาราง Content ในรูปเชิงเวลาจากโปรแกรมต้นแบบ

ELEMENT_ID	URL_ID	ELEMENT_TYPE_NAME	ORDER_NUMBER	CONTENT_TEXT	VTs	VTe
1	1	HTML	1		2009-01-20	9999-12-31
2	1	HEAD	2		2009-01-20	9999-12-31
3	1	META	3		2009-01-20	9999-12-31
4	1	TITLE	4	Google	2009-01-20	9999-12-31
...	...	...	...	...	...	...
9	1	DIV	9		2009-01-20	9999-12-31

ตารางที่ 5.11 การบันทึกในฐานข้อมูลในตาราง Element\_Attribute\_Value ในรูปเชิงเวลาจากโปรแกรมต้นแบบ

ELEMENT_ID	ATTRIB_TYPE_NAME	UNIT_TYPE	VALUE	VTs	VTe
3	content	F	text/html; charset=UTF-8	2009-01-20	9999-12-31
3	http-equiv	F	content-type	2009-01-20	9999-12-31
7	bgcolor	F	#ffffff	2009-01-20	9999-12-31
...	...	...	...	...	...
9	style	F	background:url(/intl/en_com/images/logo_plain.png) no-repeat; height:110px;width:276px	2009-01-20	9999-12-31

### 5.3.2 การนำเวปเพจต่างช่วงเวลาลงฐานข้อมูล

สามารถนำเข้าข้อมูลได้ตามที่ได้ออกแบบรูปแบบในการนำเข้าแบบเชิงเวลาได้ ในหลายช่วงเวลาเช่นเดียวกับการบันทึกฐานข้อมูลจากโปรแกรม Microsoft Access

ช่วงที่ 1 ในวันที่ 20-01-2009

ช่วงที่ 2 ในวันที่ 01-01-2009

ช่วงที่ 3 ในวันที่ 20-07-2007

ช่วงที่ 4 ในวันที่ 01-01-2007

ช่วงที่ 5 ในวันที่ 12-08-2006

ช่วงที่ 6 ในวันที่ 15-06-2006

ตารางที่ 5.12 การบันทึกในฐานข้อมูลในตาราง Content ในรูปเชิงเวลาในเวลาต่าง ๆ

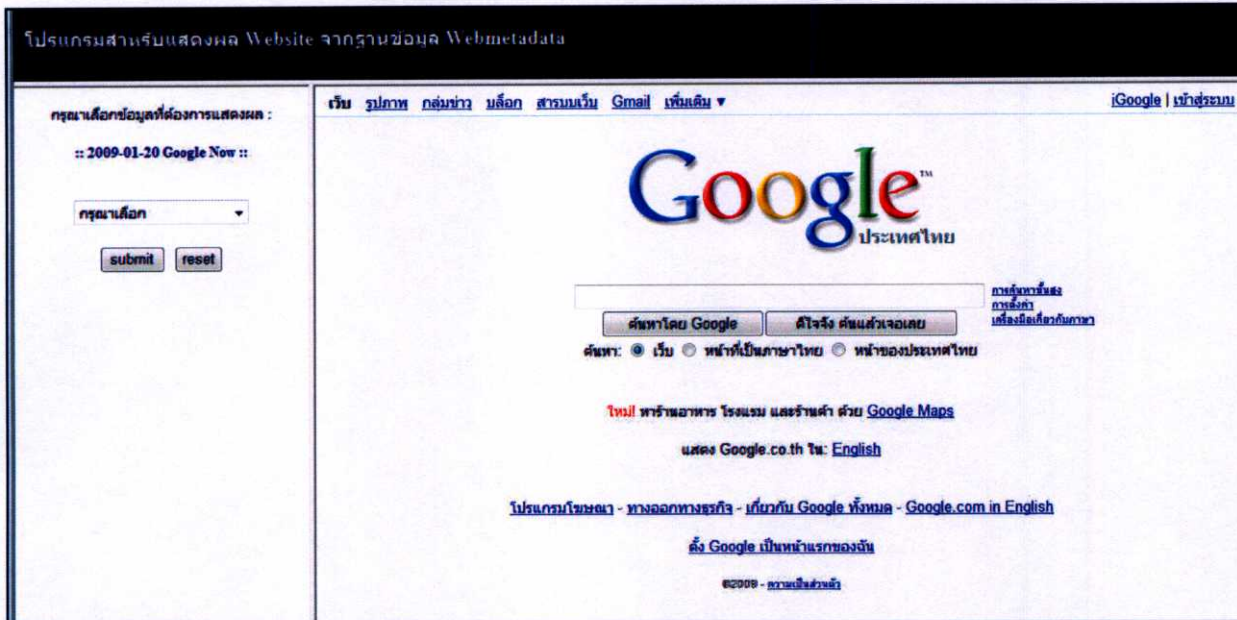
ELEMENT_ID	URL_ID	ELEMENT_TYPE_NAME	ORDER_NUMBER	CONTENT_TEXT	VTs	VTe
1	1	HTML	1		2009-01-20	9999-12-31
2	1	HEAD	2		2009-01-20	9999-12-31
3	1	META	3		2009-01-20	9999-12-31
4	1	TITLE	4	Google	2009-01-20	9999-12-31
...	...	...	...	...	...	...
9	1	DIV	9		2009-01-20	9999-12-31

ตารางที่ 5.13 การบันทึกในฐานข้อมูลในตาราง Element\_Attribute\_Value ในรูปเชิงเวลาในเวลาต่าง ๆ

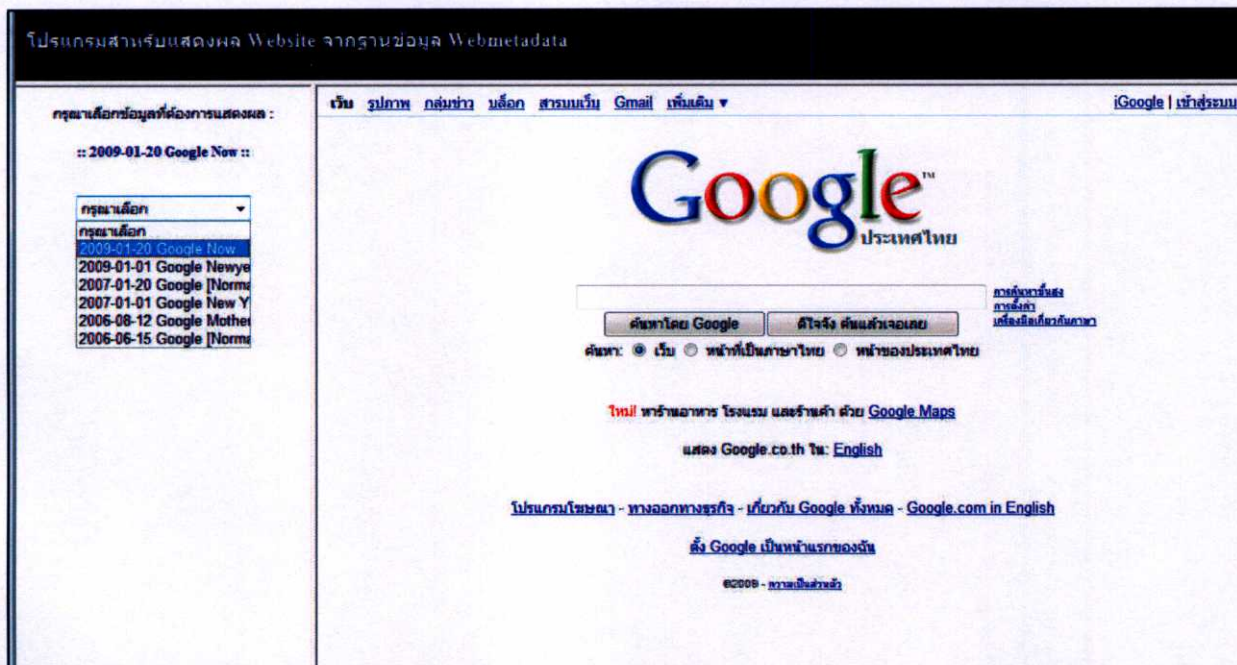
ELEMENT_ID	ATTRIB_TYPE_NAME	UNIT_TYPE	VALUE	VTs	VTe
3	content	F	text/html; charset=UTF-8	2009-01-20	9999-12-31
3	http-equiv	F	content-type	2009-01-20	9999-12-31
7	bgcolor	F	#ffffff	2009-01-20	9999-12-31
...	...	...	...	...	...
9	style	F	background:url(/intl/en_com/images/logo_plain.png) no-repeat; height:110px;width:276px	2009-01-20	9999-12-31
9	style	F	background:url(/intl/en_com/images/newyear09.png) no-repeat; height:110px;width:276px	2009-01-01	9999-12-31
9	style	F	background:url(/intl/en_com/images/logo_plain.png) no-repeat; height:110px;width:276px	2007-01-20	9999-12-31
9	style	F	background:url(/intl/en_com/images/newyear07.png) no-repeat; height:110px;width:276px	2007-01-01	9999-12-31
9	style	F	background:url(/intl/en_com/images/mothers_day06.png) no-repeat; height:110px;width:276px	2006-08-12	9999-12-31
9	style	F	background:url(/intl/en_com/images/logo_plain.png) no-repeat; height:110px;width:276px	2006-06-15	9999-12-31

### 5.3.3 การแสดงผลเว็บไซต์ผ่านจากโปรแกรมต้นแบบ

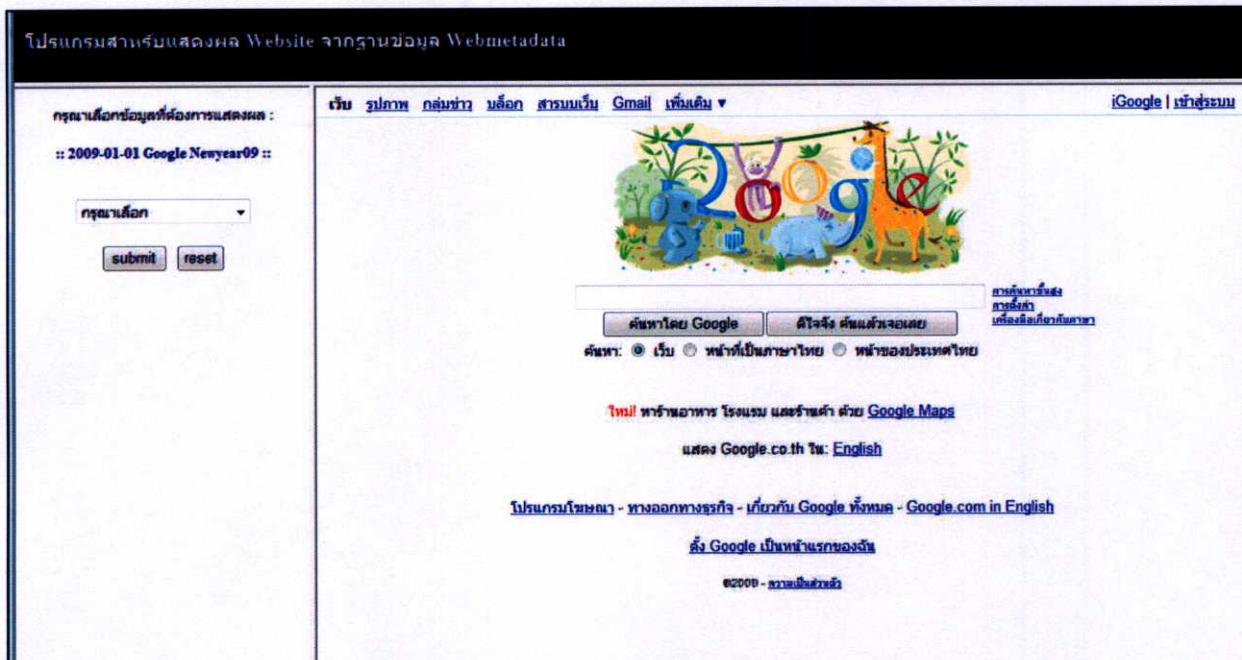
สามารถแสดงผลเว็บไซต์ผ่านจากโปรแกรมต้นแบบได้ โดยในการทดลองที่ใช้ข้อมูลที่น่าเข้าจากการทดลอง 5.3.2 โดยในการทดลองที่ใช้ข้อมูลที่น่าเข้าจากการทดลอง 5.3.2 ดังจะเห็นได้จากรูป ที่ 5.6 -5.9



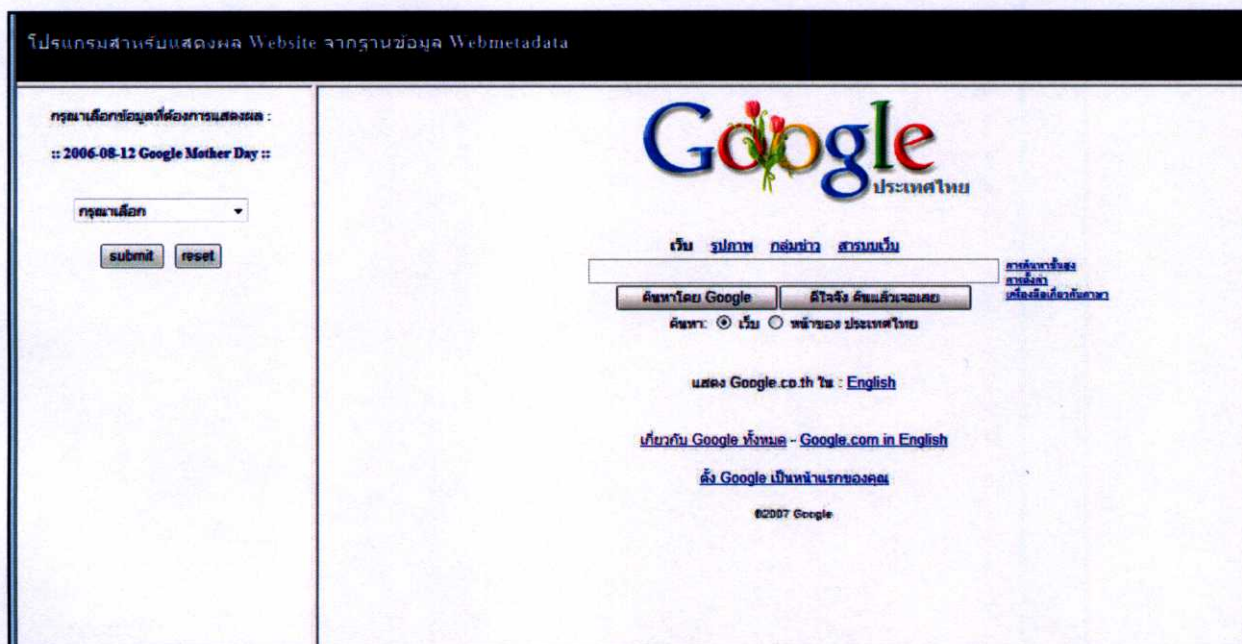
รูปที่ 5.6 ผลการนำเข้าในกลุ่ม 2009-01-20 Google Now



รูปที่ 5.7 การทำงานเมื่อต้องการเลือกแสดงผลกลุ่มข้อมูลอื่น ๆ



รูปที่ 5.8 ผลการนำเข้าไปในกลุ่ม 2009-01-1 Google Newyear09



รูปที่ 5.9 ผลการนำเข้าไปในกลุ่ม 2006-08-12 Google Mother Day

## 5.4 การนำฐานข้อมูลเชิงเวลาไปประยุกต์ใช้

### 5.4.1 การทดลองการแสดงผลในรูปแบบกำหนดการแสดงผลล่วงหน้า

จากการทดลองนำฐานข้อมูลต้นแบบที่ได้ออกแบบมาประยุกต์ใช้งานกับเว็บไซต์แห่งหนึ่งพบว่าสามารถใช้ประโยชน์จากการกำหนดเวลาล่วงหน้าในการแสดงผลได้ ดังจะแสดงให้เห็นได้จากการกำหนดกลุ่มข้อมูลของหน้าแรกจากระบบบริหารจัดการข้อมูล (ตามสถานะในรูปที่ 5.10) ที่ต้องการแสดงผล โปรโมชันในการขายต่าง ๆ ตัวอย่างเช่น ณ วันที่ 22/12/2008 ได้ทำการสร้างกลุ่มข้อมูล โปรโมชันดังรูปที่ 5.11 ซึ่งเมื่อมาดูที่หน้าแรกของเว็บไซต์จะเห็นได้ว่าโปรโมชันมีการเปลี่ยนแปลงไปตามที่กำหนดการแสดงผลไว้ ซึ่งจะเห็นได้จากรูปที่ 5.12 ในขณะเดียวกันก็จัดกลุ่มโปรโมชันและข่าวสารที่ต้องการจะนำเสนอในวันที่ 1/2/2009 ไว้ล่วงหน้าโดยกำหนดรายละเอียดของข่าวสารที่ต้องการจะนำเสนอตามรูปที่ 5.13 หลังจากนั้นเมื่อวันที่ของเซิร์ฟเวอร์ได้เปลี่ยนไปเป็นวันที่ 1/2/2009 พบว่าหน้าเว็บเพจหน้าแรกของเว็บไซต์แห่งนี้ ได้ทำการเปลี่ยนแปลง โปรโมชันและข่าวสารไปตามที่ได้กำหนดไว้ล่วงหน้าอย่างถูกต้อง ดังจะเห็นได้จากรูปที่ 5.14 และจะเห็นได้ว่าหน้าระบบแสดงสถานะของการแสดงผลกลุ่มข้อมูลเปลี่ยนไปเป็นกลุ่มของข้อมูลในวันที่ 01/02/2009 ตามรูปที่ 5.15 แทน

Homepage

Home > Homepage Management > Search Create Delete

**Publish Date:**

to  Search Reset

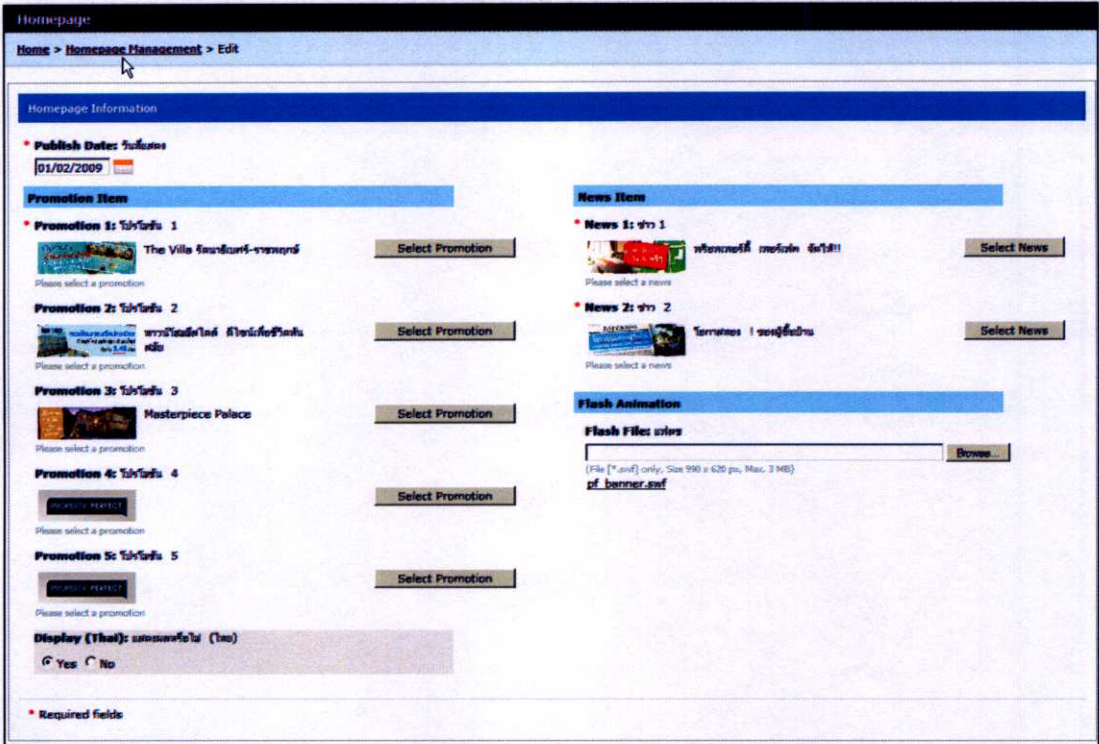
**Search Result** 4 item(s)

Display #  Rows Start Prev Page 1 of 1 Next End

<input type="checkbox"/>	Publish Date	Current	Create Date
<input type="checkbox"/>	27/02/2009		2/26/2009 10:53:36 PM
<input type="checkbox"/>	01/02/2009		1/30/2009 10:19:55 PM
<input type="checkbox"/>	29/01/2009		1/29/2009 9:57:34 PM
<input type="checkbox"/>	22/12/2008	Show Now	10/13/2008 4:21:08 PM

รูปที่ 5.10 หน้าระบบแสดงสถานะของการแสดงผลของกลุ่มข้อมูลที่กำหนดให้แสดงในช่วงวันเวลาต่าง ๆ ในขณะนี้แสดงผลกลุ่มข้อมูลในวันที่ 22/12/2008





รูปที่ 5.13 กลุ่มโปรโมชันที่ต้องการให้แสดงตั้งแต่วันที่ 1/2/2009



รูปที่ 5.14 หน้าแรกของเว็บไซต์เมื่อเวลาของเซิร์ฟเวอร์เป็นวันที่ 1/2/2009

Homepage

Home > Homepage Management > Search Create Delete

**Publish Date:**  
 to  Search Reset

**Search Result** 4 item(s)

Display #  Rows   Page  of 1

<input type="checkbox"/> Publish Date	Current	Create Date
<input checked="" type="checkbox"/> 27/02/2009		2/26/2009 10:53:36 PM
<input type="checkbox"/> 01/02/2009	Show Now	1/30/2009 10:19:55 PM
<input checked="" type="checkbox"/> 29/01/2009		1/29/2009 9:57:34 PM
<input checked="" type="checkbox"/> 22/12/2008		10/13/2008 4:21:08 PM

รูปที่ 5.15 หน้าระบบแสดงสถานะของการแสดงผลเมื่อเวลาของเซิร์ฟเวอร์แสดงผลวันที่ 1/2/2009

#### 5.4.2 การทดลองการแสดงผลในรูปแบบการดูย้อนหลังใน ช่วงเวลาที่ต้องการ

จากการทดลองพบว่าในหลักการเดียวกันสามารถใช้การปรับเปลี่ยนวันที่ของเซิร์ฟเวอร์ เพื่อย้อนดูการแสดงผลในช่วงเวลาอื่น ตามเวลาที่ต้องการได้ ตัวอย่างเช่น ปัจจุบัน (1/2/2009) หน้าแรกของเว็บไซต์แสดงผลดังรูปที่ 5.14 และในรายละเอียดมีการสร้างกลุ่มข้อมูล โปร โมชันดังรูปที่ 5.13

PROPERTY PERFECT Company Information | Investor Relations | Job | Contact | Tel. 1375

Find Your Dream Home

# 25<sup>th</sup>

ANNIVERSARY

## PROPERTY PERFECT

Our Projects | Experience Property Perfect | Home Advisor | News & Promotions | Perfect Friend Club

experience property perfect

**Hot Promotion**

**แปลงดี มีจำกัด**  
18-26 ธ.ค. นี้ โฉมดีสุดแล้ว...

**What's New**

**ประกวด Perfect Photo Contest**

ประกวด Perfect Photo Contest...  
รางวัลชนะเลิศจากคณะกรรมการประกวด...  
จาก Perfect Asia Hills Thailand

**Perfect Home Journal**

ดูภาพสวยๆ โครงการต่างๆ

**Visit Our Project**

ชมแบบโครงการที่...  
จาก แปลงใหม่ครบถ้วนแน่นอน

© 2009 Property Perfect Public Company Limited. All Rights Reserved. | Site Map Subscribe E-Newsletter

รูปที่ 5.16 หน้าแรกของเว็บไซต์เมื่อเซิร์ฟเวอร์แสดงผลในช่วงวันที่ 29/1/2009

ในการทดลองนี้ต้องการจะทราบว่าหน้าเว็บไซต์ในช่วงวันที่ 29/1/2009 ว่ามีการแสดงผลเป็นอย่างไร จึงทำการเปลี่ยนวันที่ของเซิร์ฟเวอร์เป็นวันที่ 29/1/2009 พบว่ามีการเปลี่ยนแปลงหน้าแรกของเว็บไซต์ได้จริง ตามรูปที่ 5.16 และเมื่อเข้าไปดูสถานะของการแสดงผลพบว่าการแสดงผลของกลุ่มข้อมูลเปลี่ยนไปเป็นกลุ่มของวันที่ 29/1/2009 (ตามรูปที่ 5.17) และมีรายละเอียดการกำหนดโปรโมชันและข่าวสารในการแสดงผล (รูปที่ 5.18) ตรงกับหน้าเว็บในรูปที่ 5.16 ได้ถูกต้อง

The screenshot shows a web management interface titled 'Homepage'. At the top, there is a breadcrumb trail: 'Home > Homepage Management > Search'. There are 'Create' and 'Delete' buttons on the right. Below this, there is a 'Publish Date' section with two input fields (one empty, one containing '29/01/2009') and 'Search' and 'Reset' buttons. The main area is titled 'Search Result' and shows '4 item(s)'. Below this is a table with columns 'Publish Date', 'Current', and 'Create Date'. The table contains four rows of data:

Publish Date	Current	Create Date
27/02/2009		2/26/2009 10:53:36 PM
01/02/2009		1/30/2009 10:19:55 PM
29/01/2009	Show Now	1/29/2009 9:57:34 PM
22/12/2008		10/13/2008 4:21:08 PM

รูปที่ 5.17 หน้าระบบแสดงสถานะของการแสดงผลเมื่อเวลาของเซิร์ฟเวอร์แสดงผลวันที่ 29/1/2009

The screenshot shows a web management interface titled 'Homepage Information'. It has several sections for configuration:

- Publish Date:** 29/01/2009
- Promotion Item:** Five sections labeled 'Promotion 1: ไม่พร้อม', 'Promotion 2: ไม่พร้อม', 'Promotion 3: ไม่พร้อม', 'Promotion 4: ไม่พร้อม', and 'Promotion 5: ไม่พร้อม'. Each section has a 'Select Promotion' button.
- News Item:** Two sections labeled 'News 1: ข่าว 1' and 'News 2: ข่าว 2'. Each section has a 'Select News' button.
- Flash Animation:** A section labeled 'Flash File: zins' with a file input field and a 'Browse...' button.
- Display (Thai):** A section labeled 'แสดงสินค้า (ไทย)' with 'Yes' and 'No' radio buttons.

รูปที่ 5.18 กลุ่มโปรโมชันที่ต้องการให้แสดงตั้งแต่วันที่ 1/2/2009

## บทที่ 6

### สรุปผลการวิจัย และข้อเสนอแนะ

ในปัจจุบันเว็บไซต์ถือเป็นสื่อในการเผยแพร่ข้อมูลที่มีประสิทธิภาพสูงสื่อหนึ่งและเป็นที่ยอมรับนำมาใช้ในการโฆษณา สื่อสาร หรือให้ข้อมูลต่าง ๆ แก่ผู้ที่สนใจได้อย่างกว้างขวาง โดยที่ในแต่ละเว็บไซต์ก็มีจุดประสงค์และความต้องการในการใช้งานที่แตกต่างกันออกไป อาทิ เว็บไซต์ให้ข้อมูลทั่วไป เว็บไซต์เพื่อการซื้อขายแบบ E-Commerce รวมไปถึงรูปแบบในการนำเสนอด้วยเช่นกัน ไม่ว่าจะเป็นการนำเสนอข้อความ รูปภาพ ภาพเคลื่อนไหว หรืออื่น ๆ จะทำให้เห็นได้ว่าเว็บไซต์นั้นมีประโยชน์และความหลากหลายในการใช้งานเป็นอย่างมาก ทำให้การบริหารจัดการเว็บไซต์นั้นเป็นเรื่องที่ต้องให้ความสำคัญอย่างสูง

ระบบการบริหารจัดการเว็บไซต์สำเร็จรูปในปัจจุบันนั้น รองรับการจัดการบริหารข้อมูลบนเว็บไซต์ได้ในขอบเขตที่จำกัด โดยต้องเลือกใช้ให้เหมาะสมกับลักษณะของงานที่ต้องการนำเสนอ แต่ยังคงความสามารถหลักไว้เช่นเดียวกันคือ การมีระบบจัดการข้อมูลเพื่อแก้ไขเพิ่มเติม ลบ หรือปรับแต่รูปแบบโครงสร้างของการจัดวางข้อมูลในเว็บไซต์ ทั้งนี้รูปแบบของเว็บไซต์ที่ใช้ระบบบริหารจัดการเว็บไซต์สำเร็จรูปนั้นมีข้อจำกัดเรื่องความสวยงามในการออกแบบและความแตกต่างกันในโครงสร้างของการจัดวางในหน้าเว็บเพจ เมื่อใช้ระบบการจัดการเดียวกัน แต่ถ้ามาใช้วิธีการบริหารจัดการแบบธรรมดา นั้น ต้องใช้ความสามารถของผู้ดูแลเป็นหลัก ในการปรับแต่งหรือสร้างหน้าเว็บเพจให้มีความสวยงาม แต่ทั้งนี้จะมีปัญหาในการจัดการข้อมูลตามอายุเสมอ โดยเฉพาะอย่างยิ่งเมื่อต้องมีการบริหารจัดการเว็บไซต์จำนวนมาก ซึ่งไม่ว่าจะเป็นการบริหารเว็บไซต์ในรูปแบบธรรมดาหรือระบบการบริหารจัดการเว็บไซต์สำเร็จรูปนั้น ยังมีข้อจำกัดในการบริหารจัดการเชิงเวลาซึ่งนับเป็นปัญหาสำคัญของผู้ดูแลเว็บไซต์ ทั้งในการวางแผนในการนำเสนอข่าวสารข้อมูลในวันเวลาที่กำหนด และการย้อนดูในช่วงเว็บเพจในช่วงเวลาที่ต้องการว่ามีผลแสดงผลเช่นไร

ในวิทยานิพนธ์ฉบับนี้ได้นำเสนอแนวทางการปรับปรุงการบริหารจัดการเว็บไซต์ เพื่อแก้ปัญหาในการจัดเก็บข้อมูลของเว็บไซต์ ที่ในปัจจุบันจัดเก็บฐานข้อมูลเฉพาะส่วนของข้อมูลที่อยู่ในเว็บไซต์แต่ไม่ได้จัดเก็บโครงสร้างของเว็บที่มี ทำให้ไม่สามารถนำหน้าเว็บในอดีตมาเปรียบเทียบกับปัจจุบันได้ รวมถึงการจัดการเว็บไซต์ในเชิงเวลาเพื่อการวางแผนในการกำหนดวันเผยแพร่ข้อมูลโดยอัตโนมัติ ดังนั้นในวิทยานิพนธ์นี้จึงเสนอการออกแบบเว็บเมตาเคต้าสำหรับบริหารจัดการเว็บไซต์ โดยการออกแบบโครงสร้างฐานข้อมูลให้รองรับเว็บเมตาเคต้าและโครงสร้างทางภาษา HTML เพื่อนำเสนอการบริหารเว็บไซต์ด้วยเว็บเมตาเคต้า และประยุกต์การใช้งานในการจัดการเชิงเวลาของเว็บไซต์ทั้งในการย้อนกลับเพื่อดูข้อมูลในอดีตหรือวางแผนเพื่อกำหนดเวลาแสดงผลในอนาคต

ซึ่งจากการวิจัยในวิทยานิพนธ์ฉบับนี้พบว่า การนำเวปเมตาเดต้ามาใช้ในการออกแบบระบบบริหารจัดการนั้น สามารถจัดเก็บข้อมูลของเวปไซต์ทั้งในรูปแบบของโครงสร้าง และข้อมูลที่แสดงผลในรูปแบบของฐานข้อมูลเชิงเวลาในรูปเวปเมตาเดต้าได้จริง ทำให้เกิดประโยชน์ในการบริหารจัดการเวปไซต์ ดังจะเห็นได้จากที่ได้นำมาทดลองประยุกต์ใช้จริงตามที่กล่าวมาแล้ว ในบทก่อนหน้า ทั้งนี้การทดลองได้แสดงวิธีแก้ปัญหาในเรื่องของความอิสระในการออกแบบหน้าเวปไซต์และรูปแบบในการบริหารแบบเชิงเวลาที่สามารถเตรียมข้อมูลที่ต้องการแสดงผลไว้ก่อนล่วงหน้า หรือการย้อนกลับไปดูหน้าเวปไซต์ในเวลาที่ต้องการว่าเวปไซต์มีการแสดงผลในรูปแบบใดได้อย่างเหมาะสม เพื่อแก้ไขปัญหาคาการจัดการเวปไซต์ในรูปแบบอื่น ๆ ได้เป็นอย่างดี เป็นไปตามวัตถุประสงค์ในการศึกษาในงานวิจัยชิ้นนี้

อย่างไรก็ตามงานวิจัยชิ้นนี้ถือเป็นการเริ่มต้นในการนำเสนอการจัดการเวปไซต์จากการออกแบบฐานข้อมูลเชิงเวลาในรูปเวปเมตาเดต้า มาใช้ในการบริหารจัดการเวปไซต์ ทางผู้วิจัยเห็นว่าน่าจะเป็นอีกทางเลือกหนึ่งที่น่าสนใจ และสามารถที่จะนำไปปรับปรุงและพัฒนาเพิ่มเติมสู่ระบบการจัดการบริหารเวปไซต์ให้ดียิ่งขึ้นต่อไปได้ในอนาคต

## บรรณานุกรม

- [1] Yong-gang CAO, Nu0 LI, Mao-zhong JIN and Yu-qing LAN, **A Lightweight Multimedia Web Content Management System**, IEEE Internet Computing, 2004, pp. 85-90.
- [2] W3C. **HTML 4.01 Specification**. W3C Official Web Site; 2006.  
<http://www.w3.org/TR/1999/REC-html401-19991224>.
- [3] Bundit Pasaya and Suphamit Chittayasothorn, **A Temporal Object Oriented Conceptual Schema Model**, IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987.
- [4] K Putsadee Poraphol, **The Design and Development of a temporal database using TNIAM**, KMTTL ,Science, 2006.
- [5] Google Thailand.**Source code of www.google.co.th/index.html**. Google Thailand Official Web Site; 2006. <http://www.google.co.th>.
- [6] David Bodoff, Mordechai Ben-Menachem, and Patrick C.K. Hung, **Web Metadata Standards Observation and Prescriptions**, IEEE Software, January 2005, pp. 78-85.
- [7] Mihaela Sokic, Viktor Matic and Alen Bazant, **Web content management system based on XML native database**, 25th Int. Conf. Information Technology Interfaces ITI 2003, 16-19 June 2003, Cavtat, Croatia , pp. 457-462.
- [8] XML:DB Initiative. **XML:DB Initiative for XML Databases**. XML:DB Initiative Official Web Site; 2003.<http://www.xmldb.org>.
- [9] W3C. **W3C Technical Reports and Publications**. W3C Official Web Site; 2003.  
<http://www.w3.org/TR/>
- [10] CMS Thailand. **Content Management System (CMS)**, CMS Thailand Official Web Site. 2006. <http://www.cmsthailand.com/>.
- [11] osCommerce. **osCommerce system**, osCommerce Official Web Site. 2006.  
<http://www.oscommerce.com/>.
- [12] MAMBO. **MAMBO system**, MAMBO Official Web Site. 2006.  
<http://www.mamboserver.com/>.
- [13] XOOPS. **XOOPS System**, XOOPS Official Web Site. 2006. <http://www.xoops.org/>.
- [14] DOTNETNUKE. **DOTNETNUKE System**, DOTNETNUKE Official Web Site. 2006.  
<http://www.dotnetnuke.com/>.
- [15] อัครวุฒิ ตำราเรียง, **สร้างเว็บแบบมีออาชีพด้วย MAMBO Open Source Content Management System (CMS)**, SIPA, ISBN 974-93871-7-1 พิมพ์ครั้งที่ 1, มกราคม 2549.

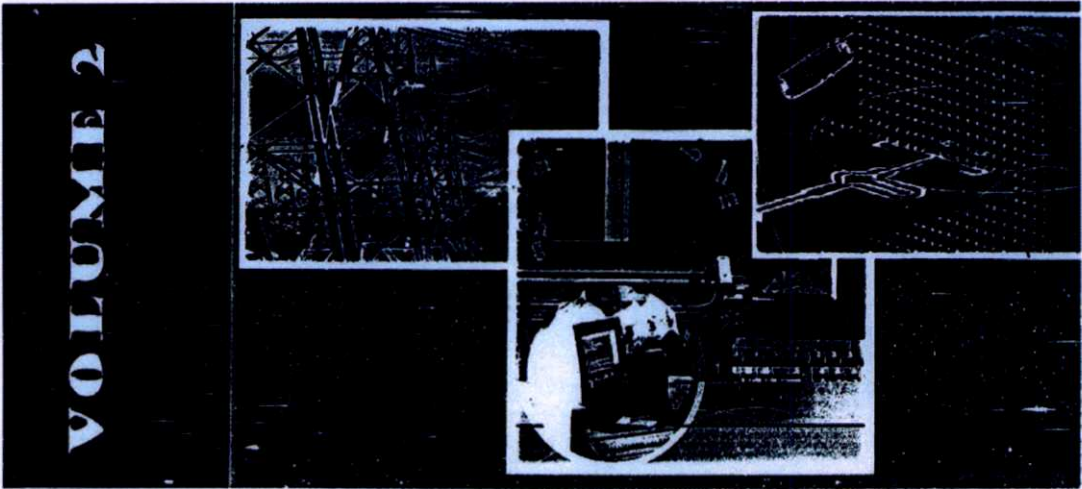
- [16] ประกิจ นัยรัตน์หิรัญ และพิชัย ยอดพฤติกการ, สุดยอดเครื่องมือสร้างเว็บ XOOOPS, ITbest, ISBN 974-94614-3-6 พิมพ์ครั้งที่ 1, กันยายน 2549.
- [17] ประชา พฤษย์ประเสริฐ, สร้างเว็บและเพิ่มลูกเล่นด้วย HTML&XHTML, บริษัท ชัคเชส มีเดีย จำกัด, ISBN 978-974-489-603-2 พิมพ์ครั้งที่ 1, มีนาคม 2550.

**ภาคผนวก**

### ภาคผนวก ก.

#### ผลงานวิจัยที่ได้รับการตีพิมพ์เผยแพร่

1. Natthaphon Sueb-Am and Mayuree Lertwatechakul, “**Web Metadata Design for Web Content Management**”, ECTI-CON 2007, pp. 1213-1216, Mae Fah Luang University, Chiang Rai, Thailand, May 9-12, 2007.

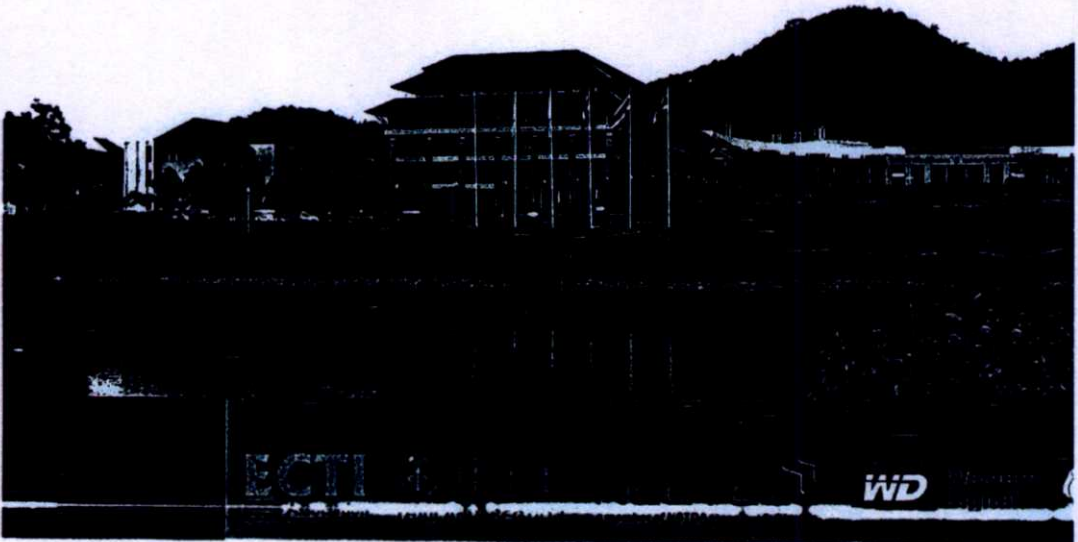


# **ECTI-CON 2007**

***Mae Fah Luang University, Chiang Rai, Thailand  
May 9-12, 2007***

## ***VOLUME 2***

- Communication Systems***
- Signal Processing***
- Computer and Information***



# Web Metadata Design for Web Content Management

Natthaphon Sueb-Am, Mayuree Lertwatechakul

Department of Information Engineering

Faculty of Engineering, King Mongkut's Institute of Technology, Ladkrabang

Bangkok, Thailand 10520

Email: s7061122@kmitl.ac.th , klmayure@kmitl.ac.th

**Abstract**— Nowadays, there are many tools have been offered to webmasters to build their websites. Therefore, Web administrators still have to cope with many problems, such as ensuring validity of available documents or changing many content and restoring data. To resolve this problem, we need Web Metadata which could be used by Web Content Management (WCM) to maintenance web structure and its content and representing schedule. In this paper, we propose a temporal database design for Web Metadata by using NIAM model, which can be used for solving problems in monitoring, designing and retrieval efficiency of the website structure and its contents.

**Keywords.** Web Metadata ,Web Content Management, Temporal web database, NIAM.

## 1. INTRODUCTION

Since the invention of websites, this technology has affected to most people's daily life. At the beginning, it was used by government and large business sectors only. However, the popularity of this technology spreads to small business and also private sectors.

There are varieties of contents, for instance: texts, images, audio clips, videos etc. Moreover, these contents have some changes regularly. The best example is portal websites, which have varieties of information, e.g. News, petrol rates, stock exchange rates, movie and music hits charts, etc. As can be seen from examples above, website management is not easy. There are so many information and data that needed to be monitored and updated very frequently. Thus, webmasters need powerful tools that can help them to manage and organize their works.

A website can be conceptually divided in the following four basic parts, like in [1]

- Content
- Structure
- Functionality
- Presentation

According to our studies, we founded that there are 2 parts of website components that are very important:

- Content
- Structure

Firstly, content is basic information that can be seen and heard from the website. In other words, all visual and audio information appeared on the website. Secondly, structure is the directions of how information is represented on the websites. Basically, the most common standard of structure is written by using HTML specifications [2]. As we have studied the HTML structure carefully, we can design the relationship of web metadata. Moreover, daily website contents and structures can also be tracked and traced. As a result, we chose NIAM model to represent database structure, which could handle temporal data for web content management.

This paper will focus on relationship of web metadata, especially in design, management and maintenance. To gain more understanding towards this research, it is needed to understand basic concepts, overview and implementation methods. The guide below described our methodology we applied on this research.

## 2. OVERVIEW AND RELATE WORK.

### 2.1 HTML Structure

HTML is a popular and powerful of computer language devised to allow website creation. These websites can then be viewed by anyone else connected to the Internet. It is constantly undergoing revision and evolution to meet the demands and requirements of the growing Internet audience under the direction of the World Wide Web Consortium (W3C): the organization charged with designing and maintaining the language.

New HTML version supports more multimedia options, scripting languages, style sheets, better printing facilities, and documents that are more accessible to users with disabilities. HTML also takes great strides towards the internationalization of documents, with the goal of making the Web truly World Wide.

HTML structure consists of elements and attributes. Elements are a part of HTML structure. They include 2 important properties, which are attributes and content. To make HTML document, these properties are used for parameter defining and follow HTML structure. An element usually has a start tag (<tag>) and an end tag (</tag>), such as <Font color="black">Content</Font>.

Attribute are used to define Element's parameter and contain 2 parts. First part is attribute name for define property's element. Another part is attribute value. Attributes are placed in the start tag (<TD width=300>) and content is located between the tags (<tag>Content</tag>). And some elements don't have any content and do not need closing tags, such as <HR>.

We can study all details of HTML structure in HTML specifications in WC3's website for this paper.

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD HTML
4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
<TITLE>My first HTML document</TITLE>
</HEAD>
<BODY>
<P>Hello world!
</BODY>
</HTML>
```

Figure 1. Example source code for basic web structure.

### 2.2. Concept of Temporal database by NIAM

In temporal relational databases [3], it consists of several types of time include this.

- Lifespan time
- Valid time
- Transaction time

Lifespan Time, Valid Time and Transaction Time are vital factors of time components in temporal databases.

Lifespan time is the period of time between the Entity type starts until its end. Any Entity type that is measured by time, will be collected in database. The components of Lifespan Time are Life Span Start (LSs), and Life Span End (LSe). The symbol for represent in NIAM model is shown below in Figure 2(a)

Valid time is the period of time, it consists of Valid Time start (VTs) and Valid Time end (VTe) are attached to each temporal fact so that the state of the fact is shown with the valid time. This enables historical facts to be manipulated as well as current facts. The symbol for represent in NIAM model is shown below in Figure 2(b)

Transaction time is simply the time that the fact is recorded into the database, which consists of Transaction Time Start (TTs) and Transaction Time End (TTe).

In temporal relational databases, lifespan, valid and transaction time are attached to each tuple.

## 3. IMPLEMENTATION

After carefully analyzed HTML Structure, it is very flexible and may hand to design a concrete database model for. We tried to make the most appropriate data model to handle all possible relations of HTML structure and content, e.g. retrieval speed, indexes and data constraints. Relational structure that would satisfy our common structure is shown in Figure. 3. After that we will Transformation from NIAM to Object Database Schema and mapping in tables for implement in web content management system. We will develop a tool to transform HTML code of webpages into tuples of these tables.



Figure 2(a). Symbol for Life span time [4].

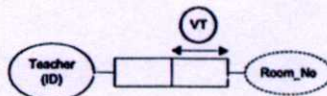


Figure 2(b). Symbol for Valid time [4].

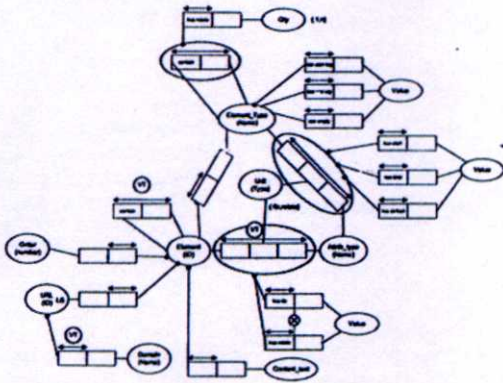


Figure 3. A NIAM conceptual schema of the Web metadata database.

From NIAM conceptual schema in Fig 3 we can map to relational schema as shown below.

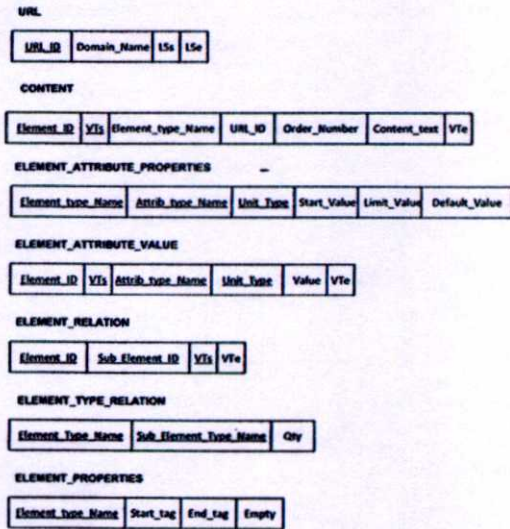


Figure 4. Web metadata relational schemas.

**URL TABLE**

URL_ID	Domain_Name	L5a	L5b
1	www.google.co.uk	2000-05-15	0000-12-31

**CONTENT TABLE**

ELEMENT_ID	URL_ID	ELEMENT_TYPE_NAME	ORDER_NUMBER	CONTENT_TEXT	Y1s	Y7s
1	1	HTML	1		2000-05-15	0000-12-31
2	1	HEAD	2		2000-05-15	0000-12-31
3	1	META	3		2000-05-15	0000-12-31
4	1	TITLE	4	Google	2000-05-15	0000-12-31
...	...	...	...	...	...	...
9	1	GVF	9		2000-05-15	0000-12-31

**ELEMENT\_ATTRIBUTE\_VALUE TABLE**

ELEMENT_ID	ATTRIB_TYPE_NAME	UNIT_TYPE	VALUE	Y1s	Y7s
2	content	F	background-color: #FFFFFF; border: 1px solid black; padding: 5px; width: 100%; height: 100%; margin: 0; text-align: center;">current type	2000-05-15	0000-12-31
2	http-equiv	F	current type	2000-05-15	0000-12-31
7	language	F	enGB	2000-05-15	0000-12-31
...	...	...	...	...	...
9	0	F	background-color: #FFFFFF; border: 1px solid black; padding: 5px; width: 100%; height: 100%; margin: 0; text-align: center;">current type	2000-05-15	0000-12-31
9	0	F	background-color: #FFFFFF; border: 1px solid black; padding: 5px; width: 100%; height: 100%; margin: 0; text-align: center;">current type	2000-05-15	0000-12-31
9	0	F	background-color: #FFFFFF; border: 1px solid black; padding: 5px; width: 100%; height: 100%; margin: 0; text-align: center;">current type	2007-01-01	9999-12-31
9	0	F	background-color: #FFFFFF; border: 1px solid black; padding: 5px; width: 100%; height: 100%; margin: 0; text-align: center;">current type	2007-01-01	9999-12-31

**ELEMENT\_RELATION TABLE**

ELEMENT_ID	SUB_ELEMENT_ID	Y1s	Y7s
9	9	2000-05-15	9999-12-31
9	10	2000-05-15	9999-12-31
11	12	2000-05-15	9999-12-31

**ELEMENT\_TYPE\_RELATION TABLE**

ELEMENT_TYPE_NAME	SUB_ELEMENT_TYPE_NAME	QTY
ADDRESS	BODY	1
APPLET	BODY	1
BODY	ADDRESS	N

**ELEMENT\_ATTRIBUTE\_PROPERTIES TABLE**

ELEMENT_TYPE_NAME	ATTRIB_TYPE_NAME	UNIT_TYPE	START_VALUE	LIMIT_VALUE	DEFAULT_VALUE
BODY	language	F	enGB	enGB	enGB
TABLE	high	R	1%	100%	100%
TABLE	width	R	1%	100%	100%

**ELEMENT\_RELATION TABLE**

ELEMENT_TYPE_NAME	START_TAG	END_TAG	EMPTY
BODY	O	O	N
AREA	V	F	E
TABLE	V	V	N

Figure 5. A part of sample database for storing content structures.

We have proved that the designed schema could store various arbitrary styles of HTML structure by analyzing many webpages of popular websites such as index.html of www.google.co.th [5], www.cnn.com, www.msn.com. We found that all analyzed structures, can be filled in the database schema. As to show some of the success experiment, we present structure include page of www.google.co.th as relational data in the Figure 5.

In Figure 5 in ELEMENT\_ATTRIBUTE\_VALUE Table and other tables show example records set from www.google.co.th in different time. As can be seen from set ①, it is represented the google webpage at 15/6/2006, by using Google's logo, which collected as logo\_pain.png (in Figure 6(a)). In set ②, on Mother Day: 12/6/2006, the logo of the webpage has changed to mothers\_day06.png (in Figure 6(b)). According to record ③ (in Figure 6(c)), on 1/1/2007 (New Year 2007), the logo has changed for celebration of New Year. The logo file became newyear07.png. After the new year's celebrations ended, the logo has changed to default logo. The file became "logo\_pain.png" as it was on 20/1/2007 (in Figure 6(a)).

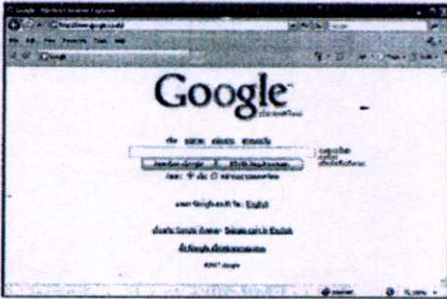


Figure 6. (a). Webpage of www.google.co.th in 15-06-2006 and current date.

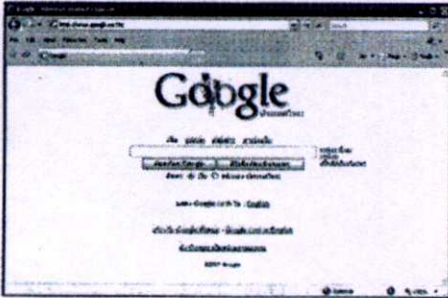


Figure 6(b). Retrieved Webpage of www.google.co.th in 12-08-2006.

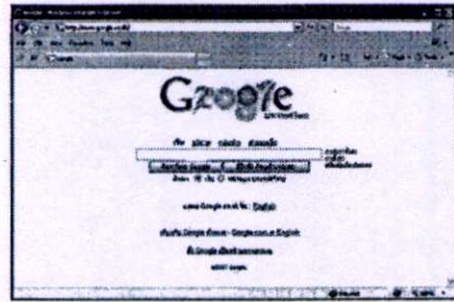


Figure 6(c). Retrieved Webpage of www.google.co.th in 1-01-2007.

#### 4. CONCLUSIONS AND FURTHER WORK.

In this paper, we have presented the Web Metadata model that useful for Web Content Management. By adding temporal feature, In the near future we will develop Web Content Management System (WCMS). The database model can maintain HTML Structure, Content and retrieval features. These may be changed time by time.

#### REFERENCES

- [1] Mihaela Sokic , Viktor Matic and Alen Bazant , "Web Content Management System Based on XML Native Database" 25<sup>th</sup> Int.Conf IEEE Internet Computing , July - August 1998 , pp. 30-37.
- [2] HTML 4.01 Specification, <http://www.w3.org/TR/1999/REC-html401-19991224>.
- [3] Bundit Pasaya and Suphamit Chittayasothorn, "A Temporal Object Oriented Conceptual Schema Model," IEEE Trans. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9<sup>th</sup> Annual Conf. Magnetics Japan, p. 301, 2001, pp 724 -727.
- [4] K Putsadee Poraphol, "The Design and Development of a temporal database using TNLAM", KMITL Science, 2006 .
- [5] "Source code of www.google.co.th/index.html", www.google.co.th.

**ภาคผนวก ข.**

HTML 4.01 Specification

**มาตรฐาน HTML**



## HTML 4.01 Specification

### W3C Recommendation 24 December 1999

This version:

<http://www.w3.org/TR/1999/REC-html401-19991224>  
 (plain text [794Kb], gzip'ed tar archive of HTML files [371Kb], a zip archive of HTML files [405Kb], gzip'ed Postscript file [746Kb, 389 pages], gzip'ed PDF file [963Kb])

Latest version of HTML 4.01:

<http://www.w3.org/TR/html401>

Latest version of HTML 4:

<http://www.w3.org/TR/html4>

Latest version of HTML:

<http://www.w3.org/TR/html>

Previous version of HTML 4.01:

<http://www.w3.org/TR/1999/PR-html40-19990824>

Previous HTML 4 Recommendation:

<http://www.w3.org/TR/1998/REC-html40-19980424>

Editors:

Dave Raggett <[dsr@w3.org](mailto:dsr@w3.org)>

Arnaud Le Hors, W3C

Ian Jacobs, W3C

Copyright ©1997-1999 W3C® (MIT, INRIA, Keio), All Rights Reserved. W3C liability, trademark, document use and software licensing rules apply.

---

### Abstract

This specification defines the HyperText Markup Language (HTML), the publishing language of the World Wide Web. This specification defines HTML 4.01, which is a subversion of HTML 4. In addition to the text, multimedia, and hyperlink features of the previous versions of HTML (HTML 3.2 [HTML32] [p.356] and HTML 2.0 [RFC1866] [p.356]), HTML 4 supports more multimedia options, scripting languages, style sheets, better printing facilities, and documents that are more accessible to users with disabilities. HTML 4 also takes great strides towards the internationalization of documents, with the goal of making the Web truly World Wide.

HTML 4 is an SGML application conforming to International Standard ISO 8879 – Standard Generalized Markup Language [ISO8879] [p.353].

## Status of this document

*This section describes the status of this document at the time of its publication. Other documents may supersede this document. The latest status of this document series is maintained at the W3C.*

This document specifies HTML 4.01, which is part of the HTML 4 line of specifications. The first version of HTML 4 was HTML 4.0 [HTML40] [p.353], published on 18 December 1997 and revised 24 April 1998. This specification is the first HTML 4.01 Recommendation. It includes non-editorial changes since the 24 April version of HTML 4.0 [p.312]. There have been some changes to the DTDs, for example. This document obsoletes previous versions of HTML 4.0, although W3C will continue to make those specifications and their DTDs available at the W3C Web site.

This document has been reviewed by W3C Members and other interested parties and has been endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited as a normative reference from another document. W3C's role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment. This enhances the functionality and interoperability of the Web.

W3C recommends that user agents and authors (and in particular, authoring tools) produce HTML 4.01 documents rather than HTML 4.0 documents. W3C recommends that authors produce HTML 4 documents instead of HTML 3.2 documents. For reasons of backward compatibility, W3C also recommends that tools interpreting HTML 4 continue to support HTML 3.2 and HTML 2.0 as well.

For information about the next generation of HTML, "The Extensible HyperText Markup Language" [XHTML] [p.357], please refer to the W3C HTML Activity and the list of W3C Technical Reports.

This document has been produced as part of the W3C HTML Activity. The goals of the HTML Working Group (*Members only*) are discussed in the HTML Working Group charter (*Members only*).

A list of current W3C Recommendations and other technical documents can be found at <http://www.w3.org/TR>.

Public discussion on HTML features takes place on [www-html@w3.org](mailto:www-html@w3.org) (archives of [www-html@w3.org](mailto:www-html@w3.org)).

## Available languages

The English version of this specification is the only normative version. However, for translations of this document, see <http://www.w3.org/MarkUp/html4-updates/translations>.

HTML 4.01 Specification

## Errata

The list of known errors in this specification is available at:  
<http://www.w3.org/MarkUp/html4-updates/errata>

Please report errors in this document to [www-html-editor@w3.org](mailto:www-html-editor@w3.org).

## Index of the HTML 4 Elements

## Index of Elements

Legend: *Optional, Forbidden, Empty, Deprecated, Loose DTD, Frameset DTD*

Name	Start Tag	End Tag	Empty	Depr.	DTD	Description
A [p.149]						anchor
ABBR [p.90]						abbreviated form (e.g., WWW, HTTP, etc.)
ACRONYM [p.90]						
ADDRESS [p.76]						information on author
APPLET [p.171]				D	L	Java applet
AREA [p.174]		F	E			client-side image map area
B [p.199]						bold text style
BASE [p.157]		F	E			document base URI
BASEFONT [p.200]		F	E	D	L	base font size
BDO [p.85]						l18N BiDi over-ride
BIG [p.199]						large text style
BLOCKQUOTE [p.92]						long quotation
BODY [p.69]	O	O				document body
BR [p.96]		F	E			forced line break
BUTTON [p.228]						push button
CAPTION [p.115]						table caption
CENTER [p.197]				D	L	shorthand for DIV align=center
CITE [p.90]						citation
CODE [p.90]						computer code fragment

## Index of the HTML 4 Elements

COL [p.120]		F	E			table column
COLGROUP [p.118]		O				table column group
DD [p.106]		O				definition description
DEL [p.99]						deleted text
DFN [p.90]						instance definition
DIR [p.109]				D	L	directory list
DIV [p.73]						generic language/style container
DL [p.106]						definition list
DT [p.106]		O				definition term
EM [p.90]						emphasis
FIELDSET [p.239]						form control group
FONT [p.200]				D	L	local change to font
FORM [p.222]						interactive form
FRAME [p.209]		F	E		F	subwindow
FRAMESET [p.206]					F	window subdivision
H1 [p.75]						heading
H2 [p.75]						heading
H3 [p.75]						heading
H4 [p.75]						heading
H5 [p.75]						heading
H6 [p.75]						heading
HEAD [p.62]	O	O				document head
HR [p.202]		F	E			horizontal rule
HTML [p.61]	O	O				document root element
I [p.199]						italic text style
IFRAME [p.217]					L	inline subwindow

## Index of the HTML 4 Elements

IMG [p.160]		F	E			Embedded image
INPUT [p.224]		F	E			form control
INS [p.99]						inserted text
ISINDEX [p.236]		F	E	D	L	single line prompt
KBD [p.90]						text to be entered by the user
LABEL [p.237]						form field label text
LEGEND [p.239]						fieldset legend
LI [p.105]		O				list item
LINK [p.154]		F	E			a media-independent link
MAP [p.174]						client-side image map
MENU [p.109]				D	L	menu list
META [p.65]		F	E			generic metainformation
NOFRAMES [p.214]					F	alternate content container for non frame-based rendering
NOSCRIPT [p.258]						alternate content container for non script-based rendering
OBJECT [p.162]						generic embedded object
OL [p.104]						ordered list
OPTGROUP [p.230]						option group
OPTION [p.230]		O				selectable choice
P [p.95]		O				paragraph
PARAM [p.167]		F	E			named property value
PRE [p.97]						preformatted text
Q [p.92]						short inline quotation
S [p.199]				D	L	strike-through text style
SAMP [p.90]						sample program output, scripts, etc.

## Index of the HTML 4 Elements

SCRIPT [p.252]						script statements
SELECT [p.230]						option selector
SMALL [p.199]						small text style
SPAN [p.73]						generic language/style container
STRIKE [p.199]				D	L	strike-through text
STRONG [p.90]						strong emphasis
STYLE [p.187]						style info
SUB [p.94]						subscript
SUP [p.94]						superscript
TABLE [p.113]						
TBODY [p.116]	O	O				table body
TD [p.125]		O				table data cell
TEXTAREA [p.234]						multi-line text field
TFOOT [p.116]		O				table footer
TH [p.125]		O				table header cell
THEAD [p.116]		O				table header
TITLE [p.62]						document title
TR [p.124]		O				table row
TT [p.199]						teletype or monospaced text style
U [p.199]				D	L	underlined text style
UL [p.104]						unordered list
VAR [p.90]						instance of a variable or program argument

## Index of Attributes

*Legend: Deprecated, Loose DTD, Frameset DTD*

Name	Related Elements	Type	Default	Depr.	DTD	Comment
abbr [p.126]	TD, TH	%Text; [p.266]	#IMPLIED			abbreviation for header cell
accept-charset [p.223]	FORM	%Charsets; [p.266]	#IMPLIED			list of supported charsets
accept [p.223]	FORM, INPUT	%ContentTypes; [p.266]	#IMPLIED			list of MIME types for file upload
accesskey [p.242]	A, AREA, BUTTON, INPUT, LABEL, LEGEND, TEXTAREA	%Character; [p.266]	#IMPLIED			accessibility key character
action [p.223]	FORM	%URI; [p.266]	#REQUIRED			server-side form handler
align [p.115]	CAPTION	%CAalign; [p.292]	#IMPLIED	D	L	relative to table
align [p.180]	APPLET, IFRAME, IMG, INPUT, OBJECT	%AAlign; [p.285]	#IMPLIED	D	L	vertical or horizontal alignment
align [p.239]	LEGEND	%LAalign; [p.290]	#IMPLIED	D	L	relative to fieldset
align [p.113]	TABLE	%TAAlign; [p.291]	#IMPLIED	D	L	table position relative to window
align [p.202]	HR	(left   center   right)	#IMPLIED	D	L	
align [p.196]	DIV, H1, H2, H3, H4, H5, H6, P	(left   center   right   justify)	#IMPLIED	D	L	align, text alignment
align [p.132]	COL, COLGROUP, TBODY, TD, TFOOT, TH, THEAD, TR	(left   center   right   justify   char)	#IMPLIED			
alink [p.69]	BODY	%Color; [p.281]	#IMPLIED	D	L	color of selected links
alt [p.181]	APPLET	%Text; [p.281]	#IMPLIED	D	L	short description
alt [p.181]	AREA, IMG	%Text; [p.266]	#REQUIRED			short description
alt [p.181]	INPUT	CDATA [p.50]	#IMPLIED			short description
archive [p.171]	APPLET	CDATA [p.50]	#IMPLIED	D	L	comma-separated archive list
archive [p.163]	OBJECT	CDATA [p.50]	#IMPLIED			space-separated list of URIs

## Index of the HTML 4 Attributes

axis [p.126]	TD, TH	CDATA [p.50]	#IMPLIED			comma-separated list of related headers
background [p.69]	BODY	%URI; [p.280]	#IMPLIED	D	L	texture tile for document background
bgcolor [p.195]	TABLE	%Color; [p.281]	#IMPLIED	D	L	background color for cells
bgcolor [p.195]	TR	%Color; [p.281]	#IMPLIED	D	L	background color for row
bgcolor [p.195]	TD, TH	%Color; [p.281]	#IMPLIED	D	L	cell background color
bgcolor [p.195]	BODY	%Color; [p.281]	#IMPLIED	D	L	document background color
border [p.130]	TABLE	%Pixels; [p.270]	#IMPLIED			controls frame width around table
border [p.180]	IMG, OBJECT	%Pixels; [p.285]	#IMPLIED	D	L	link border width
cellpadding [p.135]	TABLE	%Length; [p.270]	#IMPLIED			spacing within cells
cellspacing [p.134]	TABLE	%Length; [p.270]	#IMPLIED			spacing between cells
char [p.132]	COL, COLGROUP, TBODY, TD, TFOOT, TH, THEAD, TR	%Character; [p.266]	#IMPLIED			alignment char, e.g. char="'
charoff [p.133]	COL, COLGROUP, TBODY, TD, TFOOT, TH, THEAD, TR	%Length; [p.270]	#IMPLIED			offset for alignment char
charset [p.150]	A, LINK, SCRIPT	%Charset; [p.266]	#IMPLIED			char encoding of linked resource
checked [p.225]	INPUT	(checked)	#IMPLIED			for radio buttons and check boxes
cite [p.93]	BLOCKQUOTE, Q	%URI; [p.266]	#IMPLIED			URI for source document or msg
cite [p.100]	DEL, INS	%URI; [p.266]	#IMPLIED			info on reason for change
class [p.71]	All elements but BASE, BASEFONT, HEAD, HTML, META, PARAM, SCRIPT, STYLE, TITLE	CDATA [p.50]	#IMPLIED			space-separated list of classes

## Index of the HTML 4 Attributes

classid [p.163]	OBJECT	%URI; [p.266]	#IMPLIED			identifies an implementation
clear [p.198]	BR	(left   all   right   none)	none	D	L	control of text flow
code [p.171]	APPLET	CDATA [p.50]	#IMPLIED	D	L	applet class file
codebase [p.163]	OBJECT	%URI; [p.266]	#IMPLIED			base URI for classid, data, archive
codebase [p.171]	APPLET	%URI; [p.280]	#IMPLIED	D	L	optional base URI for applet
codetype [p.163]	OBJECT	%ContentType; [p.266]	#IMPLIED			content type for code
color [p.201]	BASEFONT, FONT	%Color; [p.281]	#IMPLIED	D	L	text color
cols [p.207]	FRAMESET	%MultiLengths; [p.270]	#IMPLIED		F	list of lengths, default: 100% (1 col)
cols [p.235]	TEXTAREA	NUMBER [p.50]	#REQUIRED			
colspan [p.126]	TD, TH	NUMBER [p.50]	1			number of cols spanned by cell
compact [p.105]	DIR, DL, MENU, OL, UL	(compact)	#IMPLIED	D	L	reduced interitem spacing
content [p.65]	META	CDATA [p.50]	#REQUIRED			associated information
coords [p.175]	AREA	%Coords; [p.269]	#IMPLIED			comma-separated list of lengths
coords [p.175]	A	%Coords; [p.269]	#IMPLIED			for use with client-side image maps
data [p.163]	OBJECT	%URI; [p.266]	#IMPLIED			reference to object's data
datetime [p.100]	DEL, INS	%Datetime; [p.266]	#IMPLIED			date and time of change
declare [p.164]	OBJECT	(declare)	#IMPLIED			declare but don't instantiate flag
defer [p.252]	SCRIPT	(defer)	#IMPLIED			UA may defer execution of script
dir [p.82]	All elements but APPLET, BASE, BASEFONT, BDO, BR, FRAME, FRAMESET, IFRAME, PARAM, SCRIPT	(ltr   rtl)	#IMPLIED			direction for weak/neutral text

## Index of the HTML 4 Attributes

dir [p.85]	BDO	(ltr   rtl)	#REQUIRED			directionality
disabled [p.244]	BUTTON, INPUT, OPTGROUP, OPTION, SELECT, TEXTAREA	(disabled)	#IMPLIED			unavailable in this context
enctype [p.223]	FORM	%ContentType; [p.266]	"application/x-www-form-urlencoded"			
face [p.201]	BASEFONT, FONT	CDATA [p.50]	#IMPLIED	D	L	comma-separated list of font names
for [p.237]	LABEL	IDREF [p.50]	#IMPLIED			matches field ID value
frame [p.130]	TABLE	%TFrame; [p.275]	#IMPLIED			which parts of frame to render
frameborder [p.210]	FRAME, IFRAME	(1   0)	1		F	request frame borders?
headers [p.125]	TD, TH	IDREFS [p.50]	#IMPLIED			list of id's for header cells
height [p.217]	IFRAME	%Length; [p.285]	#IMPLIED		L	frame height
height [p.126]	TD, TH	%Length; [p.285]	#IMPLIED	D	L	height for cell
height [p.179]	IMG, OBJECT	%Length; [p.270]	#IMPLIED			override height
height [p.172]	APPLET	%Length; [p.285]	#REQUIRED	D	L	initial height
href [p.149]	A, AREA, LINK	%URI; [p.266]	#IMPLIED			URI for linked resource
href [p.157]	BASE	%URI; [p.266]	#IMPLIED			URI that acts as base URI
hreflang [p.149]	A, LINK	%LanguageCode; [p.266]	#IMPLIED			language code
hspace [p.180]	APPLET, IMG, OBJECT	%Pxels; [p.285]	#IMPLIED	D	L	horizontal gutter
http-equiv [p.85]	META	NAME [p.50]	#IMPLIED			HTTP response header name
id [p.71]	All elements but BASE, HEAD, HTML, META, SCRIPT, STYLE, TITLE	ID [p.50]	#IMPLIED			document-wide unique id
ismap [p.179]	IMG, INPUT	(ismap)	#IMPLIED			use server-side image map
label [p.232]	OPTION	%Text; [p.266]	#IMPLIED			for use in hierarchical menus
label [p.232]	OPTGROUP	%Text; [p.266]	#REQUIRED			for use in hierarchical menus

## Index of the HTML 4 Attributes

lang [p.79]	All elements but APPLET, BASE, BASEFONT, BR, FRAME, FRAMESET, IFRAME, PARAM, SCRIPT	%LanguageCode; [p.266]	#IMPLIED			language code
language [p.252]	SCRIPT	CDATA [p.50]	#IMPLIED	D	L	predefined script language name
link [p.69]	BODY	%Color; [p.281]	#IMPLIED	D	L	color of links
longdesc [p.161]	IMG	%URI; [p.266]	#IMPLIED			link to long description (complements alt)
longdesc [p.209]	FRAME, IFRAME	%URI; [p.266]	#IMPLIED		F	link to long description (complements title)
marginheight [p.210]	FRAME, IFRAME	%Pixels; [p.270]	#IMPLIED		F	margin height in pixels
marginwidth [p.210]	FRAME, IFRAME	%Pixels; [p.270]	#IMPLIED		F	margin widths in pixels
maxlength [p.225]	INPUT	NUMBER [p.50]	#IMPLIED			max chars for text fields
media [p.187]	STYLE	%MediaDesc; [p.266]	#IMPLIED			designed for use with these media
media [p.187]	LINK	%MediaDesc; [p.266]	#IMPLIED			for rendering on these media
method [p.223]	FORM	(GET   POST)	GET			HTTP method used to submit the form
multiple [p.230]	SELECT	(multiple)	#IMPLIED			default is single selection
name [p.229]	BUTTON, TEXTAREA	CDATA [p.50]	#IMPLIED			
name [p.171]	APPLET	CDATA [p.50]	#IMPLIED	D	L	allows applets to find each other
name [p.230]	SELECT	CDATA [p.50]	#IMPLIED			field name
name [p.223]	FORM	CDATA [p.50]	#IMPLIED			name of form for scripting
name [p.209]	FRAME, IFRAME	CDATA [p.50]	#IMPLIED		F	name of frame for targetting
name [p.161]	IMG	CDATA [p.50]	#IMPLIED			name of image for scripting
name [p.149]	A	CDATA [p.50]	#IMPLIED			named link end

## Index of the HTML 4 Attributes

name [p.225]	INPUT, OBJECT	CDATA [p.50]	#IMPLIED			submit as part of form
name [p.174]	MAP	CDATA [p.50]	#REQUIRED			for reference by usemap
name [p.167]	PARAM	CDATA [p.50]	#REQUIRED			property name
name [p.65]	META	NAME [p.50]	#IMPLIED			metainformation name
nohref [p.175]	AREA	(nohref)	#IMPLIED			this region has no action
noresize [p.210]	FRAME	(noresize)	#IMPLIED		F	allow users to resize frames?
noshade [p.202]	HR	(noshade)	#IMPLIED	D	L	
nowrap [p.126]	TD, TH	(nowrap)	#IMPLIED	D	L	suppress word wrap
object [p.171]	APPLET	CDATA [p.50]	#IMPLIED	D	L	serialized applet file
onblur [p.255]	A, AREA, BUTTON, INPUT, LABEL, SELECT, TEXTAREA	%Script; [p.266]	#IMPLIED			the element lost the focus
onchange [p.256]	INPUT, SELECT, TEXTAREA	%Script; [p.266]	#IMPLIED			the element value was changed
onclick [p.255]	All elements but APPLET, BASE, BASEFONT, BDO, BR, FONT, FRAME, FRAMESET, HEAD, HTML, IFRAME, ISINDEX, META, PARAM, SCRIPT, STYLE, TITLE	%Script; [p.266]	#IMPLIED			a pointer button was clicked
ondblclick [p.255]	All elements but APPLET, BASE, BASEFONT, BDO, BR, FONT, FRAME, FRAMESET, HEAD, HTML, IFRAME, ISINDEX, META, PARAM, SCRIPT, STYLE, TITLE	%Script; [p.266]	#IMPLIED			a pointer button was double clicked

## Index of the HTML 4 Attributes

onfocus [p.255]	A, AREA, BUTTON, INPUT, LABEL, SELECT, TEXTAREA	%Script; [p.266]	#IMPLIED			the element got the focus
onkeydown [p.255]	All elements but APPLET, BASE, BASEFONT, BDO, BR, FONT, FRAME, FRAMESET, HEAD, HTML, IFRAME, ISINDEX, META, PARAM, SCRIPT, STYLE, TITLE	%Script; [p.266]	#IMPLIED			a key was pressed down
onkeypress [p.255]	All elements but APPLET, BASE, BASEFONT, BDO, BR, FONT, FRAME, FRAMESET, HEAD, HTML, IFRAME, ISINDEX, META, PARAM, SCRIPT, STYLE, TITLE	%Script; [p.266]	#IMPLIED			a key was pressed and released
onkeyup [p.255]	All elements but APPLET, BASE, BASEFONT, BDO, BR, FONT, FRAME, FRAMESET, HEAD, HTML, IFRAME, ISINDEX, META, PARAM, SCRIPT, STYLE, TITLE	%Script; [p.266]	#IMPLIED			a key was released
onload [p.254]	FRAMESET	%Script; [p.266]	#IMPLIED		F	all the frames have been loaded
onload [p.254]	BODY	%Script; [p.266]	#IMPLIED			the document has been loaded

## Index of the HTML 4 Attributes

onmousedown [p.255]	All elements but APPLET, BASE, BASEFONT, BDO, BR, FONT, FRAME, FRAMESET, HEAD, HTML, IFRAME, ISINDEX, META, PARAM, SCRIPT, STYLE, TITLE	%Script; [p.266]	#IMPLIED			a pointer button was pressed down
onmousemove [p.255]	All elements but APPLET, BASE, BASEFONT, BDO, BR, FONT, FRAME, FRAMESET, HEAD, HTML, IFRAME, ISINDEX, META, PARAM, SCRIPT, STYLE, TITLE	%Script; [p.266]	#IMPLIED			a pointer was moved within
onmouseout [p.255]	All elements but APPLET, BASE, BASEFONT, BDO, BR, FONT, FRAME, FRAMESET, HEAD, HTML, IFRAME, ISINDEX, META, PARAM, SCRIPT, STYLE, TITLE	%Script; [p.266]	#IMPLIED			a pointer was moved away
onmouseover [p.255]	All elements but APPLET, BASE, BASEFONT, BDO, BR, FONT, FRAME, FRAMESET, HEAD, HTML, IFRAME, ISINDEX, META, PARAM, SCRIPT, STYLE, TITLE	%Script; [p.266]	#IMPLIED			a pointer was moved onto

## Index of the HTML 4 Attributes

onmouseover [p.255]	All elements but APPLET, BASE, BASEFONT, BDO, BR, FONT, FRAME, FRAMESET, HEAD, HTML, IFRAME, ISINDEX, META, PARAM, SCRIPT, STYLE, TITLE	%Script; [p.266]	#IMPLIED			a pointer button was released
onreset [p.256]	FORM	%Script; [p.266]	#IMPLIED			the form was reset
onselect [p.256]	INPUT, TEXTAREA	%Script; [p.266]	#IMPLIED			some text was selected
onsubmit [p.255]	FORM	%Script; [p.266]	#IMPLIED			the form was submitted
onunload [p.255]	FRAMESET	%Script; [p.266]	#IMPLIED		F	all the frames have been removed
onunload [p.255]	BODY	%Script; [p.266]	#IMPLIED			the document has been removed
profile [p.62]	HEAD	%URI; [p.266]	#IMPLIED			named dictionary of meta info
prompt [p.236]	ISINDEX	%Text; [p.281]	#IMPLIED	D	L	prompt message
readonly [p.244]	TEXTAREA	(readonly)	#IMPLIED			
readonly [p.244]	INPUT	(readonly)	#IMPLIED			for text and passwd
rel [p.149]	A, LINK	%LinkTypes; [p.266]	#IMPLIED			forward link types
rev [p.150]	A, LINK	%LinkTypes; [p.266]	#IMPLIED			reverse link types
rows [p.207]	FRAMESET	%MultiLengths; [p.270]	#IMPLIED		F	list of lengths, default: 100% (1 row)
rows [p.235]	TEXTAREA	NUMBER [p.50]	#REQUIRED			
rowspan [p.126]	TD, TH	NUMBER [p.50]	1			number of rows spanned by cell
rules [p.130]	TABLE	%TRules; [p.275]	#IMPLIED			rulings between rows and cols
scheme [p.65]	META	CDATA [p.50]	#IMPLIED			select form of content
scope [p.125]	TD, TH	%Scope; [p.277]	#IMPLIED			scope covered by header cells

## Index of the HTML 4 Attributes

scrolling [p.210]	FRAME, IFRAME	(yes   no   auto)	auto		F	scrollbar or none
selected [p.232]	OPTION	(selected)	#IMPLIED			
shape [p.174]	AREA	%Shape; [p.269]	rect			controls interpretation of coords
shape [p.174]	A	%Shape; [p.269]	rect			for use with client-side image maps
size [p.202]	HR	%Pixels; [p.285]	#IMPLIED	D	L	
size [p.201]	FONT	CDATA [p.50]	#IMPLIED	D	L	[+]-]n e.g. size="+1", size="4"
size [p.225]	INPUT	CDATA [p.50]	#IMPLIED			specific to each type of field
size [p.201]	BASEFONT	CDATA [p.50]	#REQUIRED	D	L	base font size for FONT elements
size [p.230]	SELECT	NUMBER [p.50]	#IMPLIED			rows visible
span [p.121]	COL	NUMBER [p.50]	1			COL attributes affect N columns
span [p.119]	COLGROUP	NUMBER [p.50]	1			default number of columns in group
src [p.252]	SCRIPT	%URI; [p.266]	#IMPLIED			URI for an external script
src [p.225]	INPUT	%URI; [p.266]	#IMPLIED			for fields with images
src [p.210]	FRAME, IFRAME	%URI; [p.266]	#IMPLIED		F	source of frame content
src [p.161]	IMG	%URI; [p.266]	#REQUIRED			URI of image to embed
standby [p.164]	OBJECT	%Text; [p.266]	#IMPLIED			message to show while loading
start [p.105]	OL	NUMBER [p.50]	#IMPLIED	D	L	starting sequence number
style [p.186]	All elements but BASE, BASEFONT, HEAD, HTML, META, PARAM, SCRIPT, STYLE, TITLE	%StyleSheet, [p.266]	#IMPLIED			associated style info
summary [p.113]	TABLE	%Text; [p.266]	#IMPLIED			purpose/structure for speech output

## Index of the HTML 4 Attributes

tabindex [p.241]	A, AREA, BUTTON, INPUT, OBJECT, SELECT, TEXTAREA	NUMBER [p.50]	#IMPLIED			position in tabbing order
target [p.212]	A, AREA, BASE, FORM, LINK	%FrameTarget; [p.281]	#IMPLIED		L	render in this frame
text [p.69]	BODY	%Color; [p.281]	#IMPLIED	D	L	document text color
title [p.63]	All elements but BASE, BASEFONT, HEAD, HTML, META, PARAM, SCRIPT, TITLE	%Text; [p.266]	#IMPLIED			advisory title
type [p.149]	A, LINK	%ContentType; [p.266]	#IMPLIED			advisory content type
type [p.163]	OBJECT	%ContentType; [p.266]	#IMPLIED			content type for data
type [p.167]	PARAM	%ContentType; [p.266]	#IMPLIED			content type for value when valuetype=ref
type [p.252]	SCRIPT	%ContentType; [p.266]	#REQUIRED			content type of script language
type [p.187]	STYLE	%ContentType; [p.266]	#REQUIRED			content type of style language
type [p.225]	INPUT	%InputType; [p.273]	TEXT			what kind of widget is needed
type [p.105]	LI	%LIStyle; [p.289]	#IMPLIED	D	L	list item style
type [p.105]	OL	%OLStyle; [p.288]	#IMPLIED	D	L	numbering style
type [p.105]	UL	%ULStyle; [p.288]	#IMPLIED	D	L	bullet style
type [p.229]	BUTTON	(button   submit   reset)	submit			for use as form button
usemap [p.175]	IMG, INPUT, OBJECT	%URl; [p.266]	#IMPLIED			use client-side image map
valign [p.132]	COL, COLGROUP, TBODY, TD, TFOOT, TH, THEAD, TR	(top   middle   bottom   baseline)	#IMPLIED			vertical alignment in cells
value [p.225]	INPUT	CDATA [p.50]	#IMPLIED			Specify for radio buttons and checkboxes
value [p.232]	OPTION	CDATA [p.50]	#IMPLIED			defaults to element content

## Index of the HTML 4 Attributes

value [p.167]	PARAM	CDATA [p.50]	#IMPLIED			property value
value [p.229]	BUTTON	CDATA [p.50]	#IMPLIED			sent to server when submitted
value [p.106]	LI	NUMBER [p.50]	#IMPLIED	D	L	reset sequence number
valuetype [p.167]	PARAM	(DATA   REF   OBJECT)	DATA			How to interpret value
version [p.61]	HTML	CDATA [p.50]	%HTML.Version; [p.279]	D	L	Constant
viink [p.69]	BODY	%Color; [p.281]	#IMPLIED	D	L	color of visited links
vspace [p.180]	APPLET, IMG, OBJECT	%Pixels; [p.285]	#IMPLIED	D	L	vertical gutter
width [p.202]	HR	%Length; [p.285]	#IMPLIED	D	L	
width [p.217]	IFRAME	%Length; [p.285]	#IMPLIED		L	frame width
width [p.179]	IMG, OBJECT	%Length; [p.270]	#IMPLIED			override width
width [p.113]	TABLE	%Length; [p.270]	#IMPLIED			table width
width [p.126]	TD, TH	%Length; [p.285]	#IMPLIED	D	L	width for cell
width [p.171]	APPLET	%Length; [p.285]	#REQUIRED	D	L	initial width
width [p.121]	COL	%MultiLength; [p.270]	#IMPLIED			column width specification
width [p.119]	COLGROUP	%MultiLength; [p.270]	#IMPLIED			default width for enclosed COLs
width [p.97]	PRE	NUMBER [p.50]	#IMPLIED	D	L	

## ประวัติผู้เขียน

นายณัฐพล สืบอ่ำ เกิดเมื่อวันที่ 24 กันยายน พ.ศ.2524 ที่จังหวัดตาก สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมโทรคมนาคม ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2545 และเข้าศึกษาต่อในระดับปริญญาโท หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมสารสนเทศ ภาควิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2547 โดยในปี พ.ศ. 2545 ได้เข้าทำงานในตำแหน่งวิศวกรระบบ บมจ.พรีอเพอร์ดีเพอร์เฟค โดยรับผิดชอบการวางระบบ โครงข่าย Server และซ่อมบำรุงเครื่องคอมพิวเตอร์ ในปัจจุบันได้ดำรงตำแหน่งผู้ช่วยผู้จัดการ ฝ่ายพัฒนาธุรกิจวางแผนและการตลาด สายงานเทคโนโลยี Online บมจ.พรีอเพอร์ดีเพอร์เฟค โดยรับผิดชอบการในการหานวัตกรรมใหม่ๆ ในเชิงเทคโนโลยีสารสนเทศ มาประยุกต์ใช้กับทางธุรกิจในปัจจุบัน