

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เว็บเบสแอปพลิเคชันสำหรับเวอร์ชวลเอิร์ทโลคอลไลฟ์

Web Base Application For Virtual Earth Local Live

นายธีรพงษ์ พิเภา

นายอุทัย รัตนภาคโสภณ

2/4

๓๖๓๑๖

๕๕๕๙

เลขหมู่.....
เลขทะเบียน.....72947
วัน,เดือน,ปี 26 ส.ย. 2550

b.....11๓๗๕๑๑๗
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เว็บเบสแอปพลิเคชันสำหรับเวอร์ชวลเอิร์ทโลคอลไลฟ์

Web Base Application For Virtual Earth Local Live

โดย

นายธีรพงษ์ พิเภา

นายอุทัย รัตนภาคโสภณ

อาจารย์ที่ปรึกษา

ผศ.ดร. สุรินทร์ กิตติธรรมกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เว็บเบสแอปพลิเคชันสำหรับเวอร์ชวลเอิร์ทโลกคอลไลฟ์

Web Base Application For Virtual Earth Local Live

ผู้จัดทำ

1. นายธีรพงษ์ พิเมก รหัสนักศึกษา 47015321
2. นายอุทัย รัตนภาคโสภณ รหัสนักศึกษา 47015349

ศ. สุรินทร์ กิตติขจรกุล อาจารย์ที่ปรึกษา
(ผศ.ดร. สุรินทร์ กิตติขจรกุล)

เว็บแอปพลิเคชันสำหรับเวอร์ชวลเิร์ทโลกคอลไฟ์

นายธีรพงษ์ พิเณก	47015321
นายอุทัย รัตนภาคโสภณ	47015349
ผศ.ดร. สุรินทร์ กิตติธรรมกุล	อาจารย์ที่ปรึกษา
ปีการศึกษา 2549	

บทคัดย่อ

แผนที่ในปัจจุบันมีอยู่ 2 รูปแบบใหญ่นั้นคือ แบบ offline และแบบ online โดยที่ปัญหาของแบบ offline คือ ข้อมูลบนแผนที่ไม่ทันสมัย ไม่สามารถแสดงผลข้อมูลที่กำลังเคลื่อนไหวอยู่ได้ เพราะฉะนั้น แผนที่แบบออนไลน์จึงเป็นที่นิยมมากกว่า แต่ปัญหาสำหรับแผนที่ออนไลน์นั้นคือ ข้อมูลของประเทศไทยมีน้อย ไม่สามารถนำไปใช้งานได้เท่าที่ควร เราจึงทำการพัฒนาเว็บแอปพลิเคชันสำหรับเวอร์ชวลเิร์ทโลกคอลไฟ์ขึ้นมา โดยเราใช้ภาพแผนที่จากเวอร์ชวลเิร์ทโลกคอลไฟ์ และเราทำการเพิ่มข้อมูลของประเทศไทยเข้าไป โดยเราสามารถเพิ่มเลเยอร์ของข้อมูลได้ ด้วย เช่น เลเยอร์ของ ถนน หรือ แม่น้ำ เมื่อนำเลเยอร์มาใช้จะทำให้การโหลดข้อมูลเร็วขึ้น และเลือกเฉพาะข้อมูลที่เราสนใจ ขึ้นมาแสดงผลได้ และเราได้นำเอาเวอร์ชวลเิร์ทโลกคอลไฟ์มาประยุกต์ในการติดตามวัตถุที่เราต้องการ โดยสามารถแสดงผลได้ 2 แบบ นั่นคือ แบบ Real-time และ แบบ History-Time โดยใช้ข้อมูลจากอุปกรณ์ที่ติดตั้งไว้มาเก็บไว้ที่เว็บแอปพลิเคชันของเรา แล้วนำมาแสดงผล

Web Base Application For Virtual Earth Local Live

Abstract

Until now, there have been offline and online maps. Offline map has some problems for example data are not up to date and cannot display real-time data. Therefore, online map can be more popular. But in Thailand, online map has less data. Hence, we develop “Web Base Application Virtual Earth Local Live” to support layers of geographic data. As a result, user can view only a particular type of locations, e.g. bank layer, temple layer, etc. When the data layer is applied, the map can be loaded faster. In addition, we can apply Virtual Earth Local Live to display track motion objects.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยข้าพเจ้าได้รับคำแนะนำ และคำปรึกษาจาก ผศ.ดร. สุรินทร์ กิตติธรรมา ซึ่งเป็นอาจารย์ที่ปรึกษา ข้าพเจ้ารู้สึกซาบซึ้งในความอนุเคราะห์จากท่านอาจารย์ และขอขอบพระคุณเป็นอย่างสูง

ขอกราบพระคุณคณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุก ๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณเพื่อน ๆ พี่ ๆ น้อง ๆ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่าง ๆ และคอยให้กำลังใจเสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่อง ๆ ทำให้ข้าพเจ้าสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและคุณประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอบแต่ผู้มีพระคุณทุกท่าน

นายธีรพงษ์ พิภก

นายอุทัย รัตนภาคโสภณ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
กิตติกรรมประกาศ.....	II
สารบัญ.....	III
สารบัญภาพ.....	VI
สารบัญตาราง.....	VII

บทที่ 1 บทนำ

1.1 ความเป็นมาและเหตุผล.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ.....	1
1.4 ขอบเขตของโครงการ.....	2
1.5 เนื้อหาของรายงาน.....	2

บทที่ 2 เทคโนโลยีที่เกี่ยวข้องกับเว็บแอปพลิเคชัน

2.1 ทฤษฎีเบื้องต้นของ PHP.....	3
2.2 ประวัติของ PHP.....	3
2.3 กลไกการทำงานของเว็บเพจกับ PHP.....	3
2.4 การประมวลผลไฟล์ PHP.....	5
2.5 ความสามารถของ PHP.....	6

บทที่ 3 เทคโนโลยีที่เกี่ยวข้องกับการพัฒนาแผนที่

3.1 เทคโนโลยี AJAX (Asynchronous JavaScript and XML).....	9
3.1.1 รูปแบบการใช้งาน AJAX.....	12

สารบัญ (ต่อ)

3.2 ภาษา XML (Extensible Markup Language).....	13
3.2.1 ลักษณะโครงสร้างของ XML.....	13
3.2.2 ตัวกำหนดโครงสร้างของ XML ด้วย DTD	15
3.2.3 การแปลงรูปแบบเอกสาร XML (Transforming XML).....	17
3.2.4 DOM (Document Object Model).....	18
3.2.5 RSS (Really Simple Syndication).....	19
3.2.6 GeoRSS (Geographically Encoded Objects for RSS feeds).....	20
 บทที่ 4 Virtual Earth Local Live	
4.1 ความเป็นมาของ Virtual Earth Local live.....	21
4.2 การใช้งาน Virtual Earth Map	22
4.3 การใช้งาน Virtual Earth Map	22
4.4 การเริ่มต้นใช้งาน Virtual Earth map control SDK.....	25
 บทที่ 5 การออกแบบและขั้นตอนการดำเนินงาน	
5.1 การทำงานของระบบ.....	31
5.2 ส่วนประกอบของระบบ	33
5.3 ส่วนของ Layer บนแผนที่.....	34
5.4 ส่วนการติดตามวัตถุใด.....	36
 บทที่ 6 การทดลองและผลการทดลอง	
6.1 บทนำ การทดลองและผลการทดลอง.....	39
6.2 ขั้นตอนการทดลอง.....	39

บทที่ 7 สรุปผลการดำเนินงาน

7.1 สรุปผลการทดลอง.....	45
7.2 ปัญหาและอุปสรรค.....	45
7.3 ข้อเสนอแนะและแนวทางการแก้ปัญหา.....	45
7.4 แนวทางการพัฒนาต่อ.....	46

สารบัญภาพ

รูปที่ 2-1 แสดงการทำงานของเว็บเพจปกติ.....	4
รูปที่ 2-2 แสดงการทำงานของเว็บเพจที่มี PHP script.....	4
รูปที่ 2-3 รูปแสดงผลบนบราวเซอร์จากการทำงานของสคริปต์ PHP.....	5
รูปที่ 3.1 เปรียบเทียบระบบแอปพลิเคชันแบบมาตรฐานกับแบบที่ใช้ AJAX.....	9
รูปที่ 3.2 ระบบเว็บแอปพลิเคชันแบบมาตรฐาน (Synchronize)	10
รูปที่ 3.3 ระบบเว็บแอปพลิเคชันที่ใช้ AJAX (Asynchronous).....	10
รูปที่ 3.4 Data.php.....	12
รูปที่ 3.5 ตัวอย่างของส่วนที่ส่งข้อมูลไปให้ Web server.....	12
รูปที่ 3.6 ตัวอย่างการใช้ DOM และ Tag xml.....	13
รูปที่ 3.7 การใช้งาน DOM.....	18
รูปที่ 3.8 GeoRSS Model.....	20
รูปที่ 3.9 GeoRSS ในรูปแบบของ Point.....	20
รูปที่ 5.1 รับข้อมูลจากวัตถุ.....	31
รูปที่ 5.2 โหลดแผนที่จาก Local live.....	32
รูปที่ 5.3 แสดงภาพรวมของระบบ.....	32
รูปที่ 5.4 หลักการทำงานของระบบ.....	33
รูปที่ 5.5 แสดงการทำงานของระบบ.....	34
รูปที่ 5.6 แสดงรูปแบบของ GeoRSS.....	34
รูปที่ 5.7 แสดงการตีกรอบของ Layer.....	35
รูปที่ 5.8 รูปแบบข้อความที่ Server ส่งกลับมายัง User.....	36
รูปที่ 5.9 แสดงการทำงานของระบบ.....	37

รูปที่ 5.10 รูปแบบข้อความที่ Server ส่งกลับมายัง User.....	38
รูปที่ 6.1 แสดงระบบแผนที่.....	39
รูปที่ 6.2 ตัวอย่างแสดง Layer.....	40
รูปที่ 6.3 ตัวอย่างแสดงทีละหลายๆ Layer.....	41
รูปที่ 6.4 ตัวอย่างแสดงทีละหลายๆ Layer และเป็นภาษาไทย.....	41
รูปที่ 6.5 ตัวควบคุมในการนำข้อมูลของวัตถุมาแสดง.....	42
รูปที่ 6.6 แสดงการนำวัตถุมาแสดงบนแผนที่.....	43
รูปที่ 6.7 แสดงข้อความเมื่อไม่มีข้อมูล.....	44

สารบัญตาราง

ตารางที่ 2-1 แสดงชนิดของแอตทริบิวต์ที่สามารถใช้ในการประกาศแอตทริบิวต์	
ส่วนที่เป็นคำดีพอลต์.....	8
ตารางที่ 3-1 แสดงค่าที่สามารถได้ในส่วนที่เป็นคำดีพอลต์ในการประกาศแอตทริบิวต์.....	17
ตารางที่ 3-2 แสดงค่าที่สามารถได้ในส่วนที่เป็นคำดีพอลต์ในการประกาศแอตทริบิวต์.....	17

ภาคผนวก

ก. Source code โปรแกรม ไฟล์ index.php.....	47
ข. Schema ของฐานข้อมูล	
รูปที่ 1 รูปที่ 1 Schema table layer.....	51
รูปที่ 2 Schema table hospital.....	51
รูปที่ 3 Schema table real_history.....	51

บทที่ 1

บทนำ

1.1 ความเป็นมาและเหตุผล

ในปัจจุบันแผนที่ได้มีความสำคัญเพิ่มมากขึ้น โดยเฉพาะในตัวเมืองหรือจังหวัดที่มีขนาดใหญ่เราจึงมีแนวคิดที่จะนำแผนที่ Microsoft virtual earth local live ซึ่งเป็น free software มานำเสนอเส้นทางการเดินทางและสถานที่ ที่สำคัญต่างๆบนแผนที่ประเทศไทย เพื่อความสะดวกสบายในการเลือกเส้นทางการเดินทางหรือสถานที่ที่ต้องการจะไป เช่น University, Bank Police, Hospital ฯลฯ และเราสามารถ update เพิ่มสถานที่ที่เราต้องการได้ด้วยโปรแกรมเมอร์

สามารถติดตามวัตถุ (Object) ที่เราต้องการติดตามมาแสดงผลบนแผนที่ สามารถแสดงผลได้ทั้งแบบ real time และ history โดยใช้ hardware เป็นตัวส่งข้อมูลมาให้ ซึ่งเราไม่ได้ทำตัว hardware เราทำตัว software

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อติดตามวัตถุที่ต้องการแล้วนำข้อมูลที่ต้องการมาแสดงบนแผนที่
- 1.2.2 เพื่อความสะดวกสบายในการดูแผนที่บนอินเทอร์เน็ต
- 1.2.3 ช่วยให้มีข้อมูลเส้นทางการตัดสินใจเพิ่มมากขึ้นเพื่อประหยัดเวลาและสะดวกสบายยิ่งขึ้น
- 1.2.4 นำเอาข้อมูลหลายประเภทมาแสดงบนแผนที่ได้

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1.3.1 ช่วยค้นหาเส้นทางที่ดีที่สุดในการเดินทาง
- 1.3.2 ตรวจสอบตำแหน่งที่ต้องการ โดยได้ข้อมูลที่ถูกต้อง
- 1.3.3 ระบบแผนที่สามารถรองรับวัตถุต่างๆที่ต้องการติดตามหรือรับข้อมูลของวัตถุนั้นๆ ได้หลายชนิด
- 1.3.4 ช่วยควบคุมสิ่งที่เราต้องการติดตามได้ เช่นรถขนเงิน,เครื่องบิน เป็นต้น

1.4 ขอบเขตของโครงการ

1.4.1 สามารถนำข้อมูลของวัตถุที่ติดตามมาแสดงบนระบบแผนที่ได้

1.4.2 สามารถนำข้อมูลของแต่ละ layer มาแสดงบนแผนที่เวอร์ชวลเิร์ทโลกออนไลน์ได้

1.5 เนื้อหาของรายงาน

เนื้อหาของรายงานฉบับนี้มีทั้งหมด 5 บท โดยแต่ละบทมีเนื้อหาโดยสรุปดังนี้คือ

- บทที่ 1 เป็นเนื้อหาในส่วนของการนำซึ่งจะกล่าวถึงความเป็นมาและแนวความคิดการทำโครงการนี้ซึ่งจะประกอบด้วย วัตถุประสงค์ของการทำโครงการ ประโยชน์ที่คาดว่าจะได้รับ และขอบเขตของโครงการ

- บทที่ 2, 3, 4 กล่าวถึงเทคโนโลยีที่นำมาใช้ในการพัฒนาระบบในบทที่ 2 จะกล่าวถึงภาษา PHP ว่าการทำงานของภาษา PHP ว่ามีหลักการทำงานอย่างไร บทที่ 3 จะเป็นเทคโนโลยี AJAX DOM, XML ว่าใช้งานอย่างไร บทที่ 4 Microsoft Virtual Earth Local Live คืออะไรและนำมาใช้อย่างไร

- บทที่ 5 เป็นเนื้อหาในส่วนของการออกแบบและขั้นตอนการดำเนินงานระบบเวอร์ชวลเิร์ทโลกออนไลน์

บทที่ 2

เทคโนโลยีที่เกี่ยวข้องกับเว็บแอปพลิเคชัน

2.1 ทฤษฎีเบื้องต้นของ PHP

PHP เป็นภาษาสคริปต์ที่ทำงานทางฝั่งเซิร์ฟเวอร์ (server-side scripting language) ซึ่งมีลักษณะเป็น embedded script นั่นคือเราสามารถฝังคำสั่ง PHP ไว้ในเว็บเพจร่วมกับคำสั่ง (tag) ของภาษา HTML ได้

2.2 ประวัติของ PHP

Rasmus Lerdorf สร้างภาษา PHP ขึ้นมาในปี ค.ศ.1994 เนื่องจากเขาต้องการพัฒนาโปรแกรมเพื่อเก็บข้อมูลของผู้ใช้ที่แวะเวียนเข้ามาเยี่ยมชมโฮมเพจส่วนตัวของเขาเอง เขาเรียกโปรแกรมนี้อีกว่า PHP ซึ่งย่อมาจาก Personal Home Page Tools

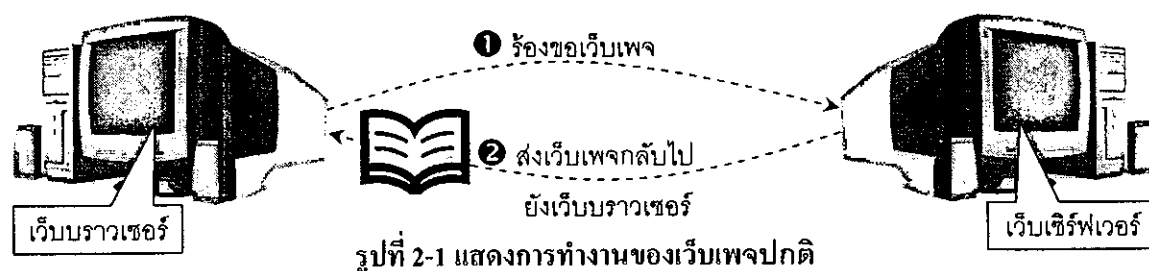
ในเวอร์ชันแรกสุดนั้น PHP ยังไม่มีความสามารถอะไรมากนัก โดยประกอบด้วยกลไกการแปลภาษาอย่างง่าย และชุดคำสั่ง/มาโคร ที่เป็นประโยชน์ต่อการสร้างสมุดเยี่ยม (guest book) และตัวนับจำนวนผู้เข้าชมเว็บ (counter) เท่านั้น

พอกกลางปี ค.ศ.1995 เขาก็ได้พัฒนาตัวแปลภาษา PHP ขึ้นมาใหม่โดยใช้ชื่อว่า PHP/FI version 2 ซึ่งได้เพิ่มความสามารถในการรับข้อมูลที่ตั้งมาจากฟอร์มของ HTML (จึงมีชื่อว่า FI หรือ Form Interpreter) นอกจากนั้นยังเพิ่มความสามารถในการติดต่อกับฐานข้อมูล mySQL อีกด้วย จึงทำให้ผู้คนเริ่มหันมาสนใจ PHP มากขึ้น

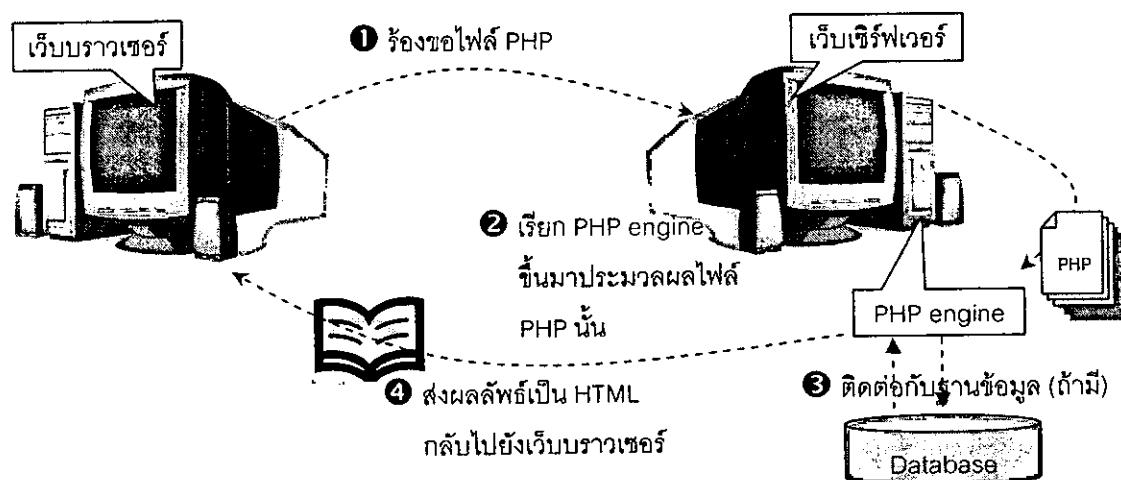
ในปี 1997 มีผู้ร่วมพัฒนา PHP เพิ่มอีก 2 คนคือ Zeev Suraski และ Andi Gutmans โดยได้แก้ไขข้อบกพร่องต่างๆ และเพิ่มเครื่องมือให้มากขึ้น กลายเป็น PHP version 3 และพัฒนาต่อมาจนถึง version 4 (PHP4) ในปัจจุบัน

2.3 กลไกการทำงานของเว็บเพจกับ PHP

สำหรับเว็บเพจธรรมดาที่โดยปกติมีนามสกุลของไฟล์เป็น htm หรือ html นั้นเมื่อเราใช้เว็บเบราว์เซอร์เปิดดูเว็บเพจใด เว็บเซิร์ฟเวอร์ก็จะส่งเว็บเพจนั้นกลับมายังเว็บเบราว์เซอร์ จากนั้นเว็บเบราว์เซอร์จะแสดงผลไปตามคำสั่งภาษา HTML ที่อยู่ในไฟล์ ซึ่งการทำงานในลักษณะนี้ทำให้ได้เว็บเพจที่เป็นแบบ static ซึ่งผู้ใช้ไม่สามารถเปลี่ยนแปลงหรือใช้งานในลักษณะที่แตกต่างได้



PHP เป็นภาษาที่ทำงานในฝั่งเซิร์ฟเวอร์ การใช้ PHP ในการสร้างเว็บเพจที่เป็นแบบ dynamic ทำโดยการฝังสคริปต์ภาษา PHP ไว้ในเว็บเพจนั้น(และมีนามสกุลเป็น .PHP) เมื่อเว็บเบราว์เซอร์ร้องขอไฟล์ PHP ไฟล์ใด เว็บเซิร์ฟเวอร์จะเรียก PHP engine ขึ้นมาแปล (interpret) และประมวลผลคำสั่งที่อยู่ในไฟล์ PHP นั้น โดยอาจมีการดึงข้อมูลจากฐานข้อมูล หรือเขียนข้อมูลลงไปยังฐานข้อมูลด้วย หลังจากนั้นผลลัพธ์ในรูปแบบ HTML (และสคริปต์ที่ทำงานฝั่งเบราว์เซอร์ เช่น client-side JavaScript) จะถูกส่งกลับไปยังเบราว์เซอร์ เบราว์เซอร์ก็จะแสดงผลตามคำสั่ง HTML ที่ได้รับมา ซึ่งย่อมไม่มีคำสั่ง PHP ใดๆ หลงเหลืออยู่ เนื่องจากถูกแปลและประมวลผลโดย PHP engine ที่ฝั่งเซิร์ฟเวอร์ไปหมดแล้ว

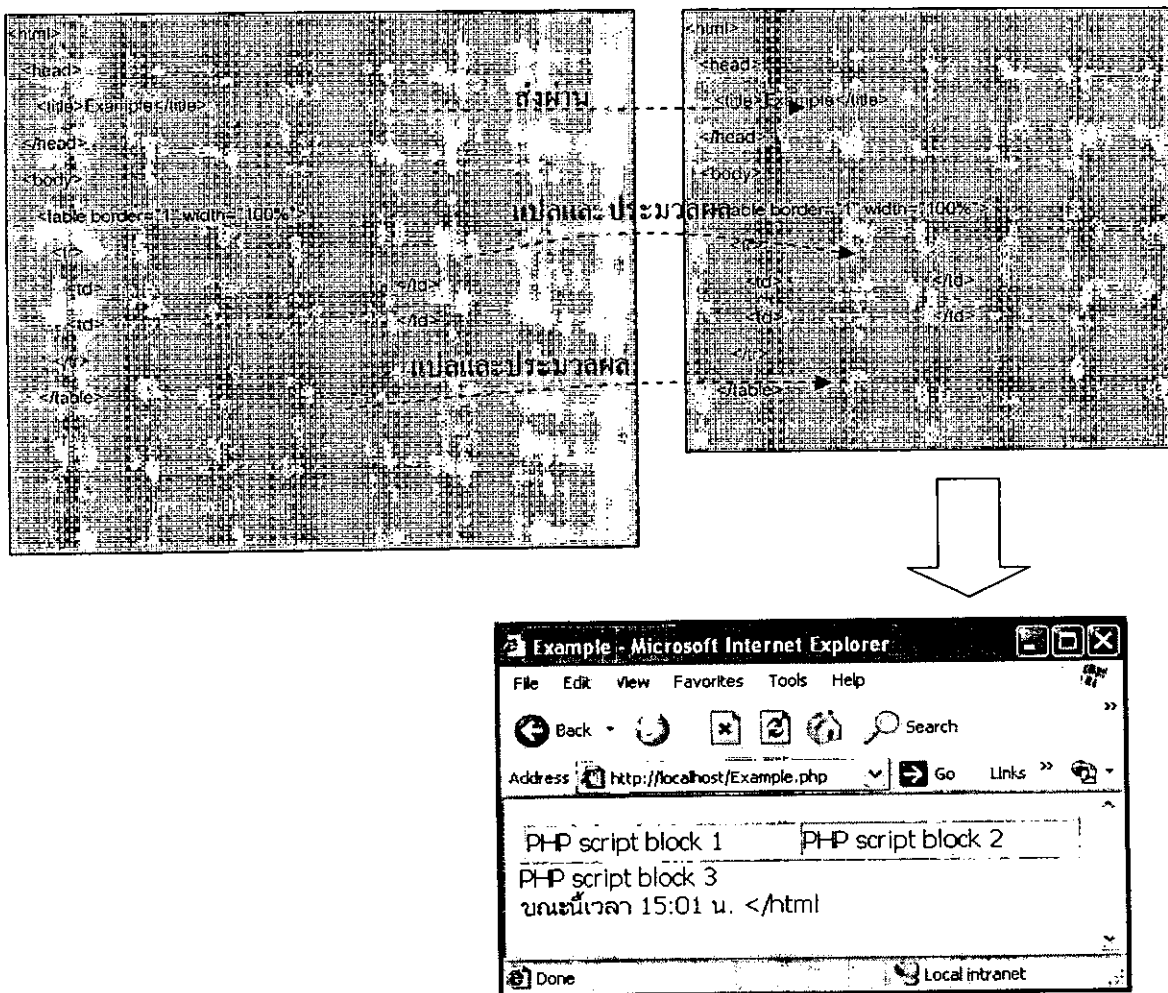


รูปที่ 2-2 แสดงการทำงานของเว็บเพจที่มี PHP script

จะเห็นว่าการทำงานของเว็บเบราว์เซอร์ในกรณีที่ร้องขอไฟล์ PHP ไม่แตกต่างจากกรณีที่ร้องขอไฟล์เว็บเพจธรรมดาเลย เพราะสิ่งที่เบราว์เซอร์ต้องกระทำก็คือการร้องขอไฟล์จากเว็บเซิร์ฟเวอร์ จากนั้นก็รอรับผลลัพธ์กลับมาแสดงผล ความแตกต่างจริงๆ อยู่ที่การทำงานฝั่งเซิร์ฟเวอร์ ซึ่งกรณีที่มีการเรียกไฟล์ PHP จะผ่านการประมวลผลก่อน แทนที่จะถูกส่งไปยังเบราว์เซอร์เลยทันที

2.4 การประมวลผลไฟล์ PHP

PHP engine จะแปลและประมวลผลเฉพาะคำสั่งที่อยู่ภายในแท็กของ PHP เท่านั้น การทำงานที่เกิดขึ้นคือ หลังจาก PHP engine ถูกเว็บเซิร์ฟเวอร์เรียกขึ้นมาประมวลผลไฟล์ PHP แล้ว มันจะส่งผ่านเนื้อหาของไฟล์ไปยังบราวเซอร์โดยไม่ทำอะไรกับเนื้อหา นั้น ยกเว้นเมื่อพบกับสัญลักษณ์ (tag) ที่ระบุจุดเริ่มต้นของบล็อกคำสั่ง PHP มันก็จะแปลและประมวลผลคำสั่งต่างๆ ไปตามลำดับ (ภายในบล็อก PHP นี้ การส่งผลลัพธ์ให้แก่บราวเซอร์ จะต้องเรียกใช้ คำสั่ง/ฟังก์ชัน ของ PHP เช่น echo หรือ print) โดยเมื่อพบสัญลักษณ์ปิดท้ายบล็อกคำสั่ง PHP PHP engine ก็จะหันกลับมาส่งผ่านเนื้อหาของไฟล์ต่อไปเช่นเดิม จนกว่าจะพบสัญลักษณ์ระบุจุดเริ่มต้นของบล็อกคำสั่ง PHP อีก และเป็นอย่างนี้เรื่อยไปจนจบไฟล์



รูปที่ 2-3 รูปแสดงผลบนบราวเซอร์จากการทำงานของสคริปต์ PHP

2.5 ความสามารถของ PHP

PHP ได้รับการพัฒนาความสามารถขึ้นมาเรื่อยๆ อย่างต่อเนื่อง ทั้งนี้เป็นเพราะมีการเปิดเผยซอร์สโค้ดของ PHP ผู้สาธารณะในลักษณะของ open source ทำให้มีหน่วยงานและองค์กรต่างๆ เข้ามาช่วยกันพัฒนา ในที่นี้จะขอยกกล่าวถึงความสามารถหลักของ PHP เท่านั้น ดังนี้

- ความสามารถในการจัดการกับตัวแปรหลายๆ ประเภท เช่น เลขจำนวนเต็ม(integer), เลขทศนิยม(float), สตริง(string) และอาร์เรย์(array) เป็นต้น
- ความสามารถในการรับข้อมูลจากฟอร์มของ HTML
- ความสามารถในการรับ-ส่ง Cookies
- ความสามารถเกี่ยวกับ Session (ตั้งแต่ PHP เวอร์ชัน 4 ขึ้นไป)
- ความสามารถทางด้าน OOP (Object Oriented Programming) ซึ่งรองรับการเขียนโปรแกรมเชิงวัตถุ
- ความสามารถในการเรียกใช้ COM component
- ความสามารถในการติดต่อและจัดการฐานข้อมูล
- ความสามารถในการสร้างภาพกราฟิก

JavaScript

Introduction JavaScript เป็นเทคโนโลยีทางด้านการพัฒนาเว็บไซต์ เกิดขึ้นมาใหม่อย่างต่อเนื่อง เป็นผลทำให้ผู้ใช้สามารถเลือกใช้เทคโนโลยีเหล่านี้มาพัฒนาเว็บไซต์ของตน ให้เกิดความสวยงามได้ JavaScript ก็เป็นหนึ่งในเทคโนโลยีที่ผู้พัฒนาเว็บไซต์นิยมนำมาใช้กับเว็บไซต์ของตน ด้วยความสามารถที่เหนือกว่า HTML ปกติ ทำให้สามารถให้ผู้พัฒนาเว็บไซต์ สร้างลูกเล่นต่างๆ ให้กับเว็บไซต์ของตนเองได้ แต่ในความสามารถที่มีนั้น บางอย่างเป็นอุปสรรคกับกลุ่มคน กลุ่มหนึ่ง ซึ่งเป็นกลุ่มที่ต้องอาศัย โปรแกรมบางอย่างที่เรียกว่า โปรแกรมอ่านหน้าจอ สำหรับอ่านข้อความบนเว็บไซต์ กลุ่มคนกลุ่มนี้คือ กลุ่มคนพิการที่มีความบกพร่องทางการเห็น หรือ คนตาบอดนั่นเอง ซึ่งความจริงก็คือ คนตาบอด ไม่สามารถมองเห็นได้ว่าตำแหน่งของ Mouse ที่กำลังชี้ขี้อยู่ที่ไหน แต่คนตาบอดสามารถจะรู้ได้จาก การใช้ Keyboard เลือกให้ไปโฟกัสในแต่ละ Object บนหน้าเว็บ ฉะนั้นแล้ว การ Coding ใน JavaScript ที่มีฟังก์ชันการรับ Input จาก Mouse อย่างเดียว โดยไม่มีส่วนของ Keyboard ด้วยนั้นจะทำให้เว็บไซต์นั้น

JavaScript คือ JavaScript เป็นภาษาโปรแกรม (programming language) ประเภทหนึ่ง ที่เรียกกันว่า "สคริปต์" (script) ซึ่งมีวิธีการทำงานในลักษณะ "แปลความและดำเนินงานไปที่ละคำสั่ง" (interpret) ภาษานี้เดิมมีชื่อว่า LiveScript ได้รับการพัฒนาขึ้นโดย Netscape ด้วยวัตถุประสงค์ เพื่อที่จะช่วยให้เว็บเพจสามารถแสดงเนื้อหา ที่มีการเปลี่ยนแปลงไปได้ ตามเงื่อนไข

หรือสภาพแวดล้อมต่างๆกัน หรือสามารถโต้ตอบกับผู้ชมได้มากขึ้น ทั้งนี้เพราะภาษา HTML เดิม นั้น เหมาะสำหรับใช้แสดงเอกสาร ที่มีเนื้อหาคงที่แน่นอน และไม่มีลูกเล่นอะไรมากมายนัก เนื่องจาก JavaScript ช่วยให้ผู้พัฒนา สามารถสร้างเว็บเพจได้ตรงกับความต้องการ และมีความน่าสนใจมากขึ้น ประกอบกับเป็นภาษาเปิด ที่ใครก็สามารถนำไปใช้ได้ ดังนั้นจึงได้รับความนิยมเป็นอย่างสูง มีการใช้งานอย่างกว้างขวาง รวมทั้งได้ถูกกำหนดให้เป็นมาตรฐานโดย ECMA ซึ่งเราจะพบว่าปัจจุบัน จะหาเว็บเพจที่ไม่ใช้ JavaScript เลยนั้น ได้ยากเต็มที การทำงานของ JavaScript จะต้องมีการแปลความคำสั่ง ซึ่งขั้นตอนนี้จะถูกจัดการ โดยบราวเซอร์ ดังนั้น JavaScript จึงสามารถทำงานได้ เฉพาะบนบราวเซอร์ที่สนับสนุน ซึ่งปัจจุบันบราวเซอร์เกือบทั้งหมดก็สนับสนุน JavaScript แล้ว อย่างไรก็ตาม สิ่งที่ต้องระวังคือ JavaScript มีการพัฒนาเป็นเวอร์ชันใหม่ๆออกมาด้วย (ปัจจุบันคือรุ่น 1.5) ดังนั้น ถ้านำโค้ดของเวอร์ชันใหม่ ไปรันบนบราวเซอร์รุ่นเก่าที่ยังไม่สนับสนุน ก็อาจจะทำให้เกิด error ได้ การทำงานของ JavaScript เกิดขึ้นบนบราวเซอร์ (เรียกว่าเป็น client-side script) ดังนั้นไม่ว่าคุณจะใช้เซิร์ฟเวอร์อะไร หรือที่ไหน ก็ยังคงสามารถใช้ JavaScript ในเว็บเพจได้ ต่างกับภาษาสคริปต์อื่น เช่น Perl, PHP หรือ ASP ซึ่งต้องแปลความและทำงานที่ตัวเครื่องเซิร์ฟเวอร์ (เรียกว่า server-side script) ดังนั้นจึงต้องใช้บนเซิร์ฟเวอร์ ที่สนับสนุนภาษาเหล่านี้เท่านั้น อย่างไรก็ตาม จากลักษณะดังกล่าวก็ทำให้ JavaScript มีข้อจำกัด คือไม่สามารถรับและส่งข้อมูลต่างๆ กับเซิร์ฟเวอร์โดยตรง เช่น การอ่านไฟล์จากเซิร์ฟเวอร์ เพื่อนำมาแสดงบนเว็บเพจ หรือรับข้อมูลจากผู้ชม เพื่อนำไปเก็บบนเซิร์ฟเวอร์ เป็นต้น ดังนั้นงานลักษณะนี้ จึงยังคงต้องอาศัยภาษา server-side script อยู่ (ความจริง JavaScript ที่ทำงานบนเซิร์ฟเวอร์เวอร์ก็มี ซึ่งต้องอาศัยเซิร์ฟเวอร์ที่สนับสนุนโดยเฉพาะเช่นกัน แต่ไม่เป็นที่นิยมนัก) การทำงานของ JavaScript จะมีประสิทธิภาพมาก ถ้ามันสามารถดัดแปลงคุณสมบัติ ขององค์ประกอบต่างๆ บนเว็บเพจ (เช่น สี หรือรูปแบบของข้อความ) และสามารถรับรู้เหตุการณ์ ที่ผู้ชมเว็บเพจโต้ตอบกับองค์ประกอบเหล่านั้น (เช่น การคลิก หรือเลื่อนเมาส์ไปวาง) ได้ ดังนั้นจากภาษา HTML เดิม ที่มีลักษณะสถิต (static) ใน HTML เวอร์ชันใหม่ๆ จึงได้มีการพัฒนาให้มีคุณสมบัติบางอย่างเพิ่มขึ้น และมีลักษณะเป็นอ็อบเจกต์ "object" มากขึ้น การทำงานร่วมกันระหว่างคุณสมบัติใหม่ของ HTML ร่วมกับ JavaScript นี้เอง ทำให้เกิดเป็นสิ่งที่เรียกว่า Dynamic HTML คือภาษา HTML ที่สามารถใช้สร้างเว็บเพจที่มีลักษณะพลวัต (dynamic) ได้นั่นเอง นอกจากนี้ อีกองค์ประกอบหนึ่งที่เกี่ยวข้อง ก็คือ Cascading Style Sheet (CSS) ซึ่งเป็นภาษาที่ช่วยให้เราควบคุมรูปแบบ ขององค์ประกอบต่างๆ บนเว็บเพจ ได้อย่างมีประสิทธิภาพมากกว่าคำสั่ง หรือแท็ก (tag) ปกติของ HTML เนื่องจาก JavaScript สามารถดัดแปลงคุณสมบัติของ CSS ได้เช่นกัน ดังนั้นมันจึงช่วยให้เราควบคุมเว็บเพจ ได้อย่างมีประสิทธิภาพมากยิ่งขึ้นไปอีก

ในการใช้งาน JavaScript เราจะต้องแทรกโค้ดของมัน เข้าไประหว่างแท็กของภาษา HTML ในเว็บเพจ โดยใช้แท็ก `<SCRIPT>...</SCRIPT>` ซึ่งมีวิธีใช้แบบง่ายๆ ดังนี้

<SCRIPT>

โค้ด JavaScript...

</SCRIPT>

อย่างไรดี ถ้าต้องการให้โค้ดชุดนี้สมบูรณ์ขึ้น เราควรเพิ่มองค์ประกอบอื่นๆ เสริมเข้าไปด้วย ซึ่งจะทำให้โค้ดมีลักษณะดังนี้

<SCRIPT LANGUAGE="JavaScript">

<!-- จุดเริ่มต้น

โค้ด JavaScript...//

จุดสิ้นสุด --></SCRIPT>

ลักษณะ / ภาษา	PHP	JavaScript
การทำงาน	server	client
โปรแกรมประมวลผล	Apache, IIS, ect..	IE, Firefox, ect..
ติดต่อฐานข้อมูล	ทำได้ง่าย	ทำได้ยาก
การประกาศตัวแปร	ไม่จำเป็นต้องระบุ data type	ไม่จำเป็นต้องระบุ data type
ตัวแปร	ใช้ \$ นำหน้าชื่อตัวแปร	ใช้ชื่อตัวแปรได้เลย

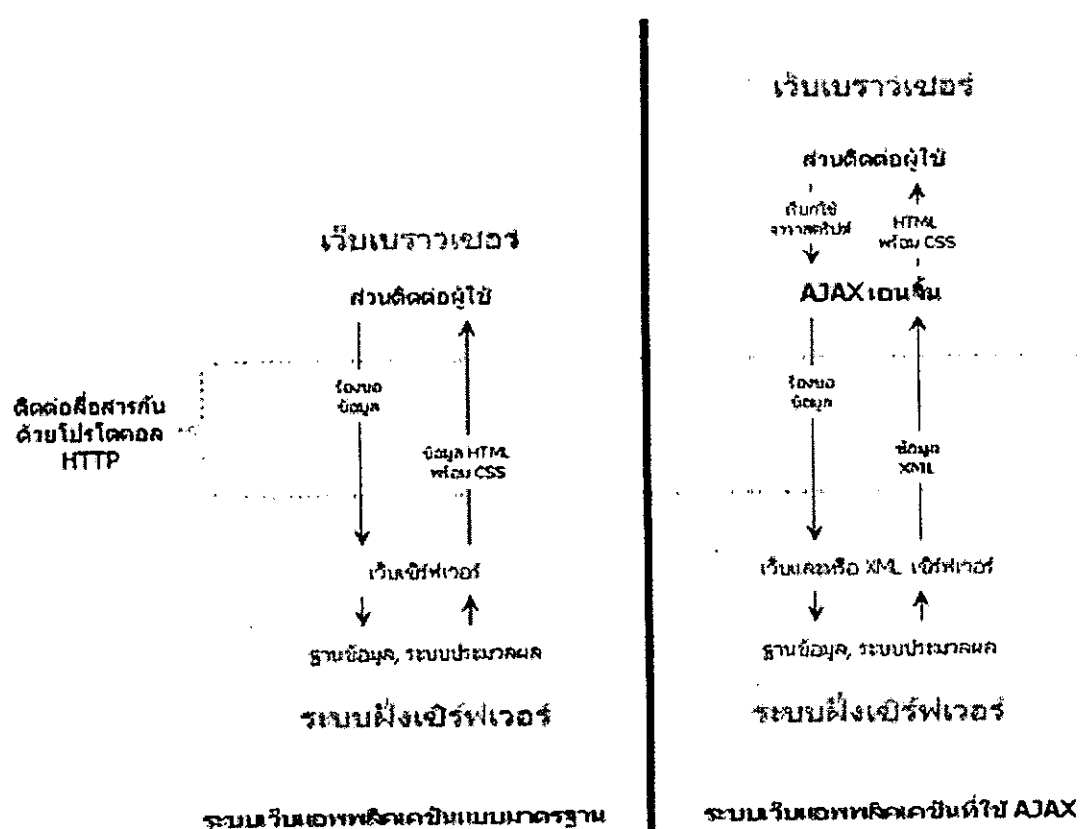
ตารางที่ 2-1 เปรียบเทียบ การทำงานของ PHP กับ JavaScript

บทที่ 3

เทคโนโลยีที่เกี่ยวข้องกับการพัฒนาแผนที่

3.1 เทคโนโลยี AJAX (Asynchronous JavaScript and XML)

เป็นเทคโนโลยีที่รวมเอาความสามารถของ JavaScript, XML, CSS และ XHTML เอาไว้ด้วยกัน Ajax เป็นการประยุกต์เอาเทคโนโลยีเก่ามาผสมผสานจนได้เทคโนโลยีใหม่ที่นำศึกษาและนำมาใช้งาน แต่ก่อนอื่นมาทำความเข้าใจหลักการทำงานของ Web กันก่อน

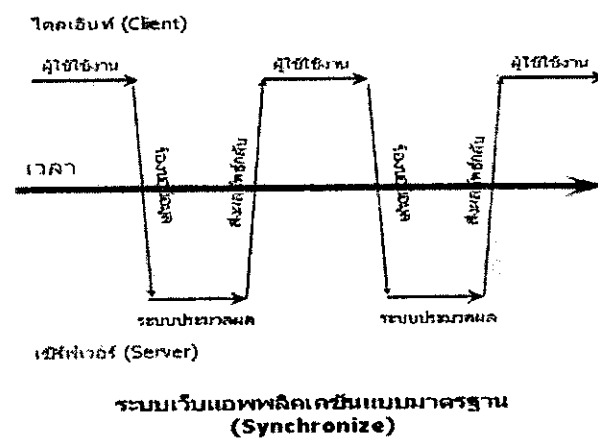


รูปที่ 3.1 เปรียบเทียบระบบแอปพลิเคชันแบบมาตรฐานกับแบบที่ใช้ AJAX

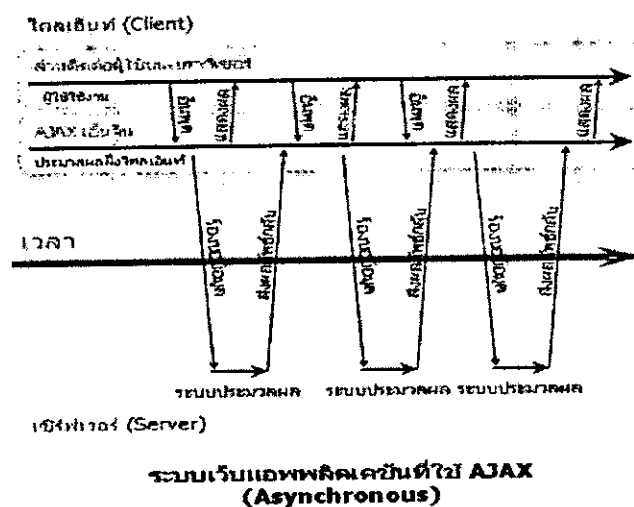
ตามปกติเมื่อเราเปิด Web Browser และพิมพ์ URL ของเว็บเพจที่ต้องการ เราจะเรียกผู้ใช้งาน client-side browser ก็จะส่งค่าไปยัง server เพื่อขอเปิดหน้า url ที่เราพิมพ์ลงไป เช่น www.google.co.th และเมื่อทาง server ได้รับค่าที่ส่งมาก็จะส่งหน้าเว็บเพจกลับมาให้ โดยเราจะเรียก server ว่าผู้ให้บริการหรือ server side เมื่อฝั่งผู้ใช้ได้รับข้อมูลจาก server ที่ส่งมาให้ browser ก็จะนำข้อมูลนั้นขึ้นหน้าจอ จากนั้นเมื่อเราคลิกเว็บหน้าอื่นๆก็จะเริ่มขั้นตอนทั้งหมดใหม่อีกครั้ง

ตัวอย่างเพื่อความเข้าใจจากโปรแกรมเครื่องคิดเลขแบบง่ายๆครับ เมื่อผู้ใช้ (client) ใสตัวเลขในค่าที่ 1 และ 2 แล้วคลิกเลือกว่าจะบวก ลบ คูณ หรือหาร เพื่อหาค่าที่ต้องการข้อมูลนี้ก็จะส่งไปยัง server เพื่อเอาค่าทั้งสองที่ผู้ใช้ป้อนไปหาค่าตามที่เลือกไว้ไปจัดการ

จะเห็นว่าเมื่อคลิกเลือก บวก ลบ คูณ หรือหาร server ก็จะเอาไฟล์ action="ไฟล์ที่จะส่งค่าไป" มาจัดการคำนวณค่าออกมา จากนั้นก็จะส่งหน้า webpage หน้าใหม่ que แสดงผลที่คำนวณตามเงื่อนไขออกมา วิธีการนี้หน้าจอ webpage จะต้องมีการ refresh ใหม่ และการรับผลที่ส่งกลับมาก็จะเป็นการส่งมาแบบทั้งหน้า webpage แบบเต็มๆ ทำให้กิน bandwidth



รูปที่ 3.2 ระบบเว็บแอปพลิเคชันแบบมาตรฐาน (Synchronize)



รูปที่ 3.3 ระบบเว็บแอปพลิเคชันที่ใช้ AJAX (Asynchronous)

การทำงานของ Ajax นั้นจะส่งเฉพาะข้อมูลที่ต้องการไปยัง server และส่งกลับมาเฉพาะข้อมูลที่ต้องการไม่ใช่การส่งทั้งหน้า webpage ใหม่ โดย Ajax อาศัย object ที่ชื่อ XMLHttpRequest เมื่อผู้ใช้เปิดหน้าเว็บแล้วมีการส่งข้อมูล Ajax ก็จะให้ XMLHttpRequest ส่งค่าไปให้ server แล้วให้ server จัดการข้อมูลนั้นตามเงื่อนไขแล้วส่งข้อมูลนั้นกลับมาในรูปแบบ XML ซึ่งก็จะใช้ javascript เป็นตัวจัดการข้อมูลที่ได้รับให้แสดงผลได้อย่างถูกต้องในหน้าเว็บเพจเดิม ตัวอย่างโปรแกรมเครื่องคิดเลขอย่างง่ายที่อาศัยการทำงานของ Ajax ดูได้ ข้อดีของ Ajax มีมากมายครับ ทั้งการแสดงผลที่เร็วกว่าไม่ต้อง refresh หน้าจอใหม่ทุกครั้งอีกทั้งข้อมูลที่ส่งไป-กลับไม่ได้ส่งไปทั้งหน้า ทำให้กิน bandwidth น้อยกว่า ตัวอย่างเว็บที่ใช้ Ajax ก็เช่น Google Mail

ก็อย่างที่กล่าวมา AJAX ไม่ใช่ ภาษาใหม่ที่ใช้มาเขียนเว็บแต่อย่างใดแต่เป็นวิธีการทำให้การทำงานระหว่าง web application กับผู้ใช้ให้มีประสิทธิภาพมากขึ้นเช่นการติดต่อกันระหว่าง web browser กับ web server เมื่อ Web browser กับ web server ต้องการที่จะแลกเปลี่ยนข้อมูลกันแทนที่จะโหลดหน้าเว็บใหม่ทั้งหมดเราต้องการที่จะโหลดข้อมูลบางส่วนเท่านั้น AJAX ยังคงใช้พื้นฐานของ JavaScript, XML, HTML, CSS

เมื่อก่อนเราใช้ Web application เมื่อเราคลิกปุ่ม submit input โดยใช้ HTML FORM ไปยัง web server หลังจากนั้น web server จะประมวลผลข้อมูลแล้วมันจะส่งค่ากลับมาให้ในรูปแบบของหน้าเว็บใหม่เมื่อ web server ส่งค่าที่เป็นหน้าเว็บใหม่อาจจะทำให้เกิดความล่าช้าในการโหลดข้อมูลใหม่ แต่ถ้าเราใช้ AJAX ในการส่งและรับข้อมูลเราก็ไม่ต้องโหลดเว็บใหม่ในการส่งข้อมูลเราจะส่งโดยใช้ HTTP request ไปยัง web server และการรับข้อมูลมาจาก web server จะมาในรูปแบบของ XML

3.1.1 รูปแบบการใช้งาน AJAX

- Web server เราต้องสร้างส่วนที่ประมวลผลแล้วส่งค่ากลับมาให้ยัง web browser ที่ร้องขอมาเช่นดังตัวอย่างเราจะสร้าง file ที่ชื่อว่า Data.php ตามรูปที่ 3.4

```
<?php
If (parameter == 1)
{
    Echo "<Item>";
    Echo "<name>Tong</name>";
    Echo "<age>23</age>";
    Echo "</Item>";
} // If
Else
{
    Echo "<Item>";
    Echo "<name>Eng</name>";
    Echo "<age>21</age>";
    Echo "</Item>";
} // Else
?>
```

รูปที่ 3.4 Data.php

- Web browser สร้างส่วนที่ส่งข้อมูลไปให้ web server และส่วนที่รับข้อมูลจาก web server ตามรูปที่ 3.5 คือตัวอย่างของส่วนที่ส่งข้อมูลไปให้ web server

```
xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
var url = "http://161.246.5.208/Map/T04.08.2549/data.php?=&parameter=1";
xmlHttp.onreadystatechange = stateChanged;
xmlHttp.open( "GET" ,url,true);
xmlHttp.send(null);
```

รูปที่ 3.5 ตัวอย่างของส่วนที่ส่งข้อมูลไปให้ Web server

ตามรูปที่ 3.5 เราสร้าง Object ที่เป็น ActiveXobject ที่บอกว่าเป็น browser ของ Microsoft, กำหนด url ที่จะส่งข้อมูลไปให้ยัง web server, กำหนด function เมื่อมีการรับข้อมูลกลับมา, กำหนดวิธี การส่งแบบ POST หรือ แบบ GET และกำหนด url จากนั้นก็ส่งข้อมูล

```

xmlDocument = new ActiveXObject('Microsoft.XMLDOM');
xmlDocument.loadXML(xmlHttp.responseText);
Item = xmlDocument.getElementsByTagName('Item');
nameTag = Item[0].getElementsByTagName('name');
ageTag = Item[0].getElementsByTagName('age');
name = nameTag[0].firstChild.data;
age = ageTag[0].firstChild.data;

```

รูปที่ 3.6 ตัวอย่างการใช้ DOM และ Tag xml

ตามรูปที่ 3.6 คือการรับข้อมูลจาก Web server โดยใช้เทคโนโลยี DOM ในการโหลดค่าออกมาจาก xmlHttp.responseText มาจากนั้นเป็นตัวอย่างว่าเอาค่าออกมาจาก Tag xml ยังไง

3.2 ภาษา XML (Extensible Markup Language)

XML ถูกกำหนดโดย กลุ่มทำงาน XML ของสถาบัน World Wide Web Consortium(W3C) กลุ่มทำงานกลุ่มนี้ ได้บรรยายถึงภาษา XML ไว้ว่า Extensible Markup Language เป็นฟอร์แมตที่อธิบายถึงรายละเอียดของโครงสร้าง และแบบของข้อมูลเป็นภาษาหรือชุดคำสั่งเกี่ยวกับข้อมูลบนเว็บที่ทำให้การพัฒนา ในส่วนของโครงสร้างข้อมูล จากหลากหลายแอปพลิเคชันมานำเสนอบนเครื่องเดสก์ท็อปด้วย XML จะทำให้การจัดการข้อมูลหรือเรียกใช้ข้อมูลจากแอปพลิเคชันต่างๆ จะเข้าสู่มาตรฐานเดียวกัน

XML และ HTML เป็นส่วนหนึ่งของภาษา SGML ซึ่ง XML จะให้รายละเอียดเกี่ยวกับข้อมูล เช่น ชื่อเมือง อุณหภูมิ ความกดอากาศ ส่วน HTML เป็นการกำหนด Tag ต่างๆ ที่จะกำหนดรูปแบบการแสดงผลของข้อมูลบนหน้าเว็บ ซึ่งข้อมูลจะสามารถแสดงออกมาได้หลายรูปแบบขึ้นอยู่กับข้อกำหนดของ HTML

3.2.1 ลักษณะโครงสร้างของ XML

XML เป็นการใช้ข้อความเพื่อบ่งบอกโครงสร้างของเอกสาร พิจารณาตัวอย่างรูปแบบโครงสร้างของหนังสือ เมื่อหนังสือประกอบด้วยจำนวนบท 2 บท ในแต่ละบทประกอบด้วยเนื้อความ

Begin Book

Begin Chapter 1

Text for Chapter 1

End Chapter 1

Begin Chapter 2

Text for Chapter 2

End Chapter 2

End Book

หนังสือที่มีอยู่ในปัจจุบันจะมีโครงสร้างที่มีรายละเอียดที่ซับซ้อนมากกว่านี้ เช่น บทนำ, สารบัญ เป็นต้น เช่นเดียวกัน ภายในส่วนเนื้อความยังประกอบด้วยโครงสร้างย่อยคือ ย่อหน้า (Paragraph) แต่ละย่อหน้ายังประกอบขึ้นจากประโยค คำและตัวอักษรด้วยลักษณะของเอกสาร XML นั้น สามารถอธิบายโดยใช้ตัวอย่างที่ 1 ได้ ดังนี้

ตัวอย่างที่ 1

```
<?xml version="1.0" encoding="windows-874"?>
```

```
<note>
```

```
  <to>John</to>
```

```
  <from>Cat</from>
```

```
  <heading>I Nice to meet You</heading>
```

```
  <body>By</body>
```

```
</note>
```

บรรทัดที่ 1 นั้นหมายความว่าเราประกาศเอกสารนี้เป็นเอกสาร XML เวอร์ชันที่ 1.0 และมีการเข้ารหัสอักขระเป็น windows-874 เพื่อให้ใช้ภาษาไทยได้ บรรทัดที่ 2 คือ Element ตัวแรกหรือตัวแม่ของเอกสาร หรือที่เรียกว่า root element ที่ทุกๆ เอกสาร XML ต้องมีโดยที่เอกสารหนึ่งเอกสารมีได้เพียง root element เดียวเท่านั้น บรรทัดที่ 3 ถึง 6 มี element 4 ตัวคือ to, from, heading และ body ซึ่งเป็น element ลูกของ element note ส่วนบรรทัดสุดท้ายก็คือแท็กปิดของ element แม้นั่นเองและทุกๆ element ในเอกสาร XML ต้องมีทั้งแท็กเปิดและแท็กปิด

3.2.1.1 Tag

Tag ใน XML มีความหมายในลักษณะเดียวกับที่ใช้ใน HTML Tag คือข้อความที่อยู่ระหว่างสัญลักษณ์ < และ > ส่วนชนิดของ tag จะมีทั้ง tag เปิดและ tag ปิดแต่ก็จะมีที่เป็นแท็กเปิดและแท็กปิดอยู่ในตัวเดียวกันด้วยเช่น <ชื่อแท็ก /> ซึ่งแท็กเปิดเป็นตัวบอกว่าเริ่ม element

และแท็กปิดเป็นตัวบอกว่าจบ element และความแตกต่างระหว่างแท็กเปิดกับแท็กปิดคือแท็กเปิดสามารถใส่ข้อมูลอธิบายเพิ่มได้ซึ่งเรียกว่า attribute แต่แท็กปิดไม่สามารถทำได้

3.2.1.2 Element

Element เป็นส่วนขยายอธิบายความหมายเพิ่มได้อีกและมีความสัมพันธ์ แบบ element แม่กับ element ลูกใน element สามารถใส่ข้อมูล เข้าไปได้ โดยข้อมูล ที่ใส่เข้าไประหว่าง แท็กเปิดกับแท็กปิดเรียกว่า Content และข้อมูลที่ใส่เพิ่มเข้าไปในแท็กเปิดเพื่ออธิบายคุณสมบัติลักษณะของ element เพิ่มเรียกว่า Attribute ทุก element ในเอกสารต้องมีทั้งแท็กเปิดและแท็กปิด

3.2.1.3 Attribute

Element ในเอกสาร XML สามารถมี Attribute ในแท็กเปิดได้และมีได้มากกว่า 1 ตัว เหมือนกับในเอกสาร HTML โดยที่ Attribute จะอธิบายถึงคุณสมบัติหรือบอกลักษณะของ element นั้นเพิ่ม คือถ้ากล่าวถึงสิ่งของหากเราไม่ได้ให้ความหมายเพิ่มเติมเราก็จะไม่มีทางรู้หรือแยกแยะสิ่งที คล้ายๆ กันออกจากกันได้ เช่น ถ้าพูดถึง บท ในหนังสือ แต่นี่เป็นความหมายโดยรวม แต่ถ้าบอกว่า บทที่ 1 ในหนังสือ เลข 1 ในที่นี้คือความหมายเพิ่มเติมให้กับบท

3.2.1.4 Content

เนื้อความหรือ Content ถือได้ว่าเป็นข้อมูลเพื่อใช้ในการแสดงให้ผู้อ่านเอกสารได้เห็น หรือ กล่าวอีกนัยหนึ่งคือ Content อยู่หลัง Tag เปิด และจบก่อนที่จะถึง Tag ปิดนั่นเอง

3.2.2 ตัวกำหนดโครงสร้างของ XML ด้วย DTD (Document Type Declaration)

Document Type Declaration ใช้ในการกำหนดโครงสร้างของเอกสาร XML โดยเมื่อเปิดเอกสาร XML ด้วย DTD ตัวประมวลผลจะตรวจสอบเอกสารว่าถูกต้องตรงกับ DTD ที่ได้กำหนดโครงสร้างไว้หรือไม่ เช่น ถ้าไม่ประกาศ Element หรือ Attribute ไว้ใน DTD แล้วจะไม่สามารถใช้ Element หรือ Attribute นั้นๆ ในเอกสาร XML ได้

3.2.2.1 รูปแบบของ DTD

DTD มีรูปแบบทั่วไปดังนี้

<! DOCTYPE Name DTD>

Name ตามตัวอย่างเป็นที่กำหนดชื่อของ Document Element (หรือ Root Element) ชื่อของ Document Element จะต้องตรงกับชื่อที่กำหนดไว้ที่นั่นเท่านั้น

DTD สามารถบรรจุชนิดของการประกาศ Markup ไว้ดังนี้

การประกาศ Element Type

โดยมีรูปแบบการประกาศ Element Type ดังนี้

<! ELEMENT ชื่อelement (รายละเอียดในelement)>

รายละเอียดในElementเป็นที่กำหนดว่าภายใน Element สามารถบรรจุอะไรได้บ้างด้านล่างนี้จะเป็นตัวอย่างของเอกสาร XML ที่มี Element Type 2 ชนิด และ ElementType ที่ประกาศชื่อ COLLECTION แสดงให้เห็นว่าสามารถบรรจุ Element ที่ชื่อ CD ได้ มากกว่า 1 Element และ Element Type ที่ชื่อ CD กำหนดว่าสามารถบรรจุได้เฉพาะ

ข้อมูลประเภท Character Data เท่านั้น

```
<? Xml version="1.0"?>
```

```
<! DOCTYPE COLLECTION
```

```
[
```

```
<! ELEMENT COLLECTION (CD) +>
```

```
<! ELEMENT CD (#PCDATA)>
```

```
]
```

```
>
```

```
<COLLECTION>
```

```
<CD>Mozart Violin Concertos 1, 2, and 3</CD>
```

```
<CD>Telemann Trumpet Concertos</CD>
```

```
<CD>Handel Concerti Grossi</CD>
```

```
</COLLECTION>
```

การประกาศ Attribute

โดยมีรูปแบบของการประกาศดังนี้

```
<!ATTLIST ชื่อของelement ชื่อของแอตทริบิวต์ ชนิดของแอตทริบิวต์ ค่าดี
```

```
ฟอลต์>
```

ชนิดของแอตทริบิวต์มีได้หลายอย่าง ดังตารางที่ 3-1

ชนิดของแอตทริบิวต์	คำอธิบาย
CDATA	ค่าที่เป็นข้อมูลแบบ Character Data (ไม่กระจายคำในพาสเซอร์)
(eval eval ...)	ค่านี้จะเป็นกลุ่มของค่าที่เตรียมไว้ให้เลือก
ID	ค่านี้เป็นเลขประจำตัวที่ไม่ซ้ำกับใคร

IDREF	ค่านี้เป็นเลขประจำตัวของelementหนึ่ง
IDREFS	ค่านี้เป็นลิสต์ของเลขประจำตัวอื่นๆ
NMTOKEN	ค่านี้เป็นชื่อที่ถูกต้องตามหลัก XML
NMTOKENS	ค่านี้เป็นลิสต์ของชื่อที่ถูกต้องตามหลัก XML
ENTITY	ค่านี้เป็นเอนติตี
ENTITIES	ค่านี้เป็นลิสต์ของเอนติตี
NOTATION	ค่านี้เป็นชื่อของ Notation ที่ได้ประกาศไว้
Xml:	ค่านี้เป็นค่าที่มีการกำหนดไว้ก่อนแล้ว

ตารางที่ 3-1 แสดงชนิดของแอตทริบิวต์ที่สามารถใช้ในการประกาศแอตทริบิวต์
ส่วนที่เป็นค่าดีฟอลต์สามารถกำหนดได้ดังตารางที่ 3-2

ค่าที่กำหนด	คำอธิบาย
#DEFAULT value	แอตทริบิวต์นี้มีค่าดีฟอลต์และระบุมาให้
#REQUIRED	ต้องมีค่าแอตทริบิวต์ให้กับelement
#IMPLIED	ค่าแอตทริบิวต์อาจไม่จำเป็นต้องให้ไว้ก็ได้
#FIXED value	ค่าของแอตทริบิวต์เป็นค่าที่กำหนดไว้ตายตัว ผู้ใช้เปลี่ยนไม่ได้

ตารางที่ 3-2 แสดงค่าที่สามารถได้ในส่วนที่เป็นค่าดีฟอลต์ในการประกาศแอตทริบิวต์

3.2.3 การแปลงรูปแบบเอกสาร XML (Transforming XML)

XSL ซึ่งย่อมาจาก Extensible Style sheet Language (XSL) เป็นภาษาที่ใช้ในการแปลงจากรูปแบบ (Transformation) ของเอกสาร XML ไปอยู่ในรูปแบบอื่นๆ และในขณะเดียวกันก็ทำการประยุกต์

รูปแบบการจัดข้อความ (Style) ด้วย XSL style sheet (สไตล์ชีต) ถูกเขียนในรูปแบบที่เหมาะสมกับ XML แต่ต้องทำการกำหนดลำดับของ Element ไว้ล่วงหน้าในการกำหนดสิ่งที่ต้องการให้กระทำในการแปลงรูปแบบโดยความหมายจะถูกกำหนดโดยตัวประมวลผล XSL เอง ซึ่งควรจะเป็นไปตามมาตรฐานของ W3C ดังนั้นจาก XSL ซึ่งเป็นแอปพลิเคชันหนึ่งของ XML และอยู่ในรูปแบบที่เหมาะสมกับ XML ดังนั้นจึงสามารถประมวลผล XSL สไตล์ชีตได้เหมือนกับเอกสาร XML ครั้งแรก XSL เป็นข้อเสนอที่ถูกระบุทั้งทางที่การแปลงรูปแบบ XML และไวยากรณ์ในการจัดการรูปแบบบนแพลตฟอร์มที่เป็นอิสระกัน (platform-independent styling grammar) แต่อย่างไรก็ตามในส่วนของการแปลงรูปแบบก้าวหน้าอย่างรวดเร็วและเป็นประโยชน์อย่างมาก ขณะที่ไม่มีแอปพลิเคชันในส่วนของการจัดรูปแบบมากนักและส่วนใหญ่จะกล่าวว่ามีประโยชน์ทางด้านการทำนอย

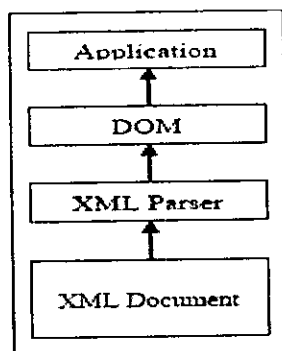
ดังนั้นจากข้อเสนอในตอนแรกจึงสามารถแบ่งออกได้เป็น 2 ส่วนที่แตกต่างกันคือ ส่วนในการแปลงรูปแบบเอกสาร ซึ่งปัจจุบันถูกเรียกว่า XSLT (XSL Transformation Language) แต่ในที่นี้จะพิจารณาเฉพาะในส่วนของการแปลงรูปแบบเท่านั้นเพื่อแปลงรูปแบบจากเอกสาร XML แบบหนึ่งไปเป็น XML อีกแบบหนึ่ง เช่น ใช้ในการแปลงเอกสาร GML ไปเป็น SVG

3.2.3.1 The XSL Transformation Language (XSLT)

XSLT ถูกกำหนดโดยองค์กร W3C ในการกำหนดชุด Element ของ XML ที่สามารถจะถูกใช้ในการสร้างสไลด์ชุดที่แปลงเอกสาร XML ให้อยู่ในรูปแบบใดๆ ซึ่งในปัจจุบันนี้ใช้ในการแปลง XML รูปแบบหนึ่งไปเป็นอีกรูปแบบหนึ่ง โดยการทำงานจะทำโดยนำเอกสาร XML ที่เหมาะสมกับ XML นั้น ไปทำการแปลงรูปแบบ เพื่อสร้างเอกสารเอาต์พุต ซึ่งขึ้นอยู่กับคำสั่งและเนื้อหาภายในสไลด์ชุดจึงสามารถสร้างรูปแบบเอาต์พุตใดๆ ก็ได้ตามที่ต้องการในการทำงาน

3.2.4 DOM (Document Object Model)

Document object model คือ เป็นวิธีที่ประมวลโครงสร้างของเอกสาร XML ให้เป็นโครงสร้างแบบต้นไม้ เพื่อให้แอปพลิเคชันสามารถเข้าหาจุดต่าง ๆ ของโครงสร้างต้นไม้ได้ โดยที่ DOM จะโหลดไฟล์ XML ทั้งไฟล์ใน Memory ซึ่งก็มีทั้งข้อดีและข้อเสีย คือ ข้อดี คือ ครั้งแรกสุดที่มีการเรียกใช้งานจะช้า แต่หลังจากนั้นการเข้าถึงจุดต่าง ๆ ของไฟล์ จะเร็วเพราะถูกเก็บใน memory แล้ว ข้อเสีย คือ ไม่เหมาะกับ file XML ที่มีขนาดใหญ่ โดยที่ขอบเขตจะถูกจำกัดด้วยหน่วยความจำที่เรียกใช้ในขณะนั้น เราสามารถที่จะทำอะไรกับข้อมูลในแอปพลิเคชันที่เราเขียนขึ้นได้ ไม่ใช่เพียงการเข้าถึงเท่านั้นแต่ยังรวมถึงการแก้ไขและเพิ่มเติมเอกสาร XML ที่เรามีอยู่อีกด้วย จึงเกิดมี Document Object Model (DOM) ขึ้นเพื่อช่วยในการทำงานกับเอกสาร XML ทั้งหมด DOM มักจะถูกใส่เข้าไปเป็นชั้นที่คั่นกลางระหว่าง XML parser กับแอปพลิเคชันที่ต้องการใช้ข้อมูลในเอกสาร XML ซึ่งหมายความว่า parser จะอ่านข้อมูลจากเอกสาร XML ให้กับ DOM จากนั้น DOM จะถูกใช้โดยแอปพลิเคชันระดับสูงกว่า แอปพลิเคชันสามารถทำงานกับเอกสาร XML ได้ตามที่ต้องการ รวมถึงการใส่ DOM เข้าไปใน DOM ตัวอื่น ซึ่งในแต่ละภาษาก็จะมีแบบจำลอง DOM เป็นของตนเอง



รูปที่ 3.7 การใช้งาน DOM

วิธีการนำ Document object model ไปใช้งานจะนำไปใช้กับข้อมูลที่อยู่ในรูปแบบของ XML file หรือ XML String เราสามารถที่จะเข้าถึงข้อมูลได้อย่างง่ายดายและสามารถที่อ่านข้อมูลนั้นออกมาได้ดังนี้

1. สร้าง Document object ขึ้นมาโดยที่ต้องตรวจสอบว่าผู้ใช้ กำลังใช้ Browser ชนิดไหนอยู่ เช่น `xmlDocument = new ActiveXObject('Microsoft.XMLDOM');` ถ้าใช้ IE หรือ `xmlDocument = document.implementation.createDocument("", 'doc', null);` ถ้าใช้ FireFox

2. นำเอา Xml File หรือ Xml string ให้กับ object ที่สร้างขึ้น

`xmlDocument.loadXML(xmlHttp.responseText);`

3. ไปยัง Element ที่ต้องการ โดยที่เราต้องรู้ชื่อ Element เช่น `Name = xmlDocument.getElementsByTagName('name');`

4. เราสามารถที่ดึงข้อมูลที่อยู่ใน Element นี้ได้โดย `data = name[0].firstChild.data;`

3.2.5 RSS (Really Simple Syndication)

RSS เป็นบริการใหม่บนเว็บไซต์ภาษา XML ใช้สำหรับดึงข่าวจากเว็บต่างๆ มาแสดงบนหน้าเว็บเพจ โดยนำมาเฉพาะหัวข้อข่าว เมื่อผู้ใช้คลิกลิงก์ก็จะแสดงรายละเอียดข่าวในเว็บต้นฉบับนั้นๆ โดยที่หัวข้อข่าวจะอัปเดตตามเว็บต้นทาง ซึ่งการดึงหัวข้อข่าวไปแสดงนั้นจะมีส่วนประกอบทั้งหมดสามส่วนคือส่วนผู้ให้บริการดึงข่าว และส่วนผู้สร้างเว็บไซต์ใช้ทั่วไปที่ต้องการดึงข่าวไปแสดง และส่วนผู้ใช้ทั่วไป

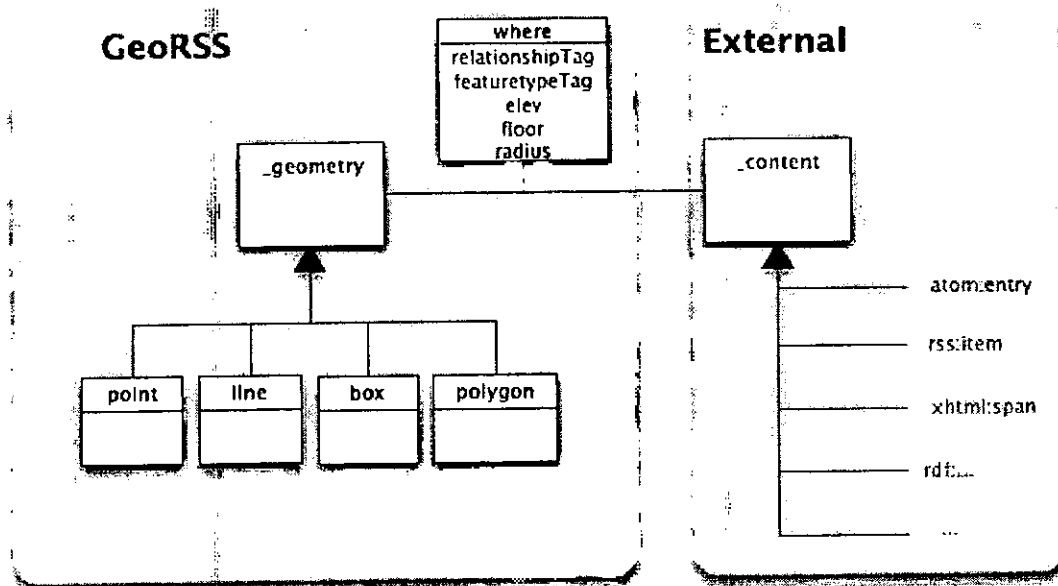
RSS ช่วยลดข้อจำกัดในการคัดลอกข้อมูลในเว็บไซต์โดยเฉพาะกรณีการละเมิดลิขสิทธิ์ ขณะที่ผู้สร้างไม่ต้องเสียเวลาทำหน้าเพจแสดงข่าว ซึ่งต้องทำทุกครั้งเมื่อต้องการเพิ่มข่าว โดย RSS จะดึงข่าวมาอัตโนมัติ ทำให้ข้อมูลบนเว็บไซต์เป็นศูนย์กลางมากขึ้น

ปัจจุบัน RSS ถูกนำมาประยุกต์ใช้เป็นรูปแบบกลางในการบริการข้อมูลทางธุรกิจ และมีการแข่งขันกันสูง โดยเฉพาะธุรกิจที่มีการแชร์ข้อมูล เช่นเว็บไซต์ข่าว เว็บบล็อก ซึ่งจะมีการแสดงข้อมูลบนหน้าต่างฟรีวิวแยกต่างหากเพื่อให้ผู้ใช้ไม่สับสน รวมถึงสามารถสืบค้นข้อมูลได้

จุดเด่นของ RSS คือผู้ใช้จะไม่จำเป็นต้องเข้าไปตามเว็บไซต์ต่างๆ เพื่อความมีข้อมูลอัปเดตใหม่หรือไม่ ขณะที่เว็บไซต์แต่ละแห่งอาจมีระยะเวลาในการอัปเดตไม่เท่ากัน บางครั้งผู้ใช้อาจหลงลืมจนเข้าไปดูเนื้อหาอัปเดตใหม่บนเว็บไม่ครบถ้วน รูปแบบ RSS จะช่วยให้ผู้สามารถรับข่าวสารอัปเดตใหม่ได้โดยไม่ต้องเข้าไปดูทุกครั้งให้เสียเวลา ได้ประโยชน์ทั้งฝ่ายผู้บริโภคและฝ่ายเจ้าของเว็บไซต์

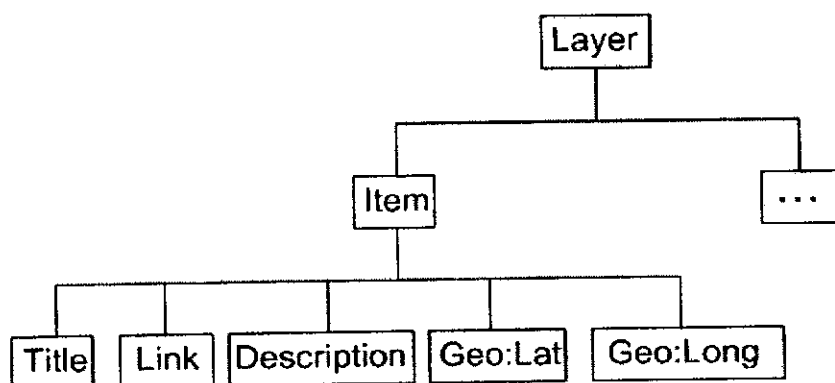
3.2.6 GeoRSS (Geographically Encoded Objects for RSS feeds)

GeoRSS คือรูปแบบหนึ่งของ RSS feed ที่อธิบายจุดต่างหรือสิ่งที่เป็นภูมิศาสตร์ซึ่งทาง W3C จัดทำขึ้นเพื่อให้เป็นมาตรฐาน



รูปที่ 3.8 GeoRSS Model

ซึ่งในเอกสารนี้เราได้นำเอารูปแบบของ GeoRSS ในรูปแบบของ point มาใช้งานและรูปแบบของ GeoRSS มีลักษณะดังนี้



รูปที่ 3.9 GeoRSS ในรูปแบบของ Point

บทที่ 4

Virtual Earth Local Live

ชุดพัฒนาซอฟต์แวร์ตัวควบคุมแผนที่ Virtual Earth รุ่นที่ 4.0 สนับสนุนให้นักพัฒนา web application สามารถสร้างแผนที่ขึ้นใช้งานบนอินเทอร์เน็ตในลักษณะคล้ายคลึงกับ Windows Live Local ซึ่งเป็นบริการแผนที่บนอินเทอร์เน็ต ที่ผู้ใช้บริการสามารถใช้สืบค้น เสาะแสวงหา วางแผน และแชร์ข้อมูลเกี่ยวกับ สถานที่ต่างๆบนแผนที่ชุดพัฒนาฯ และบริการสนับสนุนที่จัดให้มา องค์ประกอบสำคัญดังนี้ แผนที่เส้นทาง (road maps) ภาพถ่ายทางอากาศพร้อมข้อความกำกับ ภาพถ่ายทางอากาศละเอียดสูงแบบมุมมองต่ำ และ ความสามารถในการค้นหาข้อมูลจาก GeoRSS หรือข้อมูลที่แชร์กันได้ในกลุ่มผู้ใช้ Windows Local Live ขึ้นแสดงบนแผนที่เป็น สัญลักษณ์หมุดพร้อมข้อมูลกำกับ การสืบค้นข้อมูลตำแหน่งข้างเคียงการหาและบอกเส้นทางการขับรถ เป็นต้น ดังนั้น Windows Local Live จึงได้ทำแผนที่ online ขึ้นมาเพื่อให้ผู้ใช้งานสามารถที่จะนำมาใช้กับงานของตัวเองได้ องค์ประกอบดังกล่าวเป็นส่วนสนับสนุน ที่นักพัฒนาซอฟต์แวร์สามารถใช้สร้างเป็น web application ที่มีคุณสมบัติเฉพาะตัวที่ผนวกไว้ด้วยข้อมูลแผนที่ และขีดความสามารถในการสืบค้นเชิงตำแหน่ง เพื่อให้บริการแก่ผู้สนใจบนอินเทอร์เน็ต ในรูปแบบที่นักพัฒนาเป็นผู้กำหนดขึ้นเอง

ชุดพัฒนาซอฟต์แวร์ตัวควบคุมแผนที่ Virtual Earth รุ่นที่ 4.0 ของไมโครซอฟท์นี้เป็นซอฟต์แวร์ประกอบขึ้นจาก API (application programming interface) ที่จัดทำขึ้นสำหรับใช้ในการพัฒนางานแผนที่ประยุกต์บนอินเทอร์เน็ต และทรัพยากรส่วนหนึ่งที่ใช้กับ Virtual Earth ซึ่งเป็นบริการหนึ่งของไมโครซอฟท์ที่มีมาก่อนแล้ว โดยที่ Virtual Earth และ Virtual Earth Platform เป็นบริการที่พัฒนามาจาก Microsoft MapPoint Web Service ในส่วนของรุ่นที่ 4.0 สามารถที่จะใช้ตัวแผนที่ 3D และ 2D ซึ่งในแผนที่รุ่นที่ 3.0 นั้นมีลักษณะเป็น 2D อย่างเดียวสำหรับการใช้งานแผนที่ภายในขอบเขตประเทศไทย ทรัพยากรที่ทาง Virtual Earth จัดให้นับว่าน่าสนใจอย่างยิ่งในส่วน of แผนที่ เพราะมีรายละเอียดในระดับที่สามารถใช้งานได้กับถนนสายหลักทั่วประเทศ มีตำแหน่งที่ตั้งของชุมชนสำคัญถึงระดับอำเภอและตำบล ซึ่งข้อมูลเหล่านี้สนับสนุนการวิเคราะห์หาเส้นทางระหว่างจุดสองจุดในเรื่องการค้นหาเส้นทางขับรถที่ดีที่สุดได้อย่างน่าสนใจ ส่วนภาพถ่ายดาวเทียมที่มีให้บริการยังถือว่ายังไม่ดีมากนัก แต่ก็มีสิ่งชัดเจนที่บริการของภาพแบบนกเห็น (Bird's Eye Image)

4.1 ความเป็นมาของ Virtual Earth Local Live

Virtual Earth Local live คือมาจากบริษัทผลิตซอฟต์แวร์ที่ชื่อว่า Microsoft Corporation

4.2 ความรู้พื้นฐานที่จำเป็นของผู้ใช้ชุดพัฒนาฯ

ในการพัฒนางานแผนที่ประยุกต์เบื้องต้นด้วยชุดพัฒนาซอฟต์แวร์ตัวควบคุมแผนที่ Virtual Earth จำเป็นต้องใช้ความรู้พื้นฐานหลายด้าน ดังนี้

4.2.1 HTML, XHTML (Hypertext Markup Language, Extensible Hypertext Markup Language)

4.2.2 JavaScript

4.2.3 CSS (Cascading Style Sheet)

4.2.4 DOM (Document Object Model)

4.2.5 XML (Extensible Markup Language)

4.3 การใช้งาน Virtual Earth Map

การใช้งานแผนที่นั้นเราต้องใช้ภาษา Java script เป็นตัวควบคุมแผนที่ซึ่งเราจะสามารถใช้งานผ่าน Class ต่างๆดังนี้

4.3.1 VEMap Class คือ Class ที่ใช้สร้างแผนที่และควบคุมแผนที่ VEMap Class จะมี Method ดังดังนี้

- Constructor Method มีหนึ่งตัวคือ `var map = new VEMap( control id);` เป็นการสร้าง object แผนที่ขึ้นมาเพื่อนำมาใช้งานต่อและเป็นตัวอ้างอิงถึงเวลาใช้งาน เช่นตัวแปรชื่อ map คือตัวแปรที่เก็บ object แผนที่อยู่ และ control id คือ id ของ tag div ในภาษา html ที่เราต้องการที่จะแสดงแผนที่ในพื้นที่นั้น

- LoadMap Method เป็นตัวทำให้แผนที่แสดงตาม control id และเราต้องส่ง parameter คือ จุดศูนย์กลางเริ่มต้นกำหนดได้จาก object VELatLong, ระดับที่ zoom จาก 1-19, รูปแบบแผนที่ (h=hybrid, o=oblique, r=road และ r คือค่า default), ค่านี้เป็นชนิด Boolean ถ้า true

- AddLayer Method คือการเพิ่มข้อมูลเข้าไปในแผนที่ทีละหลายๆตัวเป็นชุดเดียวกัน Parameter นั้นต้องเป็น object ที่มาจาก class VELayerSpecification

- AddPolyline Method คือการลากเส้นตรงเราสามารถที่จะกำหนดทีละหลายๆจุดให้ลากเส้นไปที่เดียวได้โดย Parameter นั้นต้องเป็น object ที่มาจาก class VEPolyline

- AddPushpin Method คือการเพิ่มจุดที่เป็นรูปภาพเข้าไปทีละจุดซึ่ง Parameter นั้นต้องเป็น object ของ class VEPushpin

- AttachEvent Method คือการกำหนด Event ต่างๆที่เกิดขึ้นกับตัวแผนที่
- Clear Method คือการเอา object ที่เพิ่มเข้าไปออกมา เช่น pushpins, routes, layer and results
- DeleteAllPolylines Method คือการเอา object VEPolyline ที่เพิ่มเข้าไปออกไปทั้งหมด
- DeleteAllPushpins Method คือการเอา object VEPushpin ที่เพิ่มเข้าไปออกไปทั้งหมด
- DeleteLayer Method คือการเอา Layer ที่เพิ่มเข้าไปในแผนที่ออกตาม ID Layer
- DeletePolyline Method คือการเอา object VEPolyline ที่เพิ่มเข้าไปออกไปตาม id ที่ตั้งให้กับเส้นนั้น

- DeletePushpin Method คือการเอา object VEPushpin ที่เพิ่มเข้าไปออกไปตาม id ที่ตั้งให้กับจุดนั้น

- DetachEvent Method คือการยกเลิก Event ที่กำหนดค่าไว้
- GetCenter Method คือการเอาค่าจุดตรงกลางของแผนที่ขณะที่แสดงอยู่
- GetLeft Method คือการเอาค่าจุดซ้ายสุดของแผนที่ขณะที่แสดงอยู่
- GetMapStyle Method คือการเอารูปแบบที่แสดงอยู่มาใช้รูปแบบไหนอยู่ขณะนี้
- GetZoomLevel Method คือการเอาค่า Zoom Level ว่าขณะนี้อยู่ Level อะไร
- HideLayer Method คือการสั่งให้ซ่อน Layer ตาม ID Layer ที่กำหนด
- LatLongToPixel Method คือการแปลงค่าจาก Latitude Longitude ไปเป็นค่า Pixel
- PixelToLatLong Method คือการแปลงค่าจาก Pixel ไปเป็นค่า Latitude Longitude
- SetCenter Method คือการกำหนดค่าให้แผนที่ไปแสดงตามจุดที่กำหนด
- SetCenterAndZoom Method คือการกำหนดค่าให้แผนที่ไปแสดงตามจุดที่กำหนดและ

กำหนดระยะ Zoom

- SetMapStyle Method คือการกำหนดรูปแบบให้แผนที่ว่าจะเป็นแบบไหนมีอยู่ 4 แบบคือ (r,a,h,o)

- SetZoomLevel Method คือการกำหนดระยะ Zoom มีช่วงตั้งแต่ 1-19
- ShowLayer Method คือการสั่งให้ Layer ตาม ID Layer ที่กำหนดแสดงออกมา
- ShowMessage Method คือการแสดง Message Box บนแผนที่
- ZoomIn Method คือการเพิ่มระยะ Zoom ขึ้นหนึ่ง
- ZoomOut Method คือการลดระยะ Zoom ลงหนึ่ง

4.3.2 VEMap Events

- onchangemapstyle Event กำหนดให้ไปทำงานที่ function เมื่อมีการเปลี่ยนรูปแบบแผนที่
- onchangeview Event กำหนดให้ไปทำงานที่ function เมื่อมีการเปลี่ยนรูปแบบ view
- onclick Event กำหนดให้ไปทำงานที่ function เมื่อมีการคลิกซ้ายบนแผนที่
- oncontextmenu Event กำหนดให้ไปทำงานที่ function เมื่อมีการคลิกขวาบนแผนที่

- onendzoom Event กำหนดให้ไปทำงานที่ function เมื่อ Zoom ไปถึงระยะสุดท้าย
- onerror Event กำหนดให้ไปทำงานที่ function เมื่อเกิด error บนแผนที่
- onmouseup Event กำหนดให้ไปทำงานที่ function เมื่อมีการคลิกซ้ายบนแผนที่แล้วปล่อย
- onresize Event กำหนดให้ไปทำงานที่ function เมื่อเปลี่ยนขนาดแผนที่
- onstartzoom Event กำหนดให้ไปทำงานที่ function เมื่อ Zoom ไปถึงระยะที่หนึ่ง

4.3.3 VELatLong Class เป็น data แบบหนึ่งที่ใช้เป็นตัวกำหนดจุดบนแผนที่

- Constructor Method มีรูปแบบดังนี้ `var point = new VELatLong(13.1,100.2);`

ส่วนประกอบของ class นี้คือตัวที่หนึ่งคือ Latitude ตัวที่สองคือ Longitude

4.3.4 VECOLOR Class เป็น data แบบหนึ่งที่ใช้กำหนดสี

- Constructor Method มีรูปแบบดังนี้ `var color = new VECOLOR(255,0,0,0.4);`

ส่วนประกอบของ class นี้คือ ค่าสีในรูปแบบ RGB ตัวที่ 1 สีแดง ตัวที่ 2 สีเขียว ตัวที่ 3 น้ำเงิน ตัวที่ 4 คือค่า alpha ซึ่งมีช่วง 0.0 – 1.0

4.3.5 VEPolyline Class เป็นตัวกำหนดเส้นบนแผนที่

- Constructor Method มีรูปแบบดังนี้ `var line = new VEPolyline(id,point);` id คือเบอร์ของเส้นต้องไม่ซ้ำกัน และ point คือจุดที่กำหนดให้ลากเส้นไปตามจุด

- SetWidth Method คือการกำหนดขนาดเส้น
- SetColor Method คือการกำหนดสีให้กับเส้น

4.3.6 VEPushpin Class เป็นตัวกำหนดจุดบนแผนที่

- Constructor Method มีรูปแบบดังนี้ `new VEPushpin(id, location, icon_url, title, details, iconStyle, titleStyle, detailsStyle);`

id คือการกำหนดเบอร์ของจุดต้องไม่ซ้ำกัน

location คือตำแหน่งของจุดต้องใช้ VELatLong Class ในการกำหนดจุด

icon_url คือการกำหนดรูปของจุดตามที่อยู่รูปนั้นว่าอยู่ที่ไหน

title คือการใส่ข้อความให้กับจุดในส่วนของหัวของข้อความ

details คือการใส่รายละเอียดให้กับจุดเป็นส่วนเพิ่มรายละเอียดให้มากขึ้น

iconStyle ,titleStyle, detailsStyle คือการกำหนดรูปแบบตาม style sheet

4.3.7 VELayerSpecification Class คือการกำหนดค่าจุดหลายจุดใน Layer เดียวกันรูปในการกำหนดค่าจุดต่างเราต้องกำหนดตาม GeoRSS XML หรือ Windows Live Local collection

- Constructor Method มีรูปแบบดังนี้ `new VELayerSpecification(layerType, id, url, method, callback, iconUrl);`

layerType คือการกำหนดรูปแบบของข้อมูลที่จะให้กับ Layer ซึ่งมีสองแบบคือ VELayerType { GeoRSS ,VECollection }

id คือการกำหนดเบอร์ของ Layer ต้องไม่ซ้ำกัน

url คือที่อยู่ของข้อมูลที่ให้กับ Layer method ถ้า layerType คือ GeoRSS ต้องกำหนดค่า HTTP ว่าเป็น GET หรือ POST

callback คือการกำหนด function เมื่อเพิ่ม Layer เข้าไปในแผนที่เสร็จแล้วจะกลับมาทำงานที่ function

iconUrl คือการกำหนดรูปให้กับจุดต่างๆของ Layer

4.4 การเริ่มต้นใช้งาน Virtual Earth map control SDK

Virtual Earth map control SDK คือ ชุดพัฒนาซอฟต์แวร์แบบ API ในภาษา JavaScript ซึ่งมีส่วนประกอบของ objects, method, และ event ที่ร่วมกันทำหน้าที่ประมวลผลเพื่อแสดงแผนที่ที่โต้ตอบกับผู้ใช้ได้บนเว็บเพจ โดยใช้ map control ตามแบบของ Virtual Earth การที่ Virtual Earth map control SDK เป็นภาษาจาวาสคริปต์ทำงานในเว็บเพจทางฝั่ง client โดยมีเว็บเบราว์เซอร์ที่ใช้อยู่ตามปกติเป็นหลักในการประมวลผล ทำให้การใช้งานของแอปพลิเคชัน ที่สร้างขึ้นเพื่อใช้งานบนอินเทอร์เน็ต เผยแพร่ออกไปอย่างกว้างขวาง การประมวลผลไม่เป็นการมากกับเซิร์ฟเวอร์ของผู้ให้บริการข้อมูลแผนที่ (Virtual Earth) เป็นบริการที่จัดให้ฟรีถ้าไม่ทำการค้า สิ่งเหล่านี้ส่งผลให้มีผู้สนใจที่จะเรียนรู้การใช้งาน Virtual Earth map control SDK กันมาก เนื้อหาในแต่ละหัวข้อในบทความนี้จะช่วยให้ผู้สนใจได้เริ่มต้นเรียนรู้เพื่อนำ map control ไปใช้งานแบบทีละขั้นตอน

4.4.1 วิธีการสร้างแผนที่ในเว็บของเรา

1. ในส่วนของ Header เช่นกัน ให้เขียน script element เพื่อทำหน้าที่เรียกใช้โปรแกรม map control และทรัพยากรสนับสนุนตามแบบอย่างดังนี้

```
<script src="http://dev.virtualearth.net/mapcontrol/v4/mapcontrol.js"></script>
```

2. ในส่วน body ของเว็บเพจ ให้เขียน div element เพื่อทำหน้าที่เป็น container รองรับแผนที่ ขนาดของแผนที่กำหนดได้ด้วยค่า height และ width ของ div element นี้ ส่วนตำแหน่งจะกำหนดได้ด้วย "position", "top" และ "left" เราสามารถกำหนดรายละเอียดเหล่านี้แบบ inline (แบบที่ 1) หรือกำหนดเป็น style class ขึ้นก่อนใน style element แล้วเรียกใช้ภายหลังด้วยการระบุเป็น class attribute (แบบที่ 2) ก็ได้ ดังนี้

แบบที่ 1 <div id='myMap' style="position: absolute; width: 400px; height: 400px;"></div>

แบบที่ 2 .firstmap {position: absolute; top: 20; left: 10; width: 400px; height: 400px; border: #555555 2px solid ;}

```
<div id="myMap" class="firstmap"></div>
```

3. สร้างส่วนของ function Load map ตามรูปแบบดังนี้

```

var map;

Function getmap(){
    map = new VEMap('myMap');
    map.LoadMap();
}

```

4 กำหนดให้ Tag body ในขณะที่โหลด ไปทำงานที่ function สร้างแผนที่ดังนี้

```
<body onclick= "getmap();" >
```

ตัวอย่างในการสร้างแผนที่บนเว็บ

```
<html>
```

```
<head>
```

```
<title></title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

```
<script src="http://dev.virtualearth.net/mapcontrol/v4/mapcontrol.js"></script>
```

```
<script>
```

```
var map = null;
```

```
function getmap(){ map = new VEMap('myMap'); map.LoadMap(); }
```

```
}
```

```
</script>
```

```
</head>
```

```
<<body onclick= "getmap();" >
```

```
<div id='myMap' style="position: absolute; width: 400px; height: 400px;" ></div>
```

```
</body>
```

```
</html>
```

การ load map ให้แสดงตรงตำแหน่งทางภูมิศาสตร์ (latitude, longitude) ที่ต้องการ รวมถึงการจัดสถานะของระดับการย่อแผนที่และรูปแบบ (map style) ในคราวเดียวกัน สามารถทำได้โดยใช้ LoadMap function โดยป้อนค่าที่เหมาะสมของ location, zoom level, map style ให้แปลงให้เป็นรูปแบบดังนี้

```
map.LoadMap(new VELatLong(13.8, 100.1), 7, 'r', false);
```

4.4.2 การใช้หมุดบนแผนที่ (Using Pushpins on Map)

1. สร้าง VEPushpin object โดยกำหนดรายละเอียดซึ่งประกอบด้วย กำหนดชื่อให้กับตัว object ซึ่งต้องมีค่าไม่ซ้ำกัน, ตำแหน่งทางภูมิศาสตร์ของ pushpin, รูปภาพปักแทนหมุด, ข้อความสำหรับ (title), และคำอธิบาย (description) ดังนี้

```
Var pinID = 1;
```

```
var pin = new VEPushpin (pinID, new VELatLong (13.6, 100.33), null, 'my pushpin', 'this is pushpin');
```

2. เรียกใช้ VEMap.AddPushpin method เพื่อวางหมุดลงบนแผนที่และทำให้รูปหมุดปรากฏขึ้น ด้วยรหัสดังนี้ map.AddPushpin (pin);

4.4.3 การ customize รูปแบบของหมุด (Customizing Pushpin Style)

1. จัดค่า URL ที่ชี้ไปยัง icon หรือ image ที่เราต้องการใช้กับ pushpin ตามตัวอย่างต่อไปนี้

```
var pin = new VEPushpin(
    pinID, new VELatLong(27.69, 85.313),
    'http://dev.live.com/pict92.jpg',
    'My House', 'This is where my house is.'
);
```

2. สอดแทรก HTML ลงในส่วน of title และ description field ดังนี้

```
var pin = new VEPushpin (pinID,
    new VELatLong(27.69, 85.313),
    'http://dev.live.com/pict92.jpg',
    '<a href="http://dev.live.com/myhouse.htm">My House</a>',
    'This is where my house is.'
);
```

3. ใช้เพิ่มข้อมูล cascading style sheet (.css) ทำหน้าที่กำหนด icon, title, และ enhanced preview ตามแบบต่อไปนี้

```
<link rel="stylesheet" type="text/css" href="pushpins.css">
```

```
var pin = new VEPushpin(
    '10', //ID
    new VELatLong(47.6, -122.33), //Location
    null, //Icon
    Url 'Seattle', //Title
    'The emerald city', //Details
    'iconStyle', //Icon style class name
    'titleStyle', //Title style class name
```

```
'previewStyle' //Preview style class name
);
```

4.4.4 การเขียนเส้น และ รูปหลายเหลี่ยม (Drawing Lines on Map)

1. สร้าง array ของ VELatLong object ซึ่งทำหน้าที่กำหนดจุดต่างๆ บนเส้น polygon โดยใช้
พิกัดภูมิศาสตร์ละติจูด-ลองจิจูด ดังตัวอย่าง

```
var points = new Array(
    new VELatLong(45.01188, -111.06687),
    new VELatLong(45.01534, -104.06324),
    new VELatLong(41.01929, -104.06),
    new VELatLong(41.003, -111.05878),
    new VELatLong(45.01188, -111.06687)
);
```

2. สร้าง VEPolyline object ขึ้นใหม่ โดยกำหนด ID, array ของ VELatLong objects, สีของเส้น, และความหนาของเส้น (หน่วยเป็น pixel) ดังตัวอย่างต่อไปนี้

```
var poly = new VEPolyline('1', points, linecolor, 2);
```

3. ใช้ Method ชื่อ VEMap.AddPolyline โดยใช้ poly ซึ่งเป็น VEPolyline object ที่เพิ่งทำให้เพื่อเพิ่มเข้าไปบนแผนที่ดังตัวอย่างนี้

```
map.AddPolyline(poly);
```

ตัวอย่างการเขียนเส้น

```
var map = null;

function GetMap(){ map = new VEMap('myMap');
    map.LoadMap();
    DrawPoly();
}

function DrawPoly(){
    var points = new Array(
        new VELatLong(45.01188,-111.06687),
        new VELatLong(45.01534,-104.06324),
        new VELatLong(41.01929,-104.06),
        new VELatLong(41.003,-111.05878),
        new VELatLong(45.01188,-111.06687)
    );

    poly = new VEPolyline('1',points);
    poly.SetWidth(3);
    poly.SetColor(new VECOLOR(0,150,100,100));
```

```
map.AddPolyline(poly);
map.SetMapView(points);
}
```

4.4.5 การใช้งานชั้นข้อมูลพิเศษบนแผนที่ (Working with Layers)

ชนิดของข้อมูลสำหรับทำชั้นข้อมูลมีข้อมูล 2 ชนิดที่สามารถนำมาแสดงผลบนชั้นข้อมูลได้แบบที่ 1 ชุดข้อมูลแผนที่ซึ่งมีผู้ทำไว้เป็นของสาธารณะด้วย Windows Local Live(ซึ่งจะเรียกอย่างย่อว่า VECollection) ถ้าพบว่ามีข้อมูลประเภทนี้ที่เราสนใจจะนำมาแสดงบนแผนที่ ขั้นแรกให้หา GUID (globally unique identifier) ของ collection ตัวนั้น เช่น ชุดข้อมูลสิ่งมหัศจรรย์ 7 แห่งของโลกโบราณมี GUID เป็น "525739C1579D1124!936" ในกรณีที่เราเป็นสมาชิก Windows Local Live แล้ว และเคยสะสมรวบรวม VECollection เก็บไว้ใน Windows Live Favorites Web page (<http://favorites.live.com/>) เราอาจใช้ที่นั่นเป็นแหล่งค้นหาได้ เพราะมีการทำ permalink เชื่อมโยงกับ collection ที่รวบรวมไว้นั้นด้วย GUID ของ collection

แบบที่ 2 ข้อมูลในรูปแบบ GeoRSS ซึ่งเป็นเอกสารจำพวก xml (Extensible Markup Language) และเป็น RSS (Really Simple Syndicate) แบบขยายให้มีข้อมูลเชิงพื้นที่แทรกเข้าไปในเนื้อหาด้วยข้อมูลประเภทเดียวกับ GeoRSS มีอยู่หลายรูปแบบเป็นแบบที่ยึดตามมาตรฐาน W3C Basic Geo Vocabulary

การสร้างชั้นข้อมูลเพิ่มบนแผนที่ มีขั้นตอนดังนี้

1. สร้าง Object ชนิด VELayerSpecification ดังนี้

```
var veLayerSpec = new VELayerSpecification();
```

2. กำหนด property ต่างๆ ของ VELayerSpecification object ที่สร้างขึ้น ดังรายละเอียดต่อไปนี้

2.1 ใช้ property ชื่อ VELayerSpecification.Type เพื่อกำหนดชนิดของชุดข้อมูลที่จะใช้ โดยป้อนค่าเป็น VELayerType ซึ่งในทางปฏิบัติให้ใช้รูปแบบอย่างใดอย่างหนึ่งต่อไปนี้

```
veLayerSpec.Type = VELayerType.GeoRSS;//for GeoRSSหรือ veLayerSpec.Type =
VELayerType.VECollection;//for VECollection
```

2.2 VELayerSpecification.ID property ของชั้นข้อมูล ต้องใช้ค่าที่เป็นเอกลักษณ์ (ไม่ซ้ำใคร) เช่น

```
veLayerSpec.ID = 'layer1';
```

2.3 ถ้าคุณสมบัติของ LayerType ถูกกำหนดเป็น GeoRSS ต้องระบุค่า property ตัวที่ชื่อ VELayerSpecification.LayerSource ด้วย URL ของเพิ่มข้อมูล RSS ที่ต้องการนำมาใช้ แต่ถ้า

กำหนด property ของ LayerType เป็น VECollection ต้องระบุ property ของ LayerSource เป็นค่า GUID ของ WLL collection ชุดนั้น ดังนี้

```
veLayerSpec.LayerSource = "myLocalGeoRSS.xml";//for GeoRSSหรือ
```

```
veLayerSpec.LayerSource = "525739C1579D1124!936";//for VECollection
```

2.4 ถ้าได้กำหนด property ของ LayerType เป็น GeoRSS ต้องกำหนดค่า property ของ VELayerSpecification.Method ให้เป็น HTTP method (มี 2 แบบ 'get' หรือ 'post') ที่เหมาะสมในการใช้งาน เช่น

```
veLayerSpec.Method = 'get'; หรือ veLayerSpec.Method = 'post';
```

2.5 อาจกำหนด callback function ที่จะถูกเรียกใช้เมื่อชั้นข้อมูลได้โหลดเสร็จสมบูรณ์ โดยการตั้งค่า property ของ VELayerSpecification. FnCallback เป็นชื่อของฟังก์ชันนั้น ตัวอย่างต่อไปนี้ onFeedLoad เป็น callback function ที่กำหนดให้เรียกใช้ veLayerSpec.FnCallback = onFeedLoad;

3. เรียกใช้ method ชื่อ VEMap.AddLayer โดยใช้ VELayerSpecification object ที่กำหนดไว้ในขั้นแรก (ในที่นี้คือ veLayerSpec) เป็นพารามิเตอร์

```
map.AddLayer(veLayerSpec);
```

ตัวอย่างการใช้งาน Layer

```
var veLayerSpec = new VELayerSpecification();
```

```
veLayerSpec.Type = VELayerType.VECollection;
```

```
veLayerSpec.ID = 'layer1';
```

```
veLayerSpec.LayerSource = "1234567890ABCDEF!012";
```

```
veLayerSpec.FnCallback = onFeedLoad;
```

```
map.AddLayer(veLayerSpec);
```

```
function onFeedLoad(feed){
```

```
    alert('RSS or Collection loaded. There are ' + feed.length + ' items in this list.');
```

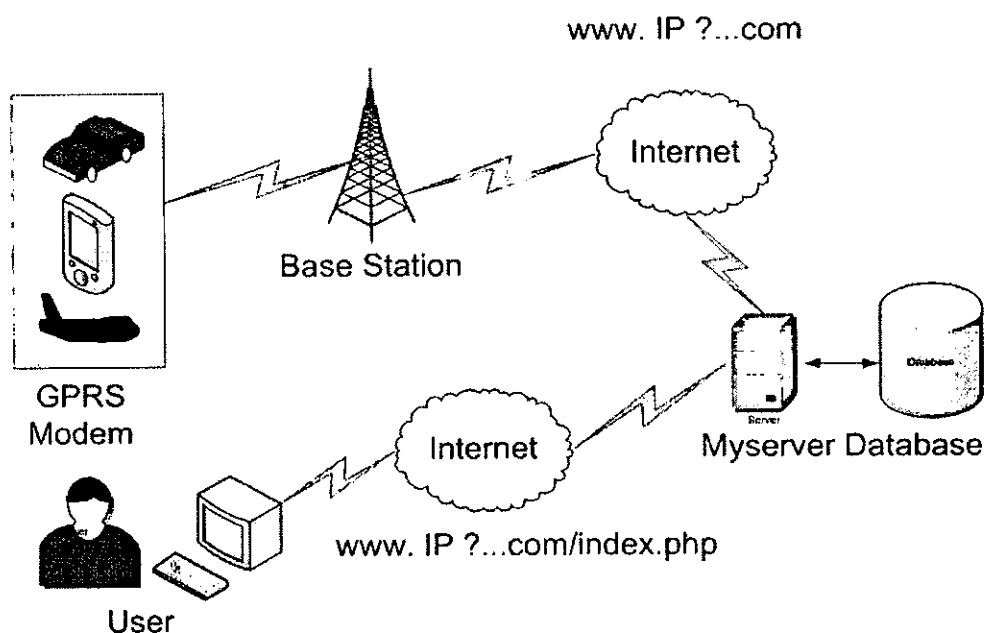
```
}
```

บทที่ 5

การออกแบบและขั้นตอนการดำเนินงาน

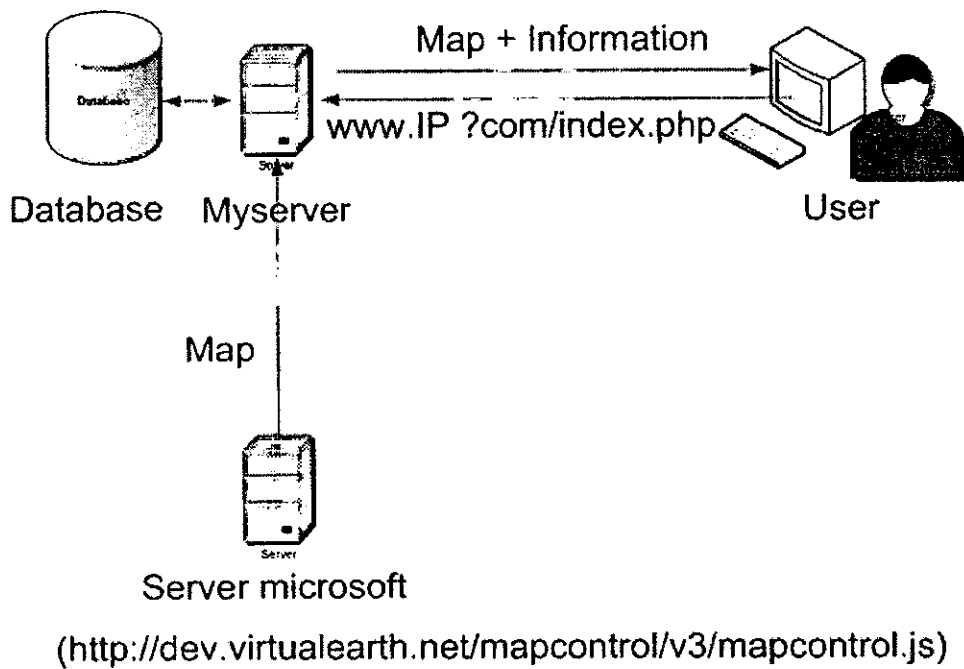
5.1 การทำงานของระบบ

เราดึงแผนที่มาจาก Local Live มาแสดงบนเว็บจากนั้นเราจะเอาข้อมูลที่เรามีอยู่เพิ่มเข้าไปในแผนที่ที่เราสามารถที่เลือกแสดงข้อมูลบางส่วนที่ต้องการได้โดยการแบ่งข้อมูลออกเป็นแต่ละ layer จากนั้นเราได้ทำส่วนรองรับข้อมูลที่ส่งมาจากระบบ GPRS เพื่อนำมาแสดงบนแผนที่และสามารถหาข้อมูลเก่าๆ (history) ที่วัตถุที่ต้องการได้อีกด้วยและต้องมีส่วนในการจัดการข้อมูลบนแผนที่



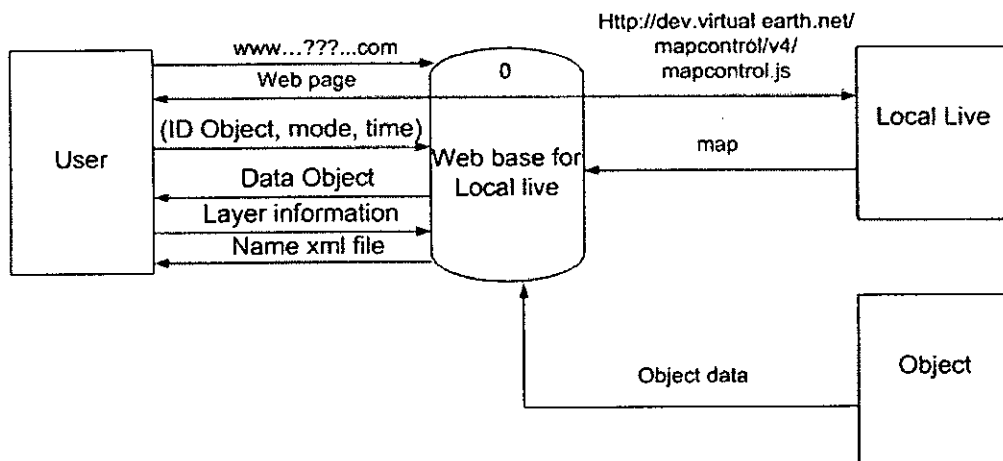
รูปที่ 5.1 รับข้อมูลจากวัตถุ

ตามรูปที่ 5.1 แสดงการทำงานของวัตถุที่ติดตามด้วยระบบ GPRS จากนั้นระบบจะส่งข้อมูลมาให้กับ Myserver แล้วเก็บไว้ในฐานข้อมูลแล้วผู้ใช้สามารถเข้ามาดูข้อมูลของวัตถุตัวนั้นๆ ได้

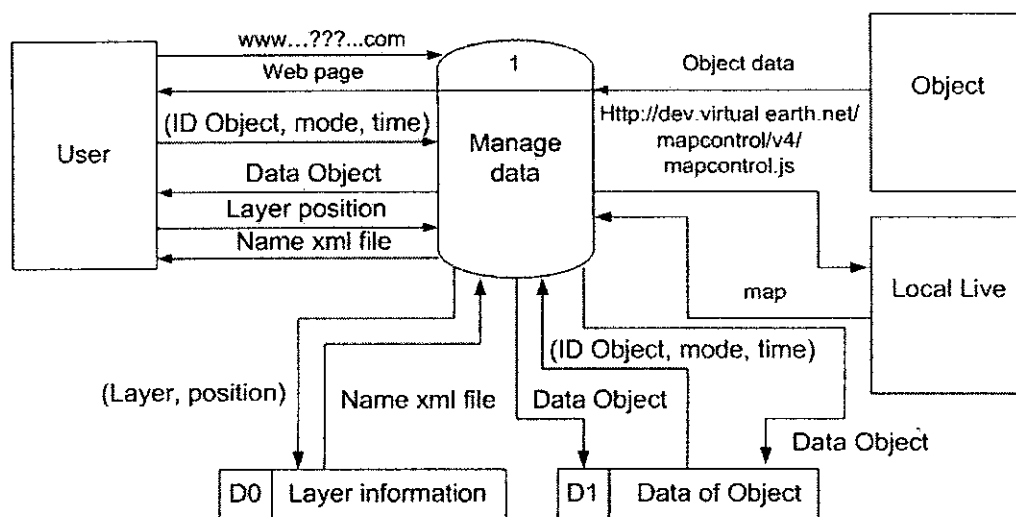


รูปที่ 5.2 โหลดแผนที่จาก Local live

ตามรูปที่ 5.2 เป็นส่วนในการดึงแผนที่มาจาก Local Live แล้วนำมาพร้อมกับข้อมูลต่างๆที่ระบบได้จัดเตรียมไว้ให้



รูปที่ 5.3 แสดงภาพรวมของระบบ



รูปที่ 5.4 หลักการทำงานของระบบ

ตามรูปที่ 5.4 เป็นลักษณะการทำงานของระบบซึ่งในระบบก็มีการทำงานออกเป็นหลายส่วน คือ

1. User คือผู้ใช้ที่เข้ามาใช้บริการของ web base for virtual earth local live
2. Local live คือผู้ให้บริการระบบแผนที่ของ virtual earth local live ที่เรานำแผนที่มาใช้งาน
3. Object คือวัตถุที่เราติดตามแล้ววัตถุนั้นได้ส่งข้อมูลเข้ามาเรื่อยๆ
4. Manage data เป็นระบบจัดการข้อมูลทั้งรับข้อมูลและค้นหาข้อมูล
5. Layer information คือข้อมูลของแต่ละ Layer ที่เราจะนำมาแสดงบนแผนที่
6. Data of object คือข้อมูลของวัตถุที่เราติดตาม

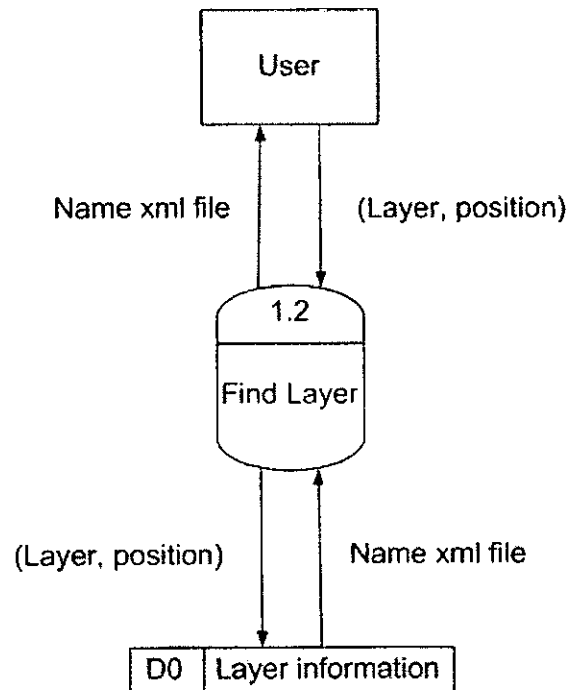
5.2 ส่วนประกอบของระบบ

- ส่วนติดต่อกับผู้ใช้งาน เป็นส่วนติดต่อกับผู้ใช้งานกับระบบเช่นส่วนที่ใช้ควบคุม Layer คือจะให้ Layer ไหนแสดงอยู่บ้าง
- ส่วนค้นหาข้อมูลของวัตถุ เป็นส่วนที่ใช้ค้นหาข้อมูลของวัตถุซึ่งแบบอยู่สองแบบคือ Real Time และ History time
- ส่วนของฐานข้อมูล มี 2 ส่วนคือ 1 ส่วนข้อมูลบนแผนที่ 2 ส่วนข้อมูลของวัตถุข้อมูลแต่ละแบบเป็นดังนี้

1. ส่วนข้อมูลบนแผนที่ประกอบด้วย ชื่อ Layer, ตำแหน่งบนแผนที่ (latitude, longitude), ชื่อของที่ตั้ง (ภาษาไทย, ภาษาอังกฤษ), รหัสเขต
2. ส่วนข้อมูลของวัตถุประกอบด้วย Id วัตถุ, ตำแหน่งบนแผนที่ (latitude, longitude), วันและเวลา

5.3 ส่วนของ Layer บนแผนที่

ลักษณะการทำงานของระบบเมื่อผู้ใช้เรียกดูข้อมูลของแต่ละ Layer ซึ่งภายในการทำงานยังได้แบ่งส่วนของแต่ละ Layer ออกเป็นส่วนๆ อีก

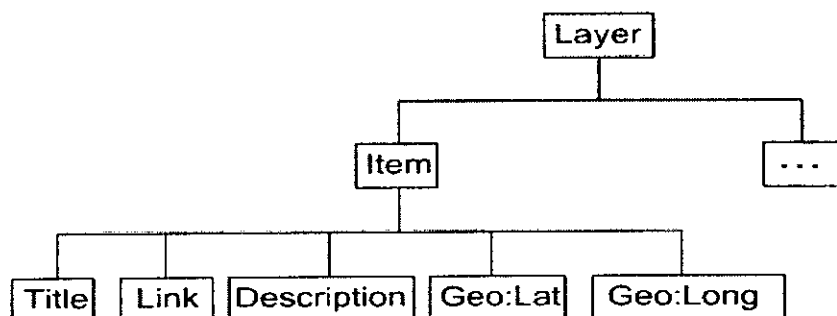


รูปที่ 5.5 แสดงการทำงานของระบบ

ข้อมูลของแต่ละ Layer สามารถนำไปใช้กับ ตัวแผนที่ได้นั้นมี 2 แบบคือ

1. เป็นรูปแบบของ GeoRSS XML File
2. VECollection เป็นรูปแบบของ local live ที่จัดเตรียมมาให้ใช้งาน

ซึ่งเราใช้แบบที่ 1 ตามรูปที่ 3.5 รูปแบบข้อมูลต้องอยู่ในรูปแบบของ XML File ที่ใช้รูปแบบของ GeoRSS



รูปที่ 5.6 แสดงรูปแบบของ GeoRSS

Item คือตัวกำหนดว่าหนึ่งจุดของ layer ใดๆ

Title คือหัวข้อหรือชื่อเรื่อง เช่น สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

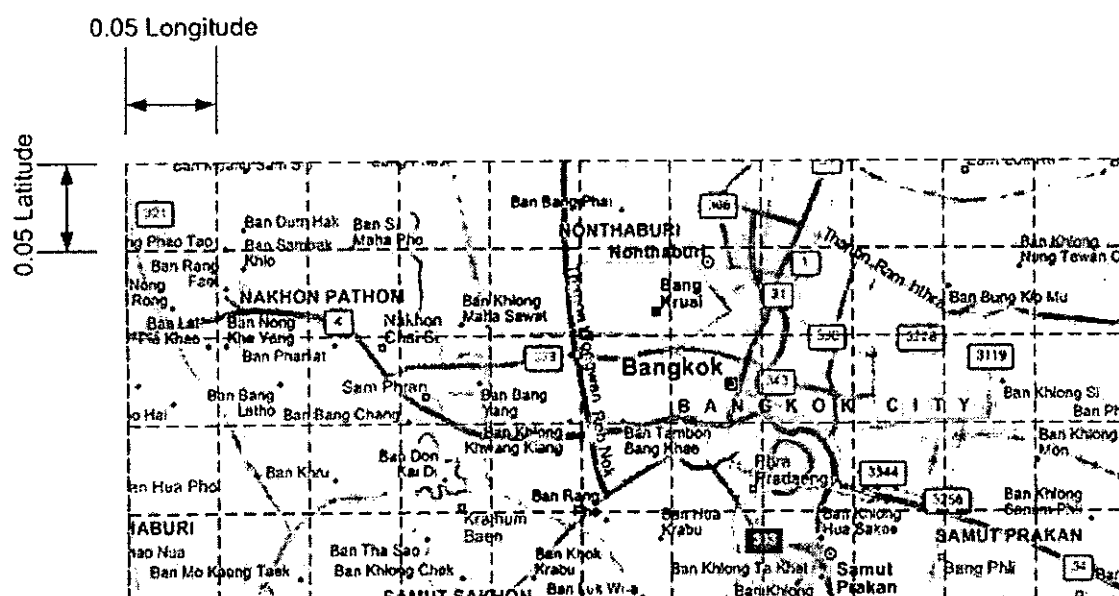
Link คือเมื่อเราต้องการให้ผู้ใช้สามารถที่ Link ไปยังเว็บที่เกี่ยวข้อง

Description คือคำอธิบายเพิ่มเติมเช่น บอกสถานที่ตั้งบอกวิธีติดต่อกับสถาบัน

Gco:Lat คือ บอกตำแหน่ง Latitude จริงบน โลก

Geo:Long คือ บอกตำแหน่ง Longitude จริงบน โลก

เราได้แบ่ง Layer ออกเป็นหลายๆ Layer เช่น ธนาคาร, วัด, โรงเรียน, มหาวิทยาลัย เป็นต้น เนื่องจากขนาดของ XML File มีขีดจำกัดเราจึงได้แบ่งกรอบออกเป็นหลายๆกรอบตามรูปที่ 5.4 เราได้กรอบสี่เหลี่ยมออกเป็น ขนาด $0.05 \text{ Latitude} * 0.05 \text{ longitude}$ เท่ากันออกทั่วประเทศในหนึ่งกรอบก็จะมีหนึ่ง XML File ที่ไม่ซ้ำกันอยู่ถ้ากรอบนั้นมีข้อมูล



รูปที่ 5.7 แสดงการตีกรอบของ Layer

การทำงานของระบบเราจะส่งจุดที่เราต้องการแสดงไปถาม Server ว่าจุดนั้นของ Layer นั้น XML File ชื่ออะไรที่จัดการบริเวณนั้นอยู่ เราได้เก็บข้อมูลนั้นอยู่ในฐานข้อมูลซึ่งมีทั้งภาษาไทยหรืออังกฤษดังในบริเวณนั้นก็จะมี 2 XML File ที่จัดการอยู่ถ้ามีข้อมูล

รูปแบบชื่อของข้อความที่ส่งไปหา Server ว่ามี XML File ใดที่จัดการส่วนนั้นอยู่มีรูปแบบดังนี้ `latitude=xx.xx&longitude=xxx.xx&layer=1,2,...;&mode=(eng or thai)` จากนั้นทาง Server จะตอบกลับมาว่า layer ที่ต้องการมี XML File ชื่ออะไรตามรูปที่ 5.5

```

<?xml version="1.0"?>
<file>
<item>
<layer> bank </layer>
<name> bank_eng 0.xml </name>
</item>
</file>

```

รูปที่ 5.8 รูปแบบข้อความที่ Server ส่งกลับมายัง User

ฐานข้อมูลที่จัดเก็บว่ารอบใดๆจัดเก็บ XML File ชื่ออะไรของแต่ละ Layer มีดังนี้

จากตาราง Layer

Layer_id คือรหัสของ Layer ที่ใช้อ้างอิงโดยที่ไม่ซ้ำกัน

Group_id คือใช้สำหรับจัดกลุ่มให้กับ Layer

Name_eng คือชื่อของ Layer ที่เป็นภาษาอังกฤษ

Name_thai คือชื่อของ Layer ที่เป็นภาษาไทย

จากตาราง Layer_data

Layer_id คือรหัสของ Layer ที่มีความสัมพันธ์กับ Layer_id ของตาราง Layer

Name_eng คือชื่อ XML File ที่ข้อมูลของ Layer เป็นภาษาอังกฤษ

Name_thai คือชื่อ XML File ที่ข้อมูลของ Layer เป็นภาษาไทย

latitude_start คือใช้กำหนดจุดเริ่มต้นของ latitude

longitude_start คือใช้กำหนดจุดเริ่มต้นของ longitude

latitude_end คือใช้กำหนดจุดสิ้นสุดของ latitude

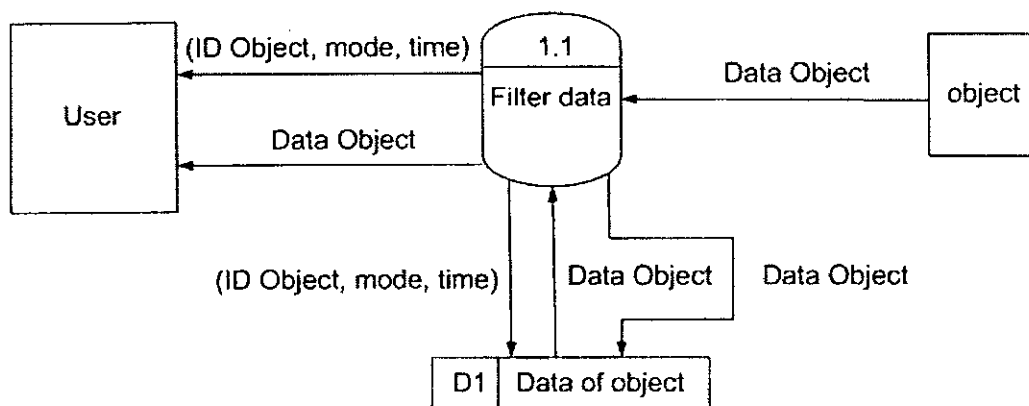
longitude_end คือใช้กำหนดจุดสิ้นสุดของ longitude

5.4 ส่วนการติดตามวัตถุใดๆ

ลักษณะการทำงานของระบบจะแบ่งออกเป็น 2 ส่วนคือ

1. ส่วนที่รับข้อมูลมาจากวัตถุ ข้อมูลที่ส่งมานั้นเป็นข้อมูลที่ยังไม่ได้ตรวจสอบ ข้อมูลอาจจะไม่ถูกต้องโดยตรวจสอบได้จากการที่เวลาผ่านไปเพียงเล็กน้อยแต่วัตถุกลับเดินทางจากหนึ่งไปยังอีกจุดหนึ่งไกลเกินไป

2. คือส่วนที่ผู้ใช้ต้องการเรียกข้อมูลของวัตถุออกมาแสดงซึ่งวิธีการเรียกดูข้อมูลจะมี 2 mode คือ real time และ history time



5.9 แสดงการทำงานของระบบ

คือเอาข้อมูลของวัตถุมาแสดงบนแผนที่หลักการทำงานก็เหมือนกับส่วนของ Layer แต่จะแตกต่างกันตรงที่ข้อความที่ส่งหากันระหว่าง user กับ server และการแสดงผลบนแผนที่ ซึ่งรูปแบบของข้อความที่ User ส่งไปยัง server มีรูปแบบดังนี้ ID=??&Mode= (real or history) ถ้าเป็นแบบ history ก็จะต้องมี &Sday=00.00.00&Eday=00.00.00&Stime=00:00:00&Etime=00:00:00 ต่อท้ายเข้าไปด้วยและทาง Server ก็จะมีรูปแบบของข้อความดังรูปที่ 5.6

ส่วนของการแสดงผลบนแผนที่นั้นเราจะต้องลากเส้นจากจุดเริ่มต้นจนไปถึงจุดสุดท้ายและเราได้บอกวันเวลาของจุดเริ่มต้นและจุดท้ายเข้าไปด้วย

```

<?xml version="1.0" ?>
<data>
<begin>
<latitude> ??? ??? </latitude>
<longitude> ??? ??? </longitude>
<dateandtime> 00.00.00 00:00:00 <dateandtime>
</begin>
<item>
<latitude> ??? ??? </latitude>
<longitude> ??? ??? </longitude>
</item>
<end>
<latitude> ??? ??? </latitude>
<longitude> ??? ??? </longitude>
<dateandtime> 00.00.00 00:00:00 <dateandtime>
</end>
</data>

```

รูปที่ 5.10 รูปแบบข้อความที่ Server ส่งกลับมายัง User

ในส่วนของฐานข้อมูลนั้นมีรูปแบบดังนี้

ID ของวัตถุเป็นชนิดตัวเลข

Dateandtime ใช้เก็บวันเวลาของวัตถุนั้นๆ เป็นชนิดวันและเวลา

Latitude ใช้สำหรับเก็บตำแหน่ง latitude ของวัตถุเป็นชนิด Double (8, 6)

Longitude ใช้สำหรับเก็บตำแหน่ง longitude ของวัตถุเป็นชนิด Double (9, 6)

Io1 และ Io2 ใช้สำหรับเก็บสถานะของ input/output ใดๆ ที่ต้องการเป็นชนิด Boolean

Distance ใช้เก็บระยะทางจากจุดก่อนหน้ามาถึงจุดตัวเองว่าห่างกันเท่าไรเป็นชนิด Double (12, 5)

Speed ใช้เก็บความเร็วจากจุดก่อนหน้ามาถึงจุดตัวเองว่าห่างกันเท่าไรเป็นชนิด Double (12, 5)

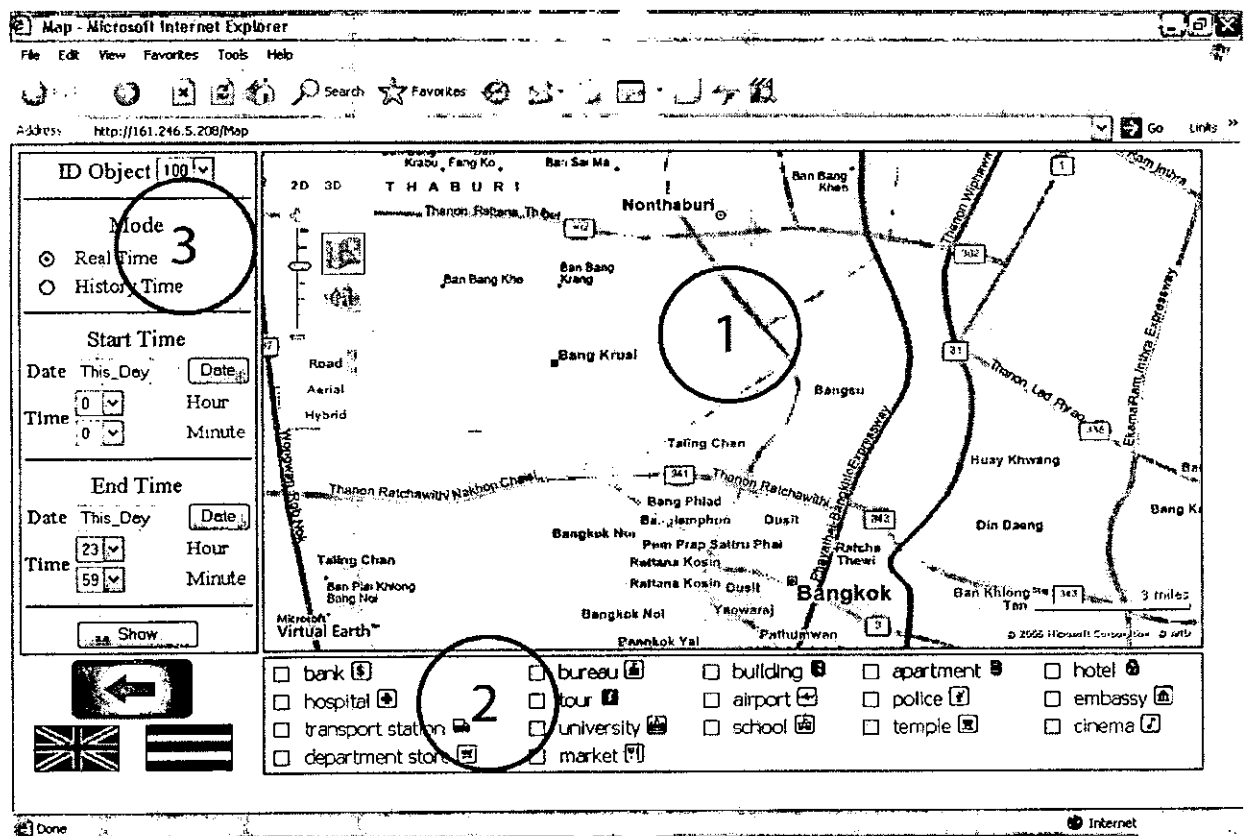
บทที่ 6

การทดลองและผลการทดลอง

6.1 บทนำ

ในส่วนของการทดลองใช้งานระบบแผนที่ทั้ง 3 ส่วนคือ (1) คือส่วนของแผนที่ที่ใช้แสดงผล (2) ส่วนควบคุมการแสดงผลของแต่ละ Layer (3) คือส่วนควบคุมการแสดงผลของวัตถุ นำทั้ง 3 ส่วนมาใช้กับระบบของแผนที่ Local live ว่าใช้งานรวมกันได้อย่างมีประสิทธิภาพหรือไม่

6.2 ขั้นตอนการทดลอง

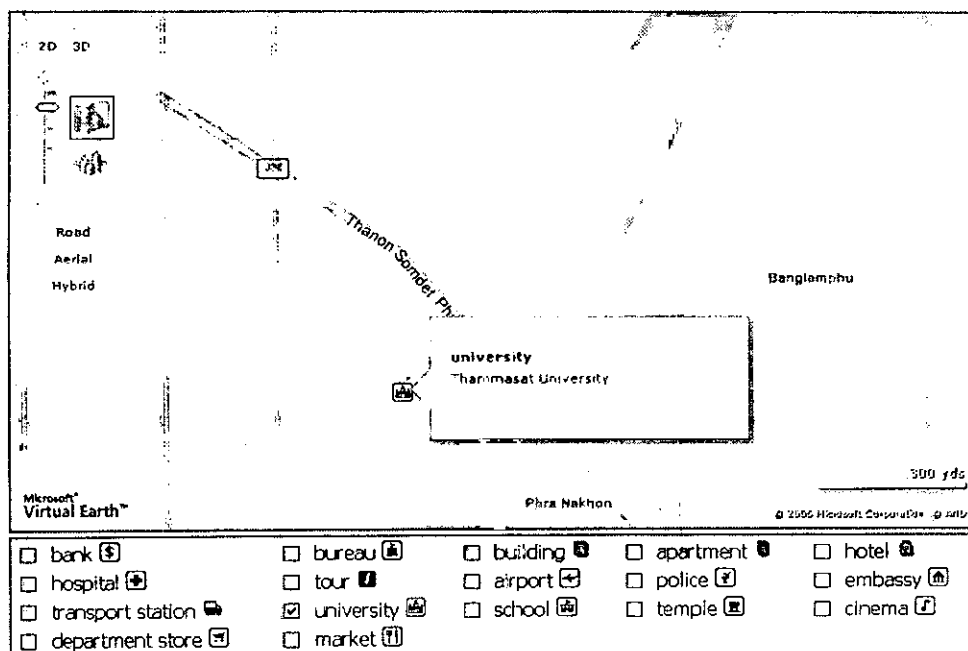


รูปที่ 6.1 แสดงระบบแผนที่

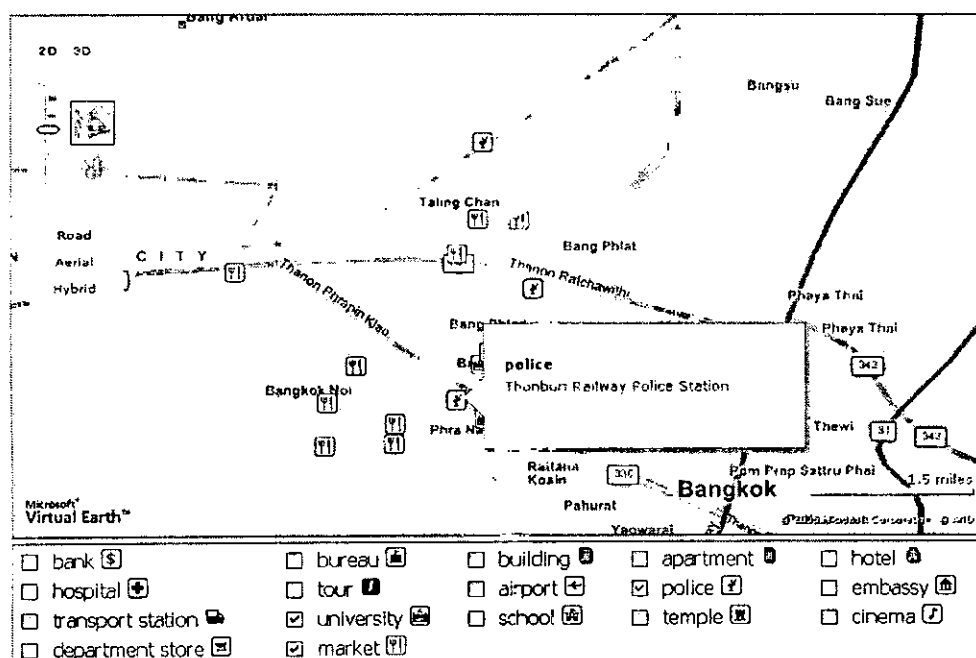
ทดลองส่วนของการแสดงผลของแต่ละ Layer (2)

ขั้นตอนแรกเราต้อง เข้าไปที่ URL <http://161.246.5.208/Map>

ส่วนของหมายเลข (2) คือส่วนควบคุม Layer ต่างจะเห็นว่าตอนนี้ยังอยู่ใน mode ภาษาอังกฤษเมื่อเราเลือกที่จะแสดง Layer ใด Layer หนึ่งการทำงานของระบบก็จะไปแสดง Layer นั้นบนแผนที่ ในส่วนของหมายเลข (1)

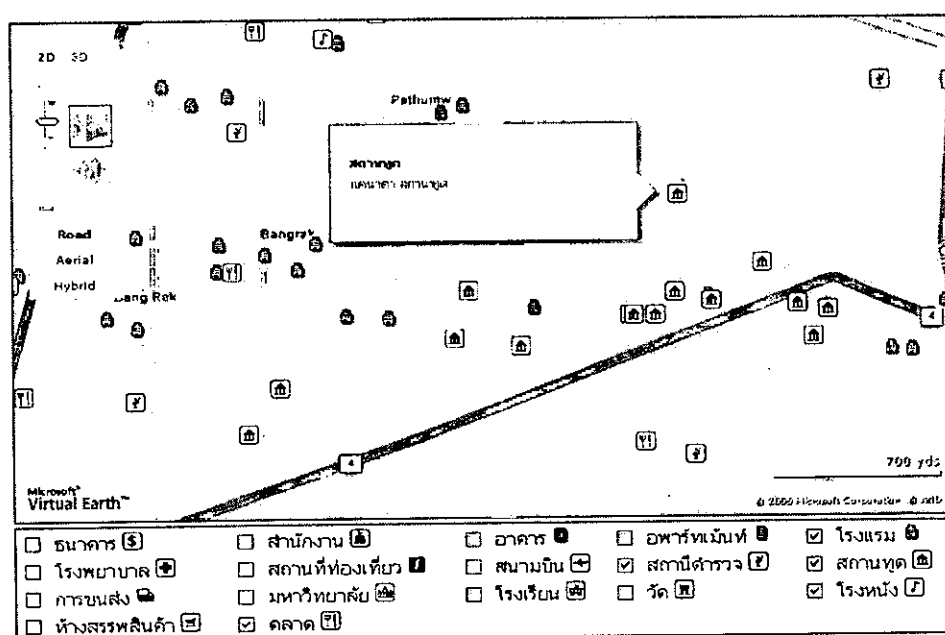


รูปที่ 6.2 ตัวอย่างแสดง Layer



รูปที่ 6.3 ตัวอย่างแสดงทีละหลายๆ Layer

เมื่อเราเลื่อนแผนที่ไปเรื่อยๆ Layer ที่เราเลือกอยู่ก็จะแสดงออกมาเลยตามจุดที่เราเลื่อน ตามมาเราสามารถเปลี่ยน mode ไปใช้ภาษาไทยได้โดยกดปุ่มไอคอนที่เป็นรูปธงชาติไทย



รูปที่ 6.4 ตัวอย่างแสดงทีละหลายๆ Layer และเป็นภาษาไทย

ทดลองส่วนของการแสดงผลของวัตถุ

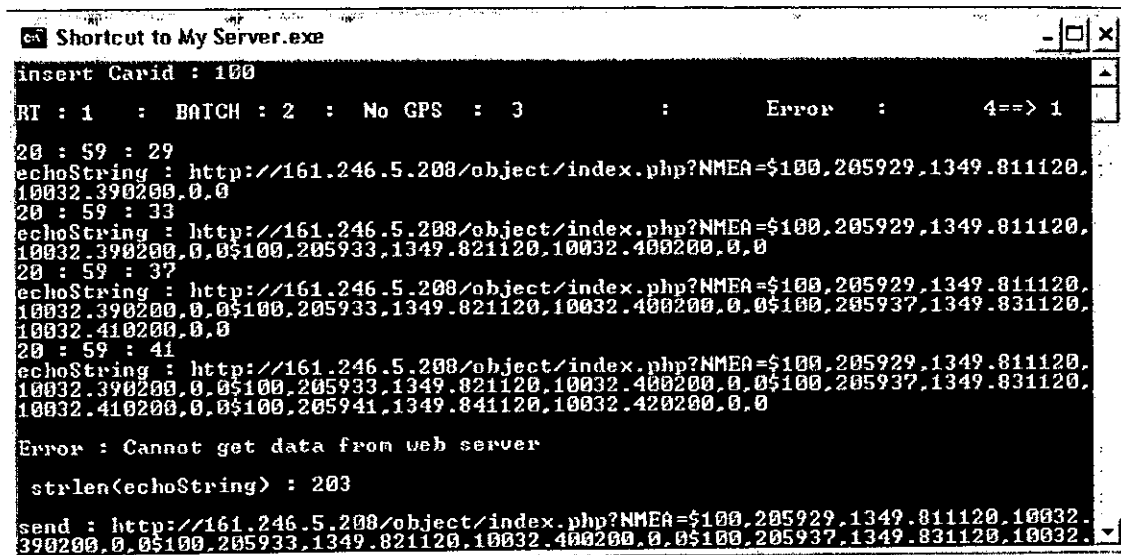
การทดลองของส่วนนี้แบ่งออกเป็น 2 ส่วนคือ

1. ส่วนของส่งข้อมูลไปที่ Server มีลำดับขั้นตอนการทำงานดังนี้

1.1 ใส่ Id ให้กับวัตถุที่ต้องการส่งไปยัง Server

1.2 เลือก mode ที่ 1

1.3 จากนั้นให้รอจนกว่าจะสมบูรณ์



```

Shortcut to My Server.exe
insert Carid : 100
RT : 1 : BATCH : 2 : No GPS : 3 : Error : 4==> 1
20 : 59 : 29
echoString : http://161.246.5.200/object/index.php?NMEA=$100,205929,1349.811120,
10032.390200,0,0
20 : 59 : 33
echoString : http://161.246.5.200/object/index.php?NMEA=$100,205929,1349.811120,
10032.390200,0,0$100,205933,1349.821120,10032.400200,0,0
20 : 59 : 37
echoString : http://161.246.5.200/object/index.php?NMEA=$100,205929,1349.811120,
10032.390200,0,0$100,205933,1349.821120,10032.400200,0,0$100,205937,1349.831120,
10032.410200,0,0
20 : 59 : 41
echoString : http://161.246.5.200/object/index.php?NMEA=$100,205929,1349.811120,
10032.390200,0,0$100,205933,1349.821120,10032.400200,0,0$100,205937,1349.831120,
10032.410200,0,0$100,205941,1349.841120,10032.420200,0,0
Error : Cannot get data from web server
strlen(echoString) : 203
send : http://161.246.5.200/object/index.php?NMEA=$100,205929,1349.811120,10032.
390200,0,0$100,205933,1349.821120,10032.400200,0,0$100,205937,1349.831120,10032.

```

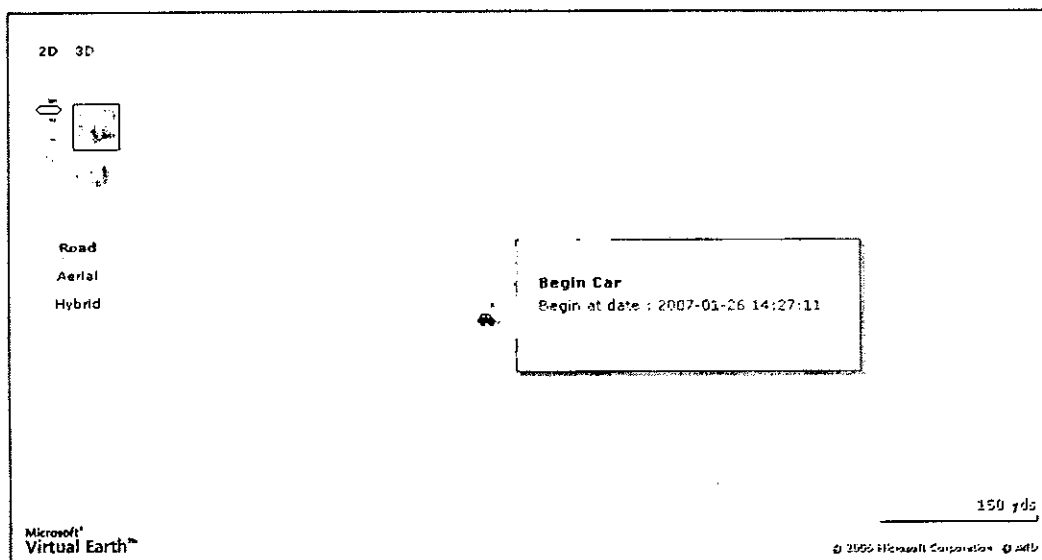
รูปที่ 6.5 แสดงการส่งข้อมูลไปที่ Server

2. การแสดงผลของวัตถุนั้นมี 2 แบบคือ 1 Real Time 2 History Time

ID Object 101		} 1
Mode ☒ Real Time ☐ History Time		} 2
Start Time Date This_Day Date Time 0 0 Hour Minute		} 3
End Time Date This_Day Date Time 23 59 Hour Minute		} 4
Show		} 5

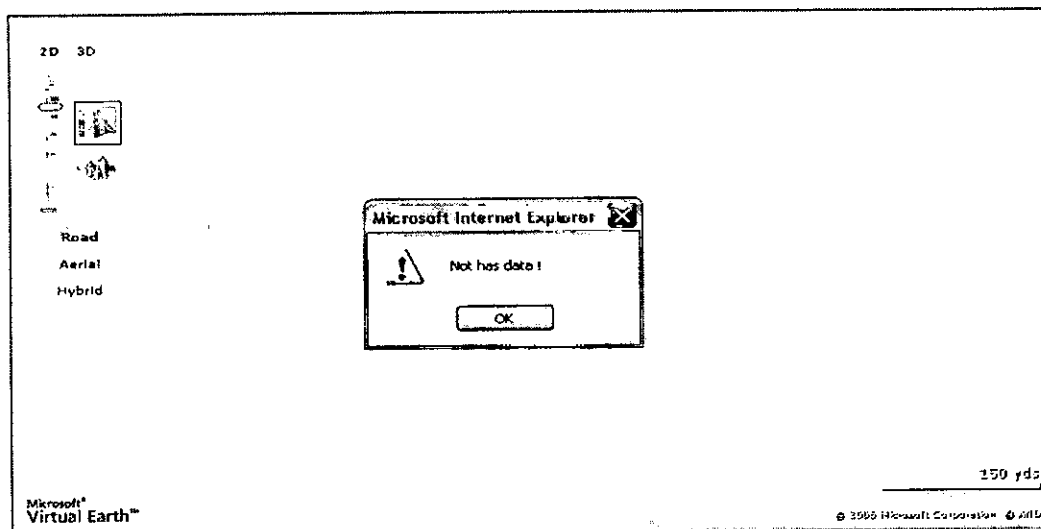
รูปที่ 6.6 ตัวควบคุมในการนำข้อมูลของวัตถุมาแสดง

เมื่อผู้ใช้ต้องการที่จะดูข้อมูลของวัตถุใดๆสามารถเลือกจาก Object ID จากช่องหมายเลข (1) และเลือกว่าต้องการดูแบบ Real Time หรือ History Time จากช่องหมายเลข (2) แบบ Real Time จะไม่นำเอาข้อมูลของช่องหมายเลข (3) และ (4) มาใช้ด้วยข้อมูลที่ให้นำมาแสดงก็นำเอาข้อมูลของวัตถุจากเวลาหนึ่งชั่วโมงที่แล้วจนถึงเวลาปัจจุบัน ตามรูปที่ 5.8 คือตัวอย่างการนำข้อมูลแบบ Real Time มาแสดงและจากนั้นก็ให้นำเอาข้อมูลใหม่มาแสดงใหม่ทุกๆทุก 10 วินาที



รูปที่ 6.7 แสดงการนำวัตถุมาแสดงบนแผนที่

ถ้าเราเลือกแบบ History Time เราก็ต้องเลือกช่วงเวลาที่ต้องการตามข้อหมายเลข (3) และ (4) การแสดงก็จะเหมือนกับ Real Time แต่จะไม่ไปเอาข้อมูลใหม่ๆ มาแสดงทุก 10 วินาทีและถ้าไม่มีข้อมูลอยู่ก็จะแสดงข้อความดังรูปที่ 6.8



รูปที่ 6.8 แสดงข้อความเมื่อไม่มีข้อมูล

บทที่ 7

สรุปผลการดำเนินงาน

7.1 สรุปผลการทดลอง

ทำให้เราเข้าใจในการสื่อสารกันระหว่าง Server กับ Client เข้าใจหลักการทำงานของภาษา PHP และสามารถนำไปใช้กับระบบแผนที่ Local Live ได้และได้เรียนรู้วิธีการแก้ปัญหาต่างๆที่เกิดขึ้นระหว่างการทำงานที่เราได้ศึกษาและทดลองการทำงานของส่วนต่างๆไม่ว่าจะเป็นส่วนของการสื่อสารกันระหว่าง Server กับ Client และวิธีการทำงานของ Local Live ว่าเราจะควบคุมการทำงานอย่างไร

7.2 ปัญหาและอุปสรรค

- ปัญหาเรื่องของข้อมูลที่จะนำมาเพิ่มบนแผนที่ในแต่ละ Layer ไม่สามารถหาข้อมูลที่ต้องการ 100%ได้เราจึงหาข้อมูลเท่าที่ทำได้และข้อมูลนั้นยังไม่ครอบคลุมทั่วประเทศและข้อมูลยังมีความซับซ้อนอยู่และความถูกต้องของข้อมูลยังไม่สามารถรับประกันได้
- เรื่องของขนาด XML File ที่มีจำกัดซึ่งเราจะสามารถเพิ่มข้อมูลเข้าไปในแต่ละ File ได้จำกัดดังนั้นเราต้องจัดการระบบให้ครอบคลุม
- ถ้า Server ของ Local Live ไม่สามารถทำงานได้ระบบของเราก็ไม่สามารถทำงานได้เช่นกันเพราะตัวแผนที่อยู่ที่ Server ของ Local Live เอง

7.3 ข้อเสนอแนะและแนวทางการแก้ปัญหา

- ศึกษาตัวแผนที่ใหม่ๆไม่ว่าจะเป็นของ Local live หรือ ใครงก็ตามเพื่อที่จะนำมาใช้กับระบบที่มีความทันสมัยมากขึ้น
- สร้างระบบที่สามารถเพิ่มข้อมูลเข้าไปในแต่ละ Layer และสามารถแก้ไขข้อมูล
- สร้างระบบที่สามารถสร้าง Layer ใหม่เข้าไปได้ตลอดเวลา
- สร้างระบบที่สามารถค้นหาข้อมูลใหม่บนอินเทอร์เน็ตในหลายๆรูปแบบและสามารถหาข้อมูลที่ดีที่สุด
- ออกแบบ User Interface ให้ดูน่าใช้งานมากขึ้น

7.4 แนวทางการพัฒนาต่อ

- ศึกษา version ใหม่ ๆ ของแผนที่เพื่อที่จะนำมาประยุกต์กับระบบของเราเอง
- สร้าง Interface ของระบบหน้าเว็บเพจที่สามารถเพิ่มจุดใดๆ เข้าไปใน Layer และ ลบจุดออกจาก Layer ได้ เพื่อง่ายและสะดวกในการเพิ่มจุดและลบจุดจาก Layer
- ระบบสามารถเข้าไปแก้ไขข้อมูลภายในจุดใดๆ ได้
- ระบบสามารถสร้าง Layer ใหม่ ๆ ได้
- สามารถที่จะค้นหาสิ่งที่ต้องการในแผนที่ได้เช่น สนามบินคอนเมืองว่าอยู่ที่ไหนในแผนที่และรายละเอียดของสนามบินว่ามีข้อมูลอะไรบ้าง

ก. Source code โปรแกรม ไฟล์ index.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<? //php header("Cache-Control: no-cache, must-revalidate"); ?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=tis-620" />
<title>Map</title>

<?php
    $mode_layer = $_GET['mode'];
    print "<script>";
    if($mode_layer == "" || $mode == "eng")print "var mode_layer = \"eng\";";
    else print "var mode_layer = \"thai\";";
    print "</script>";
    include("Control_map.js");
    include("class_system.php");
    include("func.php");
    //include("GetData.js");
?>
<script language="JavaScript" src="calendar.js"></script>
<link href="calendar-mos.css" rel="stylesheet" type="text/css">
<link rel="stylesheet" href="MapStyle.css" type="text/css" />
<script>
var mode = "real";
var idcar = "";
var sday = "";
var eday = "";
var stime = "";

```

```

var etime = "";
function Get_Map()
{
    GetMap();
    //setTimeout("alert('hello')",1250);
    var a="";
}
function Get(state)
{
    if(state == 0)
    {
        idcar = document.getElementById("id_car").value;
        if(document.getElementById("real").checked == true) mode =
document.getElementById("real").value;
        else mode = document.getElementById("history").value;
        sday = document.getElementById("s_carlender").value;
        eday = document.getElementById("e_carlender").value;
        stime = document.getElementById("s_hour").value + ":" +
document.getElementById("s_ss").value;
        etime = document.getElementById("e_hour").value + ":" +
document.getElementById("e_ss").value;
    }
    OnGet( idcar , mode , sday , eday , stime , etime);
    if (mode == "real")
    {
        //alert('hello');
        setTimeout("Get(1)",10000);
    }
}
function control_layer(name_layer)
{
    if(name_layer == "temple")

```

```

    {
        if(document.getElementById("temple").checked == true)
Show_Layer("temple");
        else Hide_Layer("temple");
    }
}
var menu_state = 0;
function Hidden_Menu()
{
    if(menu_state == 0)
    {
        document.getElementById("menu").style.left = "-220px";
        document.getElementById("myMap").style.left = "5px";
        document.getElementById("myMap").style.width = "980px";
        document.getElementById("ch").src = "img/show.png";
        menu_state = 1;
    }
    else
    {
        document.getElementById("menu").style.left = "5px";
        document.getElementById("myMap").style.left = "208px";
        document.getElementById("myMap").style.width = "780px";
        document.getElementById("ch").src = "img/hidden.png";
        menu_state = 0;
    }
}
</script>
</head>
<body onload="Get_Map();" bgcolor="#E3DFE0">
<div id="myMap" style="border:solid gray; border-width:2px;" class="Main"></div>

<?php Print_Menu(); ?>

```

```

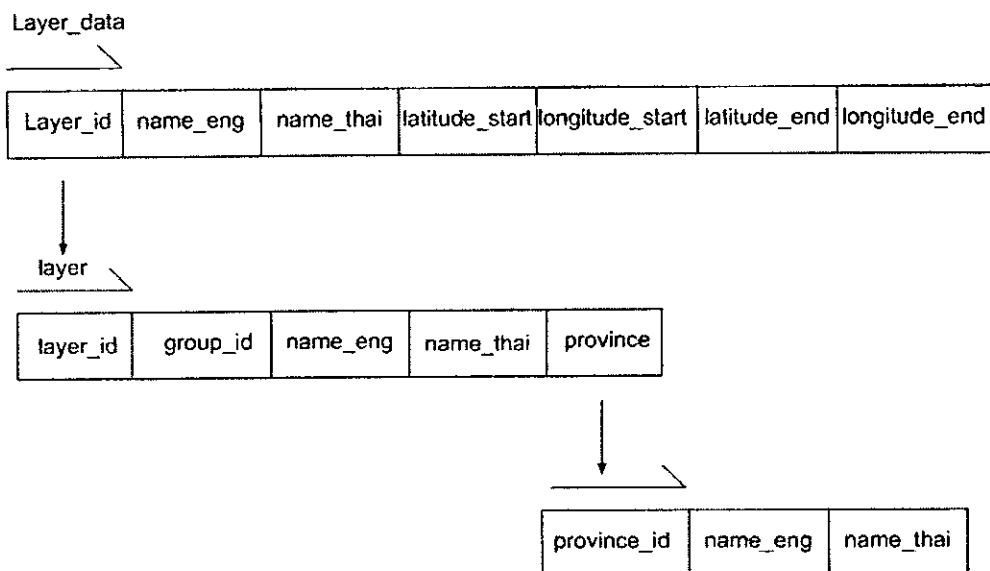
<div class="help">
<table width="190px">
<tr align="center">
    <td></td>
    <td><a href="/guide.zip"></a></td>
</tr>
<tr align="center">
    <td><a href="http://161.246.5.208/Map/index.php?mode=eng" ></a></td>
    <td><a href="http://161.246.5.208/Map/index.php?mode=thai" ></a></td>
</tr>
</table>
</div>

<div class="Control_Layer">
<table style="border: solid gray;border-width:2px; background-color: #FFFFCC;" width="780"
color="#990000">
<?php
    Set_Layer();
?>
</table>
</div>
</body>

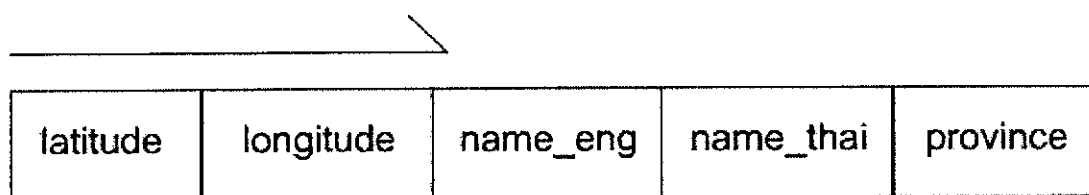
```

รูป ก. Source code โปรแกรม ไฟล์ index.php

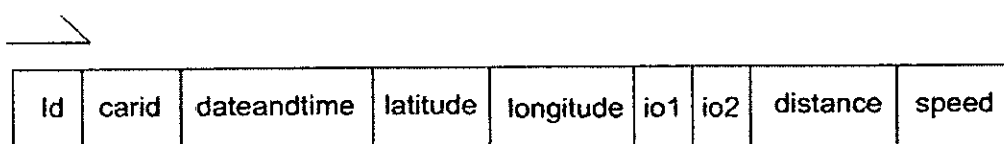
ข. Schema ของฐานข้อมูล



รูปที่ 1 Schema table layer



รูปที่ 2 Schema table hospital



รูปที่ 3 Schema table real history

บรรณานุกรม

- [1] <http://dev.live.com/virtualearth/sdk/>
- [2] <http://www.php.net/manual/en/>
- [3] <http://www.w3schools.com/>
- [4] <http://www.w3.org/2003/01/geo/>
- [5] <http://dev.mysql.com/doc/refman/5.0/en/>
- [6] <http://dev.mysql.com/doc/refman/5.0/en/>