

โปรแกรมสร้างและแปลงแผนที่

Map Generator and Converter

นายพันธ์ศักดิ์ บุญคง

นางสาวสาธิตา สะอาดเอี่ยม

เลขหมู่.....
เลขทะเบียน..... 62902
วันเดือนปี..... 23 ส.ค. 2549

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมสร้างและแปลงแผนที่
Map Generator and Converter



โดย
นายพันธ์ศักดิ์ บุญคง
นางสาวสาธิตา สะอาดเอี่ยม

อาจารย์ที่ปรึกษา
ดร. ชุติเมษฎ์ ศรีนิษฐา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมสร้างและแปลงแผนที่

Map Generator and Converter

ผู้จัดทำ

1. นายพันธ์ศักดิ์ บุญคง รหัสนักศึกษา 46015358

2. นางสาวสาธิตา สะอาดเอี่ยม รหัสนักศึกษา 46015380

อาจารย์ที่ปรึกษา

(ดร.สุติเมษฎ์ ศรีนิลทา)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมสร้างและแปลงแผนที่

นายพันธ์ศักดิ์ บุญคง	46015358
นางสาวสาธิตา สะอาดเอี่ยม	46015380
ดร.ชุตินเมษฐ์ ศรีนิลทา	อาจารย์ที่ปรึกษา
ปีการศึกษา 2548	

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้มีวัตถุประสงค์ในการนำเสนอโปรแกรม ที่มีความสามารถในการสร้างและแปลงแผนที่ เนื่องด้วยปัจจุบันมีข้อมูลแผนที่อยู่หลายรูปแบบด้วยกัน โปรแกรมประยุกต์โดยมากจะรองรับแผนที่ในรูปแบบใดรูปแบบหนึ่ง และในการทำงานบางครั้ง ต้องมีการแลกเปลี่ยนแผนที่ ทำให้เกิดแนวคิดในการพัฒนา Tools ที่มีความสามารถในการแสดงผลข้อมูลแผนที่ที่อยู่ในรูปแบบต่าง ๆ แก้ไขเพิ่มเติมข้อมูลและจัดเก็บข้อมูลแผนที่ในรูปแบบที่กำหนดได้ สำหรับรูปแบบที่โครงการนี้สนใจคือ GML (Geography Markup Language) , SVG (Scalable Vector Graphics) , PostGIS และ shp แผนที่ใน vector format เหล่านี้มีการเก็บจุดพิกัดของตำแหน่งบนพื้นโลกของสถานที่ต่าง ๆ ในรูปแบบที่แตกต่างกันออกไป การแปลง Format ของแผนที่ คือ การกำหนดค่าพิกัดต่าง ๆ ในตำแหน่ง และรูปแบบตามที่ Format นั้น ๆ กำหนด ทั้งนี้ข้อมูลอื่นที่ไม่ใช่ข้อมูลเชิงพื้นที่ อาจมีการสูญหายระหว่างแปลงได้

ปริญญานิพนธ์นี้นำเสนอการพัฒนาแผนที่ ด้วยการออกแบบและสร้างฐานข้อมูลเชิงแผนที่ และการจัดเก็บข้อมูลจะใช้ Postgre ในการจัดเก็บซึ่งผู้ใช้สามารถเรียกดูข้อมูลแผนที่ และสามารถแปลงข้อมูลแผนที่ของตนเอง ให้ออกมาอยู่ในรูปแบบของเอกสาร GML(Geography Markup Language) หรือ SVG (Scalable Vector Graphics) ได้

Map Generator and Converter

Mr.Pansak Boonkong 46015358

Miss.Satita Sa-ard-jeam 46015380

Dr.Chutimet Srinilta Advisor

Academic Year 2005

ABSTRACT

This thesis aims at presenting the program that is capable of generating and converting map. Today , there are a number of map data forms. Most applications serve only one format and is unavoidable. This gives rise to the attempt to develop the tools that are capable of presenting map data in various forms , modifying , adding and storing data in the set format. In this project , the emphasis is put upon the following file format : GML , SVG , PostGIS and shp. Maps in these vector formats record coordinate indicating the positions on Earth of places in many different forms. Map format convert is the way in which coordinate of the bearing and the forms is assigned in accordance to the format. Note that other data may be lost during converting process.

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำและ คำปรึกษาจาก ดร.ชุติเมษย์ ศรีนิลทา ซึ่งเป็นอาจารย์ผู้ควบคุมปริญญาบัตร ข้าพเจ้ารู้สึกซาบซึ้งในความอนุเคราะห์จากท่านอาจารย์ และขอขอบพระคุณเป็นอย่างสูง

ขอกราบพระคุณคณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุก ๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกๆเรื่อง ทำให้ข้าพเจ้าสามารถทำปริญญาบัตรฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากปริญญาบัตรฉบับนี้ ข้าพเจ้าขอบแต่ผู้มีพระคุณทุกท่าน

นายพันธ์ศักดิ์ บุญคง

นางสาวสาริตา สะอาดเอี่ยม

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มาของ โครงการ.....	1
1.2 วัตถุประสงค์ของ โครงการ.....	1
1.3 ขอบเขตของ โครงการ.....	1
1.4 วิธีการดำเนินการ.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.6 ส่วนประกอบของปริญญานิพนธ์.....	2
บทที่ 2 ทฤษฎีพื้นฐานที่ใช้ในวิทยานิพนธ์.....	4
2.1 ภาษา XML (Extensible Markup Language).....	4
2.1.1 เอเลเมนต์ (Element).....	4
2.1.2 แอตทริบิวต์ (Attribute).....	5
2.1.3 คอมเมนต์ (Comment).....	6
2.1.4 XLink, XPath , Xpointer.....	6
2.1.4.1 XLink.....	6
2.1.4.2 Xpath.....	8
2.1.4.3 XPointer.....	8
2.1.5 โอเปอเรเตอร์ (Operator).....	8
2.1.6 Namespaces.....	9
2.1.7 DTD และ XML Schema.....	9
2.1.8 XSL.....	9
2.1.9 XSLT.....	10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.1.10 XML Element.....	10
2.1.11 การแปลง XSL ไปเป็น XSLT.....	10
2.2 ภาษา GML (Geography Markup Language).....	10
2.3 ภาษา SVG (Scalable Vector Graphic).....	15
2.3.1 ความรู้เบื้องต้นเกี่ยวกับภาษา SVG.....	15
2.3.2 Link ของ SVG	15
2.4 การติดต่อกับฐานข้อมูล.....	17
2.4.1 PostgreSQL.....	17
2.4.1.1 หลักการทำงานของ PostgreSQL.....	18
2.4.1.2 ภาษา query ที่ใช้.....	18
2.4.2 PostGIS.....	19
2.4.2.1 ลักษณะทั่วไปของ PostGIS.....	19
2.4.2.2 การใช้รูปแบบมาตรฐาน OpenGIS.....	20
2.4.2.3 การสร้าง Spatial table.....	21
2.5 ShapeFile.....	22
2.5.1 ทำไมต้องเป็น ShapeFile.....	22
2.5.2 ShapeFile Technical Description	23
2.5.3 Naming Convention.....	23
2.5.4 Numeric Type.....	23
2.5.5 องค์ประกอบของ Main File.....	24
2.5.5.1 Byte Order.....	24
2.5.5.2 The Main File Header.....	25
2.5.5.3 Record Header.....	26
2.5.5.4 Main File Record Content.....	27
2.5.5.5 Null Shapes.....	27
2.5.5.6 Shape Type in X, Y Space.....	27
2.5.5.7 Measured Shape Type in X,Y Space.....	32
2.5.5.8 Shape Types in X,Y,Z Space.....	36
2.5.6 องค์ประกอบของ Index File.....	44

สารบัญ (ต่อ)

	หน้า
2.5.6.1 The Index File Header.....	45
2.5.6.2 Index Records.....	45
2.5.7 องค์ประกอบของ dBASE File.....	45
บทที่ 3 หลักการทำงานของโปรแกรม.....	46
3.1 โครงสร้างและการออกแบบของโปรแกรม.....	46
3.2 การนำข้อมูลจัดเก็บลงฐานข้อมูล.....	46
3.3 การนำข้อมูลจาก ShapeFile เก็บลงฐานข้อมูล.....	47
3.4 การนำข้อมูลจาก GML เก็บลงฐานข้อมูล.....	49
3.5 การสร้างแผนที่ในรูปแบบเอกสาร GML.....	51
3.6 การนำข้อมูลจาก SVG เก็บลงฐานข้อมูล.....	52
3.7 การสร้างแผนที่ในรูปแบบเอกสาร SVG.....	54
บทที่ 4 การทดลองและผลการทดลอง.....	56
4.1 Input.....	56
4.2 Output.....	58
บทที่ 5 บทวิจารณ์และสรุป.....	60
5.1 บทสรุป.....	60
5.2 วิจารณ์สิ่งที่ได้จากโครงงาน.....	60
5.3 ปัญหาอุปสรรคและแนวทางแก้ไข.....	60
5.4 แนวทางการพัฒนาต่อ.....	61
บรรณานุกรม.....	62

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงรายละเอียดของ main file header.....	25
2.2 แสดงค่าประจำตัวต่างๆใน Shape Type.....	26
2.3 แสดงรายละเอียดของ main file record header.....	27
2.4 แสดงความจุ Null Shape Record.....	27
2.5 Point Record Contents.....	28
2.6 Multipoint Record Contents.....	28
2.7 PolyLine Record Contents.....	29
2.8 Polygon Record Contents.....	31
2.9 PointM Record Contents.....	32
2.10 MultiPointM Record Contents.....	33
2.11 PolyLineM Record Contents.....	34
2.12 PolygonM Record Contents.....	36
2.13 PointZ Record Contents.....	36
2.14 MultiPointZ Record Contents.....	37
2.15 PolyLineZ Record Contents.....	39
2.16 PolygonZ Record Contents.....	41
2.17 MultiPatch Record Contents.....	44
2.18 รายละเอียดของ Index Records.....	45
3.1 แสดงขอบเขตของข้อมูลและคำอธิบาย.....	47
3.2 แสดง Element ของ GML.....	49
3.3 แสดงรายละเอียดของ Element ที่สำคัญในเอกสาร SVG.....	53
3.4 แสดงรายละเอียดของ Attribute ที่สำคัญในเอกสาร SVG.....	54

สารบัญรูป

รูปที่	หน้า
2.1 แสดงภาพรวมขั้นตอนการทำงาน.....	11
2.2 Program ที่ได้ทดลอง Implement ในระดับชั้นหมู่บ้าน.....	12
2.3 แสดงรายละเอียดของข้อมูลที่ถูกเลือก.....	12
2.4 แสดงการเลือกพื้นที่และรายละเอียด.....	13
2.5 แสดงรายละเอียดข้อมูลที่ถูกเลือก.....	13
2.6 ภาพแผนที่ ในรูปแบบ SVG ที่แสดงบน Web Brower.....	16
2.7 แสดง tag ของ SVG.....	16
2.8 องค์ประกอบของ main file.....	24
2.9 แสดงตัวอย่างของรูป Polygon.....	31
2.10 แสดงการเก็บของ Parts และ Points	31
2.11 แสดงตัวอย่างของชนิดของเส้น MultiPatch	42
2.12 ส่วนประกอบของ Index File	44
3.1 แสดงโครงสร้างและการออกแบบของโปรแกรม	46
3.2 แสดงขอบเขตและชนิดของข้อมูลแบบ point	48
3.3 แสดงขอบเขตและชนิดของข้อมูลแบบ Line	48
3.4 แสดงขอบเขตและชนิดของข้อมูลแบบ Polygon	49
3.5 แสดงไฟล์เอกสาร amphoe.gml	50
3.6 แสดงฐานข้อมูลที่ได้จากไฟล์ amphoe.gml	50
3.7 แสดงไฟล์เอกสาร amphoe.svg	53
4.1 แสดงการ Import ไฟล์เข้ามาโปรแกรม	56
4.2 แสดงภาพแผนที่ที่ถูก Import มาจาก shape file	57
4.3 แสดงฐานข้อมูลของ Shapefile	57
4.4 แสดงการสร้างข้อมูลแผนที่	58
4.5 แสดงภาพที่ได้จากการสร้างภาพแผนที่	59

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปริญญานิพนธ์

เนื่องจากเทคโนโลยีมีการพัฒนาอย่างต่อเนื่อง จากแผนที่ที่ใช้มือเขียนพัฒนามาเป็นการใช้เทคโนโลยีทางด้านคอมพิวเตอร์ ในการช่วยสร้างแผนที่ ที่มีความแม่นยำในการคำนวณ อัตราส่วนของแผนที่จริงกับแผนที่ที่เราสร้างขึ้น โดยในปัจจุบันมีการเขียน Software ที่ช่วยอำนวยความสะดวกในการสร้างแผนที่ ซึ่งในแผนที่ที่ผ่านการทำงานของ Software ตัวนี้ สามารถที่จะนำไปใช้ได้หลากหลายรูปแบบ เพื่อจะก่อให้เกิดประโยชน์สูงสุดในการใช้งานให้กับผู้ใช้ ซึ่งสามารถแปลงไฟล์ให้ออกมาในรูปแบบต่าง ๆ เพื่อความเหมาะสมกับการใช้งาน และไฟล์ที่แปลงออกมาได้จะเป็นไฟล์มาตรฐานที่ทั่วโลกให้การยอมรับ

ระบบสารสนเทศภูมิศาสตร์ (Geographic Information System: GIS) หมายถึง เครื่องมือที่ใช้ระบบคอมพิวเตอร์ เพื่อช่วยในการนำเข้า จัดเก็บ จัดเตรียม ดัดแปลง แก้ไข จัดการ และวิเคราะห์ พร้อมทั้งแสดงผลข้อมูลเชิงพื้นที่ ตามวัตถุประสงค์ต่างๆ ที่กำหนดไว้ ดังนั้น GIS จึงเป็นเครื่องมือที่มีประโยชน์เพื่อใช้ในการจัดการ และบริหารการใช้ทรัพยากรธรรมชาติและสิ่งแวดล้อม และสามารถติดตามการเปลี่ยนแปลงข้อมูลด้านพื้นที่ ให้เป็นไปอย่างมีประสิทธิภาพ เนื่องจากเป็นระบบที่เกี่ยวข้องกับระบบการไหลเวียนของข้อมูล และการผสมผสานข้อมูลจากแหล่งต่างๆ เช่น ข้อมูลปฐมภูมิ (Primary Data) หรือข้อมูลทุติยภูมิ (Secondary Data) เพื่อให้ได้สารสนเทศที่มีคุณค่าและสามารถนำไปใช้ในการบริหารจัดการได้อย่างมีประสิทธิภาพ

1.2 วัตถุประสงค์ของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้มุ่งหวังเพื่อพัฒนา Software ที่สามารถอำนวยความสะดวกในการใช้แผนที่ได้มากขึ้น ผู้ใช้สามารถนำเข้าข้อมูลจากเอกสารที่ต่างชนิดกันได้ เช่น Layer ของถนน อาจมาจาก GML (Geography Markup Language) ส่วน Layer ของแม่น้ำอาจมาจาก SVG (Scalable Vector Graphic) และสามารถ Convert file ให้ออกมาอยู่ในรูปแบบของ SVG หรือ GML ได้ หรือทำการแก้ไขข้อมูลบางส่วนให้ตรงกับความต้องการของผู้ใช้

1.3 ขอบเขตของปริญญานิพนธ์

ในปริญญานิพนธ์ฉบับนี้ได้ทำการสร้าง Software ที่มีความสามารถในการสร้างและเอกสารแปลงข้อมูลแผนที่ออกมาเป็นในรูปแบบของฐานข้อมูล (Postgres), ภาษา SVG (Scalable Vector Graphic) ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Graphic) หรือ ภาษา GML (Geography Markup Language) หรือต้องการเพียงข้อมูลนำออกมาเพื่อใช้ประโยชน์ในด้านต่าง ๆ ตามความต้องการของผู้ใช้ และโปรแกรมตัวนี้ยังสามารถทำการแก้ไขแผนที่ที่อาจจะมีการเพิ่มขึ้นของข้อมูล หรือมีการเปลี่ยนแปลงของข้อมูลได้อีกด้วย ซึ่งการแก้ไขนี้จะเลือกแก้ไขแบบเป็น Text , Point , Line , Polyline หรือ Polygon ก็ได้ซึ่งโปรแกรมตัวนี้จะใช้ภาษา C#.net เป็นตัวเขียนโปรแกรมขึ้นมา และในอนาคตแล้วโปรแกรมตัวนี้จะสามารถพัฒนาให้มีประสิทธิภาพเพิ่มขึ้นได้อีก และตรงตามความต้องการใช้งานตามยุคตามสมัย ส่วนของฐานข้อมูลนั้นจะใช้ Postgres เป็นตัวเก็บข้อมูลต่าง ๆ ของแผนที่

1.4 วิธีการดำเนินการ

1. ศึกษาเทคโนโลยีที่เกี่ยวข้องที่จะนำไปใช้ในการดำเนินงานในปฏิญานิพนธ์นี้
2. การนำเข้าข้อมูล SVG , GML และ ตารางจาก Shapefile มาแสดงผล
3. ทำการนำออกข้อมูลแผนที่ที่ได้รับเข้ามา มาแสดงผลออกเป็น GML และ SVG
4. ทำให้ตัวโปรแกรมสามารถให้ผู้ใช้แก้ไขข้อมูลในฐานข้อมูล

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถนำไฟล์ที่สร้างโดยโปรแกรมไปพัฒนาต่อได้โดยง่าย
2. สามารถนำไฟล์ที่แปลงด้วยโปรแกรมไปใช้ร่วมกับอุปกรณ์ PDA , Mobile ได้
3. สามารถแก้ไขแผนที่ที่เรามีอยู่ให้มีข้อมูลตรงกับความเป็นจริงในปัจจุบันได้
4. สามารถแปลงแผนที่ให้ตรงตามความต้องการใช้งานของผู้ใช้ได้
5. สามารถที่จะทำการเปลี่ยนแปลง แก้ไขข้อมูลต่างๆในฐานข้อมูลได้

1.6 ส่วนประกอบของปฏิญานิพนธ์

ปฏิญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความสำคัญและที่มาของปฏิญานิพนธ์ วัตถุประสงค์ของปฏิญานิพนธ์ ขอบเขตของปฏิญานิพนธ์ วิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปฏิญานิพนธ์

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในปฏิญานิพนธ์ ซึ่งประกอบด้วยทฤษฎี XML , SVG , GML และ Shape File พร้อมทั้งบรรยายทฤษฎีทั้งหมดโดยละเอียด

บทที่ 3 กล่าวถึงฟังก์ชันการทำงานที่สำคัญ ๆ ของตัวโปรแกรม และการแสดงผลในรูปแบบต่าง ๆ ของแผนที่ รวมถึงการจัดเก็บข้อมูล

บทที่ 4 กล่าวถึงการทดลองและผลการทดลองนั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 เป็นบทวิจารณ์และสรุป ซึ่งกล่าวถึงบทสรุปของปริญญาณิพนธ์ วิจารณ์สิ่งที่ได้รับจากปริญญาณิพนธ์ และข้อเสนอแนะสำหรับเป็นแนวทางในการพัฒนาต่อ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีพื้นฐานที่ใช้ในปริญญานิพนธ์

ในหัวข้อนี้จะกล่าวถึงทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องในการวิจัย และพื้นฐานของตัวภาษาที่ใช้ในระบบของโปรแกรมตัวนี้

2.1 ภาษา XML (Extensible Markup Language)

คอมโพเนนต์พื้นฐานของ XML คือ เอเลเมนต์ (Element) , แอตทริบิวต์ (Attribute) และคอมเมนต์ (Comment)

2.1.1. เอเลเมนต์ (Element)

เอเลเมนต์ (Element) เป็นสิ่งทีนำมาใช้เพื่อกำหนดเครื่องหมาย บางส่วนของเอกสาร XML ซึ่ง XML Element มีรูปแบบตามตัวอย่าง 2.1 ต่อไปนี้

`<ElementName>Content</ElementName>` (ตัวอย่าง 2.1)

เนื้อหาจะอยู่ภายในแท็ก XML โดย XML ครอบเนื้อหาภายในไว้ แต่สามารถมีเอเลเมนต์ที่ไม่มีเนื้อหาอยู่ภายในก็ได้ เราเรียกว่า Empty Element ใน XML สามารถเขียนได้ตามตัวอย่าง 2.2 ดังนี้

`<ElementName/>` หรือ `<ElementName></ElementName>` (ตัวอย่าง 2.2)

เอเลเมนต์ซ้อนเอเลเมนต์ สามารถซ้อนกัน ได้ตัวอย่างเช่นถ้าเราต้องการรวมกลุ่มข้อมูลทั้งหมดไว้ภายใต้เอเลเมนต์ชื่อเดียวกันทำได้ดังนี้

`<Patient>`

`<PatientName>Name</PatientName>`

`<PatientAge>Age</PatientAge>`

(ตัวอย่าง 2.3)

`<PatientWeight>Weight</PatientWeight>`

`</Patient>`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น XML Element สามารถบรรจุเอเลเมนต์อื่น ๆ ได้ อย่างไรก็ดีตาม แท็กเริ่มต้นทุกแท็กต้องมีแท็กปิดท้ายที่สอดคล้องกันเสมอ

ชื่อของเอเลเมนต์ต้องตั้งตามกฎต่อไปนี้

- ชื่อต้องไม่ประกอบด้วยอักขระช่องว่าง ไม่ว่าจะ เป็นเพียงหนึ่งช่องว่างหรือมากกว่า ถ้าชื่อเป็นอักขระตัวเดียว อักขระนั้นต้องเป็นพยัญชนะที่สามารถเป็นได้ทั้งตัวพิมพ์ใหญ่ (A-Z) หรือตัวพิมพ์เล็ก (a-z)
- ชื่อสามารถเริ่มต้นด้วยพยัญชนะหรือเครื่องหมาย Underscore (_)
- นอกจากอักขระตัวแรกเราสามารถใช้อักขระตัวใดก็ได้
- ชื่อของเอเลเมนต์มีลักษณะเป็น Case Sensitive (คือการคำนึงถึงตัวอักษรใหญ่หรือเล็กด้วย) ดังนั้น PatientName , PATIENTNAME และ patientname เป็นเอเลเมนต์ที่แตกต่างกัน

ตัวอย่างที่ 2.4 ชื่อของเอเลเมนต์ต่อไปนี้ เป็นรูปแบบที่ถูกต้อง

Fred

_Fred

Fredd123

FredGrüß

(ตัวอย่าง 2.4)

ตัวอย่างที่ 2.5 ชื่อของเอเลเมนต์ต่อไปนี้ เป็นรูปแบบที่ไม่ถูกต้อง

Fred 123

-Fred

123

(ตัวอย่าง 2.5)

จากตัวอย่าง 2.5 ชื่อเอเลเมนต์แรกประกอบไปด้วยช่องว่างระหว่างชื่อ ชื่อที่สองเริ่มต้นด้วยขีด (Dash) และชื่อที่สามเริ่มต้นด้วยตัวเลขแทนที่จะเป็นพยัญชนะหรือเครื่องหมาย Underscore (_)

2.1.2. แอตทริบิวต์ (Attribute)

แอตทริบิวต์ (Attribute) เป็นกลไกสำหรับการเพิ่มรายละเอียดข้อมูลไปยังเอเลเมนต์ เช่น ในเรคอร์ด PatientWeight ของเอกสาร XML เราไม่ทราบว่า PatientWeight มีหน่วยของน้ำหนักเป็นปอนด์หรือกิโลกรัม เพื่อให้รู้ว่า PatientWeight มีหน่วยน้ำหนักเป็นปอนด์ เราจะต้องเพิ่มแอตทริบิวต์ Unit และระบุค่าเป็น LB ตามตัวอย่างที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<PatientWeight Unit="LB" > Weight </PatientWeight > (ตัวอย่าง 2.6)

แอตทริบิวต์สามารถรวมอยู่ในเฉพาะแท็กเริ่มต้นเท่านั้นและมีลักษณะที่เหมือนอิลเมนต์ คือเป็น Case Sensitive ค่าของแอตทริบิวต์ต้องถูกครอบอยู่ในเครื่องหมาย Double Quotation (“”) แอตทริบิวต์สามารถใช้ร่วมกับ Empty Element เช่น ตามตัวอย่าง 2.7 ต่อไปนี้

< PatientWeight Unit="LB" /> (ตัวอย่าง 2.7)

เราสามารถประกาศแอตทริบิวต์ได้เพียงครั้งเดียวเท่านั้นในอิลเมนต์

2.1.3. คอมเมนต์ (Comment)

คอมเมนต์ (Comment) คอมเมนต์เป็นรายละเอียดที่ฝังอยู่ในเอกสาร XML เพื่อให้ข้อมูลเพิ่มเติมที่อยู่ภายในเอกสารคอมเมนต์ใน XML ในรูปประโยค (Syntax) ที่เหมือนกับรูปแบบคำสั่งที่ใช้ใน HTML ซึ่งทำให้ข้ามไปในขณะที่แอปพลิเคชันประมวลผลเอกสาร ตามตัวอย่าง 2.8

<!-- Comment text --> (ตัวอย่าง 2.8)

คำสั่งในการประมวลผล Processing Instruction (PI) ต้องเริ่มต้นด้วย “<?” และตามด้วยชื่อของ Application ที่จะรับข้อมูล PI target และข้อความที่จะส่งผ่านไปยัง application ตามตัวอย่าง 2.9

<?nameprocessor SELECT * FROM blah?> (ตัวอย่าง 2.9)

ซึ่งคำสั่งนี้จะถูกส่งไปยัง Parsers แต่จะไม่ถูกส่งไปยัง Application

2.1.4. XLink, XPath , Xpointer

XLink, XPath , Xpointer ความสามารถในการเชื่อมโยงเอกสารหนึ่งไปยังอีกเอกสารหนึ่ง

2.1.4.1 XLink คือการเชื่อมโยงไปเอกสารหนึ่งทั้งเอกสาร

Attribute ของ XLink ที่นำมาใช้กับ Element เป็นดังนี้

- type ใช้กำหนดชนิดของ element ของ XLink ที่กำลังจะสร้าง เช่น xlink:type = “simple” ใช้สำหรับ element แบบ simple

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- href คือ attribute บอกตำแหน่งใช้กำหนด URI(Uniform Resource Identifier)ในการดึงข้อมูลเช่น

```
<myelementxlink:herf="http://semaferna.com/home.htm">
```

- role ซึ่งถูกใช้ใน link ส่วนขยายเพื่อกำหนดหน้าที่ของแหล่งข้อมูล link ในรูปแบบที่เครื่องสามารถเข้าใจได้ ใช้แสดงความหมายและนัยสำคัญของลิงค์ให้กับแอปพลิเคชันซอฟต์แวร์จุดประสงค์เพื่อจัดเตรียมข้อมูลโดยรวมกับลิงค์ ซึ่งมนุษย์ไม่สามารถอ่านทำความเข้าใจได้ การใช้งานแอตทริบิวต์ ROLE ทำให้เราจัดเตรียม Application ร่วมกับข้อมูลรายละเอียดเกี่ยวกับ link มากกว่าที่จะเรียนรู้แต่เพียงว่า link ไปที่ใด

- title ซึ่งถูกใช้ใน link ส่วนขยายเพื่อกำหนดหน้าที่ของแหล่งข้อมูล link ในรูปแบบที่มนุษย์สามารถเข้าใจได้ title จะถูกนำไปสร้าง tool tip ซึ่งจะปรากฏขึ้นถ้าผู้ใช้เลื่อนเมาส์ไปวางอยู่เหนือลิงค์

- show กำหนดวิธีการที่แหล่งข้อมูลจะถูกนำมาแสดงผล เมื่อถูกอ่านเข้ามา ซึ่งมีค่าที่เป็นไปได้อยู่ 4 ค่าคือ

1. new ใช้ระบุว่าแหล่งข้อมูลจะถูกแสดงผลในหน้าต่างแยกต่างหาก
2. replace ใช้ระบุว่าเอกสารปัจจุบันถูกแทนที่ด้วยเอกสารใหม่
3. embed ใช้ระบุว่า xlink element นั้นจะถูกแทนที่ด้วยแหล่งข้อมูลใด เช่น รูปภาพ
4. undefined แอปพลิเคชันจะมีอิสระในการเชื่อมโยงกับลิงค์ด้วยวิธีที่เห็นว่าเหมาะสม

- actuate ใช้กำหนดว่าเมื่อไรที่แหล่งข้อมูลที่กำหนดจะถูกอ่านเข้ามาใช้ ซึ่งมีค่าที่เป็นไปได้อยู่ 3 ค่าคือ

1. onLoad แหล่งข้อมูลจะถูกดึงเข้ามาทันทีที่เอกสารนั้นถูกโหลด
2. onRequest แหล่งข้อมูลจะถูกดึงก็ต่อเมื่อผู้ใช้ต้องการมัน
3. undefined แอปพลิเคชันจะมีอิสระในการเชื่อมโยงกับลิงค์ด้วยวิธีที่เห็นว่าเหมาะสม

- Attribute from และ to เพื่อกำหนดทิศทางของ link โดยที่ link ของ xlink นั้นไม่จำเป็นต้องเป็นแบบทิศทางเดียว

ชนิดของ link

- Simple link เป็นลิงค์แบบทิศทางเดียวที่เชื่อมโยงแหล่งข้อมูลเพียงสองแหล่ง

- Extended link คือสิ่งที่เชื่อมความสัมพันธ์ระหว่างแหล่งข้อมูลจำนวนใด

เอกสารนี้เป็นเอกสารจำนวนหนึ่งเข้าด้วยกันงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Locator ถูกใช้สำหรับชี้แหล่งข้อมูลหนึ่งที่อยู่ห่างไกลที่เป็นส่วนหนึ่งของ link
- Arc ถูกใช้สำหรับชี้กฎการท่องจากแหล่งข้อมูลหนึ่งไปยังอีกแหล่งข้อมูลหนึ่งใน link
- Resource ใช้กำหนดแหล่งข้อมูลหลากหลายที่เป็นส่วนหนึ่งของ link
- Title สามารถให้อ่านข้อมูล title ของ link ที่มนุษย์อ่านเข้าใจได้

2.1.4.2 XPath

XPath คือภาษาที่ใช้เพื่อหาและได้ส่วนที่ต้องการใน XML files ส่วนที่ต้องการนั้นระบุโดยใช้ path ซึ่งคล้ายกับ path ที่ใช้ในการหา file ในคอมพิวเตอร์

XPath ทำหน้าที่ในการชี้ตำแหน่ง (address) ของส่วนต่าง ๆ ของเอกสาร XML เพื่อที่จะอนุญาตให้เราได้รับชิ้นข้อมูลที่เราต้องการได้ การสร้าง Path บอกตำแหน่ง

จะเริ่มต้นจากรากของเอกสาร (root node) เสมออย่างเช่น `"/order/item/part-number"` หมายถึง ให้เลือก element ใด ๆ ที่มีชื่อว่า `<part-number>` ซึ่งเป็นลูกของ element `<item>` ซึ่งเป็นลูกของ element `<order>` ซึ่งเป็นรากของเอกสารนั่นเอง

2.1.4.3 XPointer

XPointer ได้จัดเตรียมการเข้าถึงค่าของแอตทริบิวต์หรือเนื้อหาของเอเลเมนต์ใดที่อยู่ในเอกสาร XML ตำแหน่งใดก็ตาม การใช้ XPointer ทำคุณสามารถลิงค์ไปยังส่วนต่างๆ ของข้อความ หรือเลือกที่จะไปสู่แอตทริบิวต์หรือเอเลเมนต์ที่ต้องการ และไปยังเอเลเมนต์ต่างๆ เป็นต้น ข้อกำหนดลักษณะลักษณะเฉพาะไม่ได้เป็นตัวกำหนดว่าแอปพลิเคชันใดสามารถทำงานร่วมกับข้อมูลที่ถูกกินค่าโดย XPointer

2.1.5 โอเปอเรเตอร์ (Operator)

การกำหนดตำแหน่งของโหนดนั้นขึ้นอยู่กับชื่อของมัน โดยไม่สนใจว่ามันมาจากส่วนไหนในเอกสารโดยใช้สัญลักษณ์ `"/"` เช่น `"/customer"` จะเลือกเอา element `<customer>` ใด ๆ ที่มีอยู่ในเอกสารมา โดยไม่สนใจว่ามันขึ้นอยู่กับ element ไหน

XPointer

การเชื่อมโยงไปอีกเอกสารหนึ่งเฉพาะส่วนที่เราต้องการ (นี่คือชื่อแตกต่างจาก xlink) เช่น

`http://www.semaferna.com/default/page.xml#xpointer(//section-one)`

จะเห็นว่าเมื่อใส่ xpointer เข้าไปแอปพลิเคชันที่รับรู้ xpointer จะสนใจในส่วน of element `<section-one>` เท่านั้น และจะเห็นว่ามันคล้าย ๆ กับ xpath ซึ่งแท้จริงแล้ว xpointer นั้นสร้างขึ้นมาจาก xpath แต่จะไม่อ้างอิงถึงโหนดอีกต่อไป แต่จะใช้คำว่าตัวบอกตำแหน่ง (location) แทน เอกสารซึ่งตัวบอกตำแหน่งเหล่านี้สามารถเป็นได้ทั้งโหนดที่ใช้ใช้ใน xpath และ โหนดชนิดใหม่ ๆ ก็ารค่าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

point หรือ range ตัวใดตัวหนึ่งก็ได้ นอกจากนี้แทนที่จะสัมพันธ์กับโหนดเซต มันจะสัมพันธ์กับ location set แทน

2.1.6 Namespaces

Namespaces หมายถึงอะไรก็ตามที่เราใช้จำแนก element ของเอกสาร XML และในบางครั้งเราก็ยังใช้จำแนก attribute ได้อีกด้วยเนื่องจากโปรแกรมเมอร์อาจจะใส่ข้อมูลที่แตกต่างกันลงไป ใน element เดียวกัน เช่นในกรณีที่เราต้องผสมผสาน XML ที่เป็นของเราเองและ XHTML element ที่หลากหลายจากเอกสารหลายประเภทเข้าไว้ในเอกสาร XML ฉบับเดียว

XML Namespaces ทำงานอย่างไร

การที่จะใช้ XML Namespaces ในเอกสารของเรานั้น element ทั้งหมดก็ยังจำเป็นต้องถูกกำหนดค่าด้วยชื่อที่ตรงตามคุณสมบัติที่กำหนด หรือ QName ซึ่งจะประกอบไปด้วยส่วนประกอบสองส่วนนั่นคือ ส่วนที่เฉพาะ (local par) คือชื่อทั้งหลายที่เราหนดให้กับ element มาโดยตลอดและ Namespaces Prefix ซึ่งระบุว่าชื่อดังกล่าวนี้ขึ้นอยู่กับ Namespaces ไค

2.1.7 DTD และ XML Schema

การกำหนดโครงสร้างของเอกสาร XML ทำได้ 2 วิธีหลัก ๆ คือ DTD และ XML Schema โดยไม่ว่าจะกำหนดตามวิธีใดก็ตามแต่ เรียกว่าเอกสาร XML นั้น “valid”

เหตุผลที่ต้องกำหนดโครงสร้างของเอกสาร ก็เพื่อให้เอกสาร XML นั้นมีรูปแบบที่ถูกต้องพร้อมที่จะนำไปใช้งาน เพราะธรรมชาติของเอกสาร XML ก็คือการแลกเปลี่ยนข้อมูลระหว่างองค์กรหรือหน่วยงาน ดังนั้นฝ่ายที่รับเอกสาร XML จากองค์กรอื่น ต้องมั่นใจว่าเอกสารนั้นมีความถูกต้อง สามารถนำไปใช้งานได้ พุดให้ง่าย ๆ อีกอย่างหนึ่งก็คือ DTD หรือ XML Schema เปรียบเหมือน “สัญญา” ระหว่างหน่วยงาน หรือผู้ที่เกี่ยวข้องกับเอกสาร XML นั้น ๆ เมื่อใดก็ตามที่มีการสร้างเอกสาร XML ขึ้นมาเก็บข้อมูล จะต้องปฏิบัติตามข้อกำหนดใน DTD หรือ XML Schema หากไม่ทำตาม ก็ถือว่าผิดสัญญาและเอกสาร XML นั้นขอมนำไปใช้งานไม่ได้

สรุปความสำคัญของ DTD และ XML Schema ได้เป็น 2 แง่มุมคือ

1. เป็นการกำหนดรูปแบบโครงสร้างของเอกสาร XML (หรือเรียกว่า Data Model) ว่าประกอบไปด้วยแท็กอะไรบ้าง , มี Attribute อะไรบ้าง , ค่าของ Attribute เป็นอะไรได้บ้างหรือลักษณะการซ้อนแท็กเป็นอย่างไรได้บ้าง เป็นต้น
2. เป็นสัญญาระหว่างองค์กร ที่จะแลกเปลี่ยนข้อมูลกันในรูปแบบของ XML เพื่อให้เอกสารที่สร้างขึ้นมา มีรูปแบบตรงตามที่ตกลงกันไว้ และจะได้สามารถนำเอกสาร XML นั้นไปใช้งานได้ถูกต้อง

2.1.8 XSL

เอกสาร XSL คล้ายกับเอกสาร CSS ในกรณีที่ทั้งสองเอกสารมีการกำหนดสไตล์ที่ประยุกต์ไปยังเอเลเมนต์ที่แน่นอน แต่มีสิ่งที่แตกต่างกันเล็กน้อย CSS กำหนดสไตล์การพิมพ์ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉพาะ XHTML Element เท่านั้น ในทางตรงกันข้าม สไตล์ที่กำหนดเอกสาร XSL จะประยุกต์ไปยังเอกสาร XML เอกสาร XSL ทั้งเอกสารอาจใช้สไตล์ที่กำหนดใน CSS เพื่อสร้างโค้ดจากข้อมูล XML เอกสาร XSL ต้องวางบนเซิร์ฟเวอร์เดียวกัน รวมทั้งไฟล์ที่อ้างถึงเอกสารดังกล่าวด้วย

2.1.9 XSLT

เป็นข้อเสนอแนะจาก W3C ในการกำหนดชุด Element ของ XML ที่สามารถจะถูกใช้ในการสร้างสไตล์ชีตที่แปลงเอกสาร XML ให้อยู่ในรูปแบบใด ๆ ซึ่งในปัจจุบันนี้ใช้ในการแปลง XML หนึ่งไปเป็นอีกรูปแบบหนึ่ง โดยการทำงานจะทำโดยนำเอกสาร XML ที่เหมาะสมกับ XML ไปทำการแปลงรูปแบบเพื่อสร้างเอกสารเอาต์พุต ซึ่งขึ้นอยู่กับคำสั่งและเนื้อหาภายใน สไตล์ชีตจึงสามารถสร้างรูปแบบเอาต์พุตใด ๆ ก็ได้ตามที่ต้องการในการทำงาน

2.1.10 XML Element

XSLT ใช้เอเลเมนต์ที่เราใช้กับ XSL ในลักษณะที่เหมือนกัน เอเลเมนต์ต่อไปนี้ ซึ่งไม่มีการเปลี่ยนแปลง ได้แก่ apply-templates , attribute , element , comment , for-each , copy , if , choose , when และ otherwise โดย pi Element ที่ใช้ใน XSL จะเรียกว่า processing instruction ใน XSLT

XSLT เอเลเมนต์ต่อไปนี้ใน XSL นั้นไม่ได้มีอยู่ใน XSLT ได้แก่ script , cdata , entity-ref และ eval Element

2.1.11 การแปลง XSL ไปเป็น XSLT

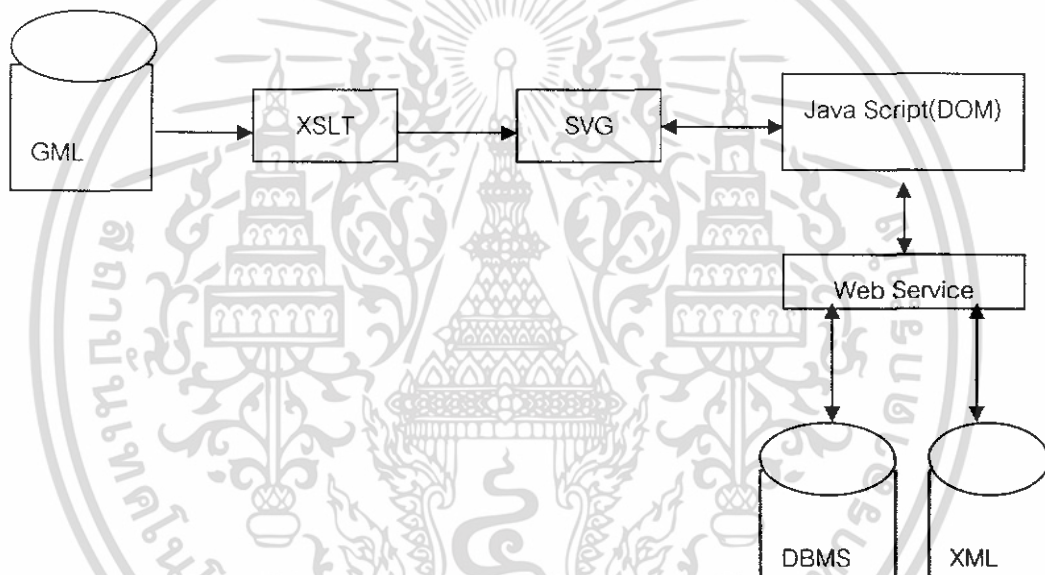
เราจำเป็นต้องอัปเดต XSL Style Sheet เพื่อให้เป็นไปตามข้อกำหนดของ W3C โดยเฉพาะอย่างยิ่ง เราจำเป็นต้องเปลี่ยนแปลง XSL และ Namespace ที่ส่งออกมา เนื่องจากบล็อก <xsl:eval> และ <xsl:script> ไม่ได้รับการสนับสนุนใน XSLT แต่จะมีการเปลี่ยนแปลงไปยังกลไกของ XSLT เรายังต้องเพิ่ม Jscript ของฟิงชัน XTL Runtime ในที่ๆจำเป็น เนื่องจากอ็อบเจกต์ XLT Runtime เป็นส่วนเพิ่มเติมอื่นๆ คือ เรายังต้องมีการใส่เครื่องหมายทั้งหมดที่เทียบเท่า XML อย่างเช่น \$ITS จะถูกแทนที่ด้วยเครื่องหมาย \square IT , \$gt\$ จะถูกวางด้วยเครื่องหมาย \square gt และอื่นๆ เครื่องหมายที่ไม่เท่ากับ XML ดังเช่น ieq นั้น ต้องเขียนฟิงชันพิเศษในการรองรับเอง

2.2 ภาษา GML (Geography Markup Language)

ภาษา GML (Geography Markup Language) เปรียบเสมือนที่เก็บข้อมูลภูมิศาสตร์ในรูปแบบของ XML มาแล้วแปลงข้อมูลด้วยภาษา XSLT ให้อยู่ในรูปแบบของ SVG เพื่อแสดงผลเป็นรูปภาพของแผนที่ และสามารถที่จะโต้ตอบกับผู้ใช้ได้โดยแทรก Code ของ JavaScript เข้าไป เมื่อเกิดการเลือกอาณาบริเวณในแผนที่ก็จะใช้ Java Script ที่แทรกอยู่ใน SVG File เพื่อใช้ javascript DOM ค้นหาข้อมูลใน GML ที่เป็นข้อมูลที่มีความสัมพันธ์กันกับพื้นที่ที่เลือกนั้นแล้วนำข้อมูลที่ได้จากการค้นหานั้น โดยข้อมูลที่ได้อาจเป็น ข้อมูลในส่วนที่ใช้ในการติดต่อกับเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

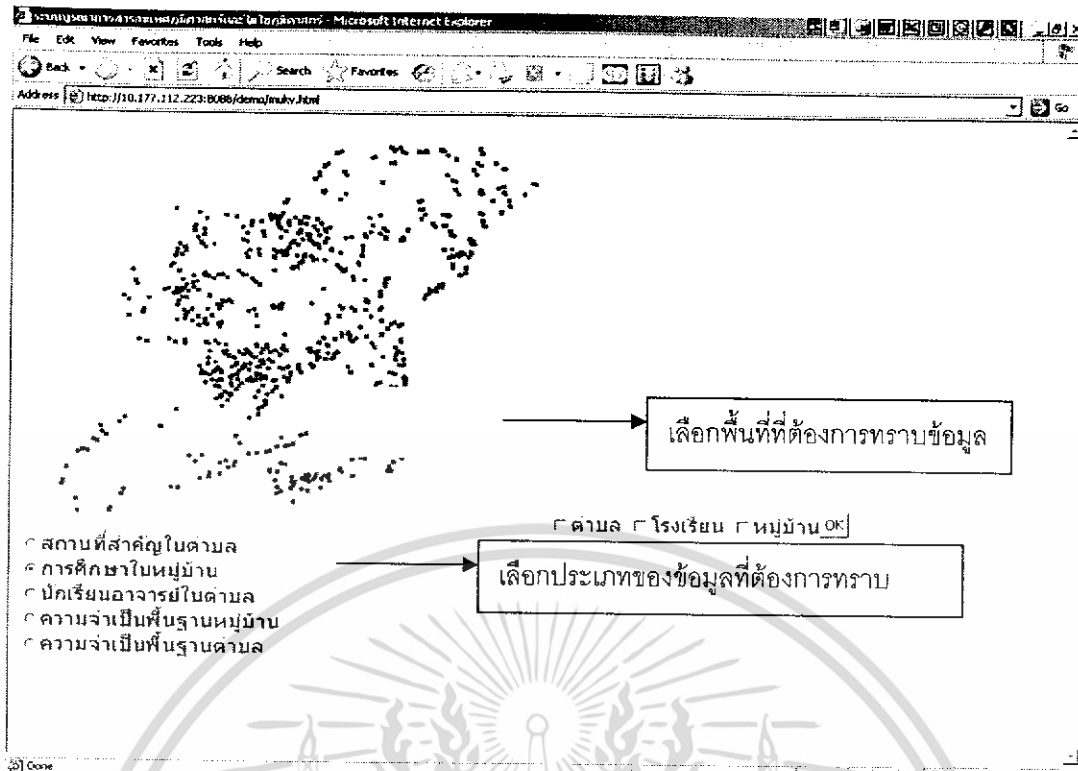
Webservice ว่าใช้ Webmethod ไหน หรือใช้ข้อมูลในตารางไหน หรือจะ parameter อะไรไป เพื่อที่จะให้ Webservice แล้ว Java Script ก็จะติดต่อกับ Webservice โดยใช้ Soap Protocol ทางฝั่ง ของ Webservice ก็จะนำข้อมูลที่รับจาก Client มาประมวลผลเพื่อหาข้อมูลที่ต้องการและส่ง ผลลัพธ์กลับไปเพื่อแสดงผลในฝั่ง Client

หลักการในการดำเนินงานได้แบ่งเป็น 2 ส่วน คือ ส่วนภูมิสารสนเทศ กับส่วนสารสนเทศ ในส่วนภูมิสารสนเทศ ได้ใช้เทคโนโลยี ของ GML มาเป็นส่วนเก็บข้อมูลพิกัดของแผนที่และ แสดงผลแผนที่ด้วย SVG ในส่วนของ สารสนเทศได้ใช้ Program C#.net มาพัฒนาในส่วนของ Webservice และใช้ Microsoft Access เก็บข้อมูลสารสนเทศ และใช้ Java Script เปรียบเสมือน ตัวกลางระหว่างข้อมูลทั้งสองคือเป็นตัวรับข้อมูลจากผู้ใช้ว่าต้องการทราบข้อมูลในส่วนไหนและ นำข้อมูลนั้น ไปติดต่อกับ Web service และรับข้อมูลกลับมาเพื่อแสดงผล



รูปที่ 2.1 แสดงภาพรวมขั้นตอนการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



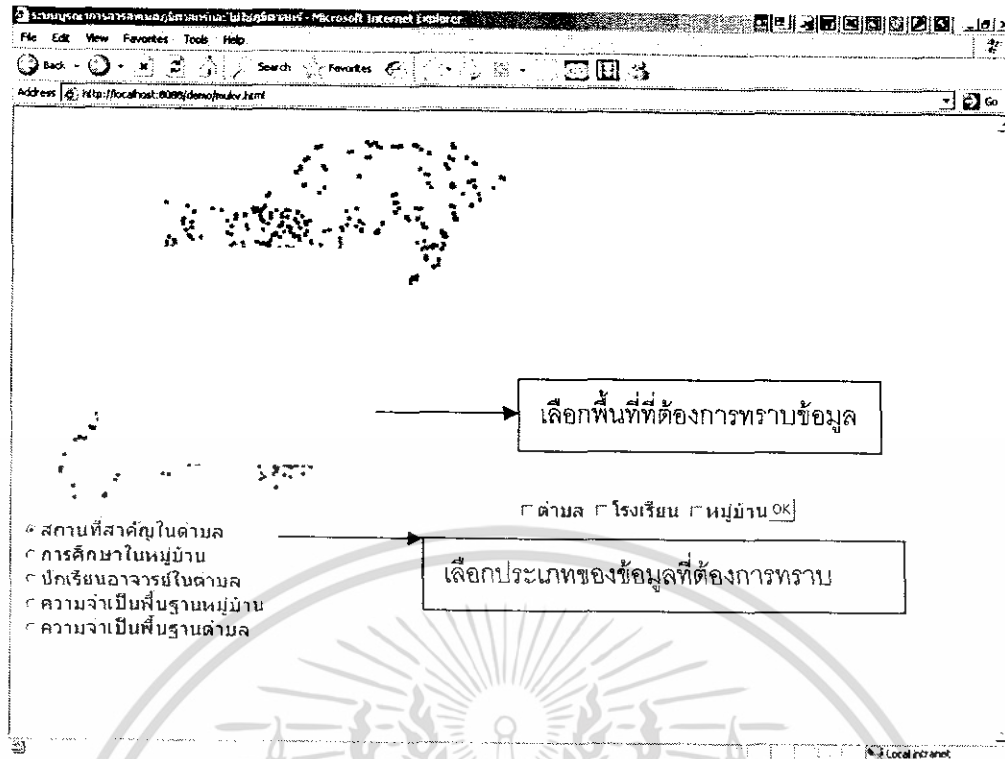
รูปที่ 2.2 Program ที่ได้ทดลอง Implement ในระดับชั้นหมู่บ้าน

หมู่บ้าน	ผู้จบการศึกษาภาคบังคับ	ประชากรทั้งหมด	ร้อยละ
คูตโง้งใหญ่	515	791	65
คันสา	77	282	27
โนนสว่าง	303	411	74
หนองนอก	438	800	55
หนองหอม	866	1258	69
ป่าทราย	211	422	50
คอนม้าย	322	494	65
โคตสูง	269	835	32
บางทรายใหญ่	537	826	65
บางทรายใหญ่	703	1017	69
สงเปลือย	140	238	59
ท่าฝักนอก	117	201	58
คอนม่วงพัฒนา	130	167	78
คำฮี	911	1158	79
นาโสกน้อย	287	360	80
แก่นเต่า	486	673	72
หนองหญ้าไซ	637	893	77
โพนทราย	449	646	70
โพนทราย	160	399	40
ม่วงหัก	656	776	85
คาง	492	698	70
บ้านนันทน์	246	330	75
สนสะอาด	359	443	81
กุดแม่	798	950	84

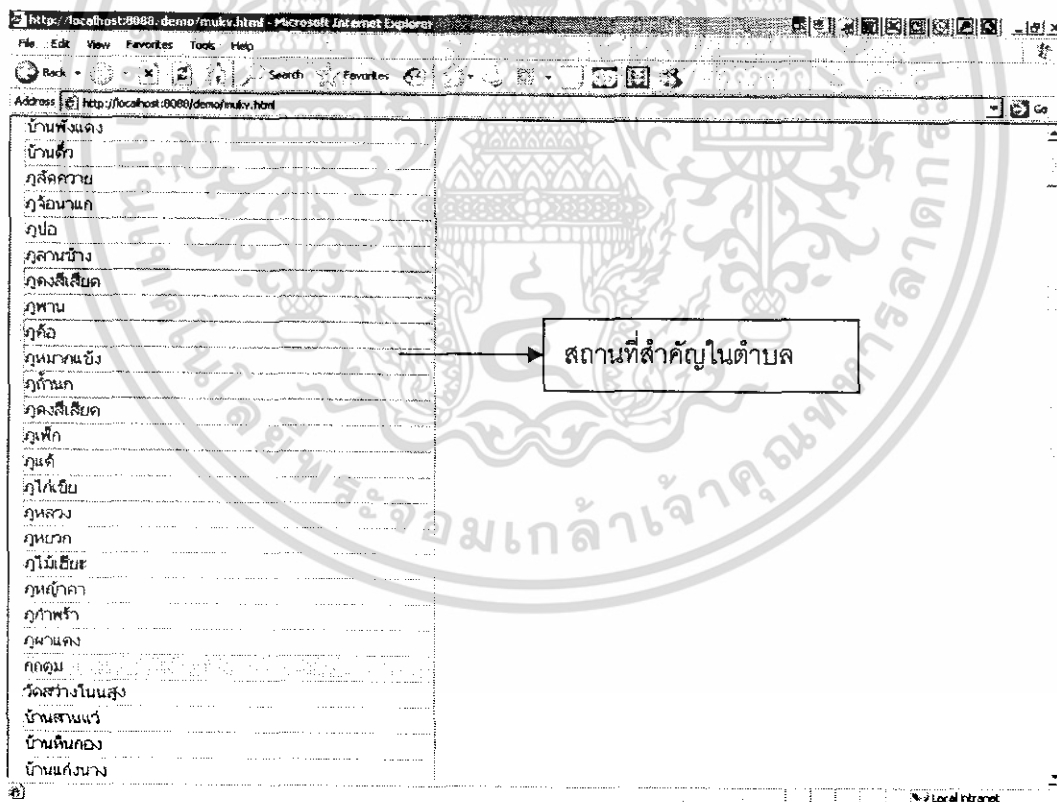
แสดงผลการศึกษาของหมู่บ้าน

รูปที่ 2.3 แสดงรายละเอียดของข้อมูลที่ถูกเลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงการเลือกพื้นที่และรายละเอียด



รูปที่ 2.5 แสดงรายละเอียดข้อมูลที่ถูกเลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีในการเลือกใช้ภาษา GML ในการแสดงผลที่

- ได้แผนที่ที่มีคุณภาพดีขึ้น เนื่องจาก GML จะเปลี่ยนข้อมูลทางภูมิศาสตร์ทั้งหมดให้อยู่ในรูปของ feature หรือ object ซึ่งสามารถจะนำไปแสดงผลในรูปแบบที่ไม่จำกัดความละเอียดในการแสดงผล
- สามารถใช้งานได้บนบราวเซอร์โดยไม่ต้องซื้อโปรแกรมที่ใช้แสดงผลทางฝั่ง Client เมื่อข้อมูล GML ถูกส่งโดยฝั่ง Client จะถูกเปลี่ยนให้อยู่ในรูปแบบของภาพแผนที่บนบราวเซอร์โดยใช้ภาษา SVG ในการแสดงออกมาเป็นภาพ ซึ่งเราสามารถแสดงแผนที่บนบราวเซอร์ โดยไม่จำเป็นต้องติดตั้งซอฟต์แวร์อื่น ๆ เพิ่มเติมบราวเซอร์สามารถรองรับภาพแบบเวกเตอร์ได้ และสามารถดาวน์โหลด SVG ซึ่งเป็นโปรแกรมเสริมของ Adobe Corporation ได้โดยไม่ต้องเสียค่าใช้จ่ายใด ๆ
- สามารถสร้างแผนที่ที่มีรูปแบบเฉพาะได้ เนื่องจาก GML จะเก็บข้อมูลของแผนที่ เช่น ที่ตั้งของ feature นั้น , ลักษณะทางภูมิศาสตร์ , ชนิดของ feature และ attribute ต่าง ๆ โดยไม่กำหนดข้อมูลหรือวิธีการแสดงของข้อมูล แผนที่นั้น ๆ ซึ่งทำให้ผู้ใช้สามารถสร้างแผนที่ให้มีรูปแบบได้ตามที่ต้องการ
- สามารถทำการแก้ไขแผนที่ได้ แผนที่ที่เขียนโดย GML จะถูกดาวน์โหลดและทำการ render บนบราวเซอร์ซึ่งเมื่อ GML ถูกเปลี่ยนเป็น SVG ผู้ใช้จะสามารถใช้โปรแกรมบน Client ในการเพิ่มเติมตัวอักษร หรือทำการเปลี่ยนแปลงภาพบนแผนที่ และสามารถบันทึกเก็บเป็นไฟล์ภาพเพื่อนำไปใช้ในกรณีอื่น เช่น การพิมพ์ออกทางเครื่องพิมพ์หรือทำการส่ง E-mail ได้
- สามารถเชื่อมโยง link ไปลงในแผนที่ได้ ประโยชน์ที่สำคัญของ GML คือผู้ใช้สามารถฝัง URL ลงใน feature ที่มีแผนที่ ความสามารถนี้ทำให้สามารถออกแบบแผนที่ให้มีความแปลกใหม่และทันสมัยยิ่งขึ้น
- ทำการ Query ได้ดียิ่งขึ้น หากผู้ใช้ต้องการที่จะ click ไปบน feature บนแผนที่และค้นหาข้อมูลเพิ่มเติมเกี่ยวกับ feature นั้น ๆ เช่น ชื่อหรือความลึกของแม่น้ำ ซึ่งไม่สามารถทำได้บนแผนที่ที่สร้างขึ้นโดยใช้ไฟล์ภาพ (GIF / JPG) สำหรับแผนที่ที่ใช้ GML นั้นเราจะสามารถระบุ feature ที่ถูก click นั้นได้
- สามารถกำหนดเนื้อหาที่ต้องการแสดงหรือต้องการได้รับเพราะ GML เป็นภาษาที่อ้างอิงถึง feature ทำให้ผู้ใช้สามารถกำหนดได้ว่าต้องการให้มี feature ใดแสดงบนแผนที่บ้าง ซึ่งจะช่วยลดระยะเวลาในการรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ใช้เห็นประโยชน์ด้านการศึกษา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถแสดงผลได้บนทุกอุปกรณ์ที่ใช้เบราว์เซอร์ได้ ข้อมูลทางภูมิศาสตร์ใน GML สามารถส่งไปยังอุปกรณ์ชนิดใดก็ได้ที่สามารถรองรับ XML การแสดงผลแบบ XML ได้เช่น เครื่อง PDA หรือโทรศัพท์มือถือทำให้แผนที่นั้นสามารถใช้งานได้อย่างกว้างขวาง

2.3 ภาษา SVG (Scalable Vector Graphics)

2.3.1 ความรู้เบื้องต้นเกี่ยวกับภาษา SVG

SVG มี XML เป็นพื้นฐาน ที่สร้างขึ้นเพื่อกำหนดลักษณะการแสดงผลในรูปแบบสองมิติโดย SVG จะกำหนดรูปแบบของวัตถุสามอย่างคือ vector graphic shape (เช่น ส่วนของเส้นตรง และเส้นโค้ง), รูปและตัวอักษร โดยสามารถที่จะกำหนด grouped, styled, transformed หรือ composited ให้แก่วัตถุใน SVG ได้ อีกทั้งยังเพิ่มคุณสมบัติ Transformations, clipping paths, alpha masks, filter effects, template object และอีกมากมาย เนื่องจาก SVG เป็น XML ดังนั้นจึงมีการสร้าง DOM สำหรับ SVG โดยอ้างอิงตามมาตรฐานของ W3 ซึ่งทำให้ SVG มีคุณสมบัติ dynamic และ interactive ตามไปด้วย แม้แต่ event handlers อย่าง onmouseover หรือ onclick ก็สามารถกำหนดลงในแต่ละวัตถุของ SVG ได้ด้วยอีกทั้งการอ้างอิงตามมาตรฐานทำให้ SVG สามารถคุยกับ XML อื่นๆ ใน web page เดียวกันได้

SVG มีสองส่วนคือส่วนที่เป็น XML-based file format และส่วน programming API สำหรับ graphical application SVG สร้างขึ้นบนมาตรฐานอื่น ๆ มากมายเช่น XML JPEG PNG DOM SMIL CSS เป็นต้น ความสามารถในการขยาย SVG และ embed metadata มีประโยชน์ในการ mapping community ตัวอย่างเช่น graphical element สามารถระบุ native object เช่นทะเลสาบได้ ขอมให้ application interact กับ object ในลักษณะ graphical

SVG เป็นส่วนประกอบที่สมบูรณ์แบบสำหรับ OpenGIS ที่เป็น GML format GML ก็เป็น XML-based ที่ใช้อธิบาย geographical element เช่น แม่น้ำและถนน GML สามารถ convert เป็น SVG โดยใช้ XML pipeline สำหรับ online display

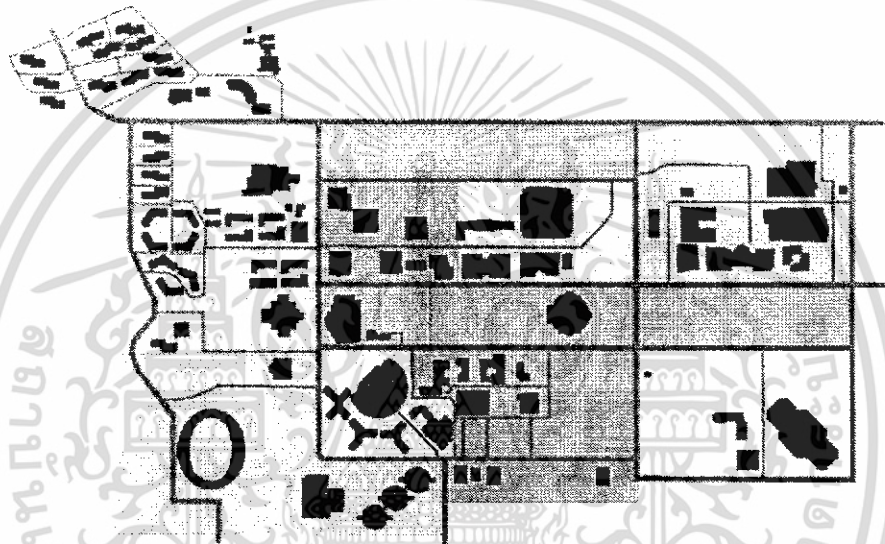
2.3.2 Link ของ SVG

SVG มี 'a' element คล้ายกับ 'a' element ของ HTML ที่ใช้ระบุ link ซึ่ง SVG ใช้ XLink สำหรับทุก link definitions

SVG 1.0 ต้องการแค่ให้ user agents support XLink's notion ของ simple links โดยแต่ละ simple link เกี่ยวข้องกับเฉพาะ 2 resources เท่านั้นคือ 1 local และ 1 remote ด้วย arc ที่ชี้จาก former ไปยัง latter

Simple link หนึ่งนั้นกำหนดสำหรับหนึ่ง render element เท่านั้นดังนั้นถ้า 'a' element มี 3 'circle' element จะต้องมีการสร้าง link สำหรับแต่ละ circle และแต่ละ rendered element ใน 'a' element คือ local resource (source anchor สำหรับ link)

Remote resource (ปลายทางของ link) จะกำหนดโดย URI ซึ่งระบุโดย XLink href attribute บน 'a' element โดย remote resource นั้นอาจเป็น Web resource ใด ๆ ก็ได้ เช่น ภาพ , video clip , sound bite , program , SVG document อื่น , HTML document , element ใน document ปัจจุบัน , element ใน document อื่น เป็นต้น) เมื่อทำการ activate link เหล่านี้ user จะไป visit resource เหล่านั้น



รูปที่ 2.6 ภาพแผนที่ ในรูปแบบ SVG ที่แสดงบน Web Browser

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20001102//EN"
"http://www.w3.org/TR/2000/CR-SVG-20001102/DTD/svg-20001102.dtd">
<svg width="678" height="400" viewBox="672355 -1556851.93 2705 1771" onzoom="zooming0">
  <desc>Thammasat Internet Map</desc>
  <defs>
    <script xlink:href="/data/Thammasat_Internet_Map.js" type="text/javascript">
    </script>
  </defs>

  <g id="1">
    <g id="AsiagameResident" style="fill:#FFC7AE;stroke:#000000;stroke-width:0.50">
```

รูปที่ 2.7 แสดง tag ของ SVG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การติดต่อกับฐานข้อมูล

2.4.1 PostgreSQL

เป็น Object-Relational DBMS โดยสามารถใช้รูปแบบของภาษา SQL ได้เกือบทั้งหมด และสามารถใช่ subselects , transactions , user-defined types และ functions ได้ อีกทั้งยังเป็น Database ซึ่งให้ Source code ฟรี ด้วย

ประวัติการพัฒนา

- เป็น Project ของ Prof. Michael Stonebraker ที่ มหาวิทยาลัย Berkeley
- ซึ่งเดิมพัฒนามาจาก Ingres ซึ่งเป็นที่รู้จักในชื่อ CA-Ingres II ซึ่ง Ingres ใช้ ภาษา query , QUEL เป็นภาษาของตัวเอง ปัจจุบันได้หยุดพัฒนาไปแล้ว แต่ยังสามารถนำมาใช้ได้ฟรี
- ต่อมา Prof. Stonebraker ได้นำมาพัฒนาเป็น Postgres (มาจาก after Ingres) ซึ่งได้ใช้ ภาษา query เป็น POSTQUEL เป็น Postgres version 4.2
- ต่อมาในช่วง ปี 1987 Postgres ได้เสนอ rules , procedures , time travel , extensible types และ object-relational concepts
- Postgres ถูกนำมาใช้ เพื่อการค้า ในชื่อว่า Illustra (ปัจจุบัน ถูก Informix ซื้อไป และรวมเข้าไว้ใน Universal Server)
- ต่อมา นักศึกษาปริญญาเอก 2 คน คือ Andrew Yu และ Jolly Chen ได้พัฒนา Postgres ให้ใช้ภาษา query ตามรูปแบบที่เป็นมาตรฐาน แทนที่ ภาษา POSTQUEL เดิม ซึ่งได้เผยแพร่ในปี 1995 จึงเรียกเป็น Postgres95 หรือ version 5 หลังจากนั้น การพัฒนาต่อ โดย กลุ่มพัฒนาทาง Internet
- ปัจจุบัน เปลี่ยนชื่อเป็น PostgreSQL พัฒนาต่อเนื่องเป็น version 7.1

ลักษณะโครงสร้าง

ระบบที่ใช้ PostgreSQL จะติดตั้ง PostgreSQL ไว้ที่เครื่อง Server ซึ่งเป็นที่เก็บ database ด้วย และยังสามารถ ติดตั้ง PostgreSQL ได้มากกว่า 1 ชุดใน Server เครื่องเดียว

ผู้ดูแลระบบ PostgreSQL จะใช้ชื่อว่า postgres ซึ่งเป็นผู้ดูแลทั้ง ตัวโปรแกรม และ database ซึ่งสามารถทำงานกับบางคำสั่งเฉพาะ เพื่อจัดการ database และ ผู้ใช้บริการ (user) ซึ่งผู้ดูแลระบบ database (postgres) จะคล้ายการทำงานของ superuser ในระบบ Unix หน้าที่ของ postgres สามารถ สร้างชื่อ user และกำหนดสิทธิและระดับการใช้งานต่างๆได้

PostgreSQL ใช้รูปแบบการทำงาน แบบ Client/Server ซึ่งในการทำงานจะประกอบด้วย 3 process ทำงานร่วมกัน คือ

1. Postmaster เป็น supervisory daemon process ซึ่งจัดการติดต่อระหว่าง Frontend กับ Backend process ในการ allocate share buffer , จัดการค่าเริ่มต้นต่างๆในระหว่างเริ่มทำงาน และเก็บบันทึกการเข้าใช้ระบบและความผิดพลาดต่างๆที่เกิดขึ้น

2. Postgres เป็น backend process เพื่อจัดการ database ถือว่าส่วนนี้เป็น process ที่ทำงานจริงๆ เช่น ทำงานตาม query โดย Postmaster จะสั่งให้สร้าง Backend process สำหรับทุกๆ การเชื่อมต่อกับ Frontend ดังนั้น Postgres นี้จะทำงานที่ server

3. Frontend เป็น application ซึ่งจะทำงานที่เครื่อง client และจะส่งคำสั่งการเชื่อมต่อหรือคำสั่งต่างๆ มาที่ Postmaster แล้ว Postmaster จึงส่งต่อการทำงานไปที่ Postgres

2.4.1.1 หลักการทำงานของ PostgreSQL

การทำงานจะแบ่ง process ที่ทำงานดังที่กล่าวมาแล้ว คือ

1. ในส่วนของ Supervisory daemon process คือ Postmaster
2. ในส่วนของ User's Frontend application เช่น โปรแกรม psql หรือ CGI-Perl

3. และในส่วน Backend database servers คือ Postgres

เมื่อโปรแกรม ทาง Frontend ต้องการข้อมูล หรือทำงานกับ database โดยเรียกผ่านทาง library libpq ซึ่ง library libpq นี้ จะส่ง requests ผ่านทาง Network ไปยัง Postmaster เมื่อ Postmaster ได้รับ request ดังกล่าว ทาง Postmaster จะสร้าง Backend process ขึ้นที่ server เพื่อติดต่อกับ Frontend แทน การทำงานนั้นจะเกิดขึ้นระหว่าง Frontend กับ Backend โดยไม่ผ่าน Postmaster อีก และ Postmaster ก็ทำงานต่อไป คือรอรับ request อื่นๆต่อไป

Library libpq จะให้ หนึ่ง Frontend สามารถติดต่อกับได้หลาย Backend processes แต่การทำงานยังเป็นแบบ single threaded เนื่องจาก library libpq ยังไม่สามารถทำ multithreaded ได้

ตามหลักการที่กล่าวมา ดังนั้น Postmaster กับ Backend จะต้องทำงานอยู่ที่เครื่องเดียวกัน คือ database server แต่ Frontend จะทำงานที่เครื่องใดก็ได้

2.4.1.2 ภาษา query ที่ใช้

ภาษา query ที่ PostgreSQL ใช้อิงตามมาตรฐาน SQL3 ซึ่งกำลังจะเป็น มาตรฐานของภาษา Query โดยมีการเพิ่มคุณสมบัติต่างๆ เช่น extensible type system, inheritance, functions และ production rules จากภาษา POSTQUEL เดิม และยังเป็นไปตามมาตรฐาน ANSI SQL ซึ่งประกอบด้วยคำสั่ง คือ

abort transaction, alter table, begin transaction, change acl, close, copy, create aggregate, create database, create function, create index, create operator, create rule, create table, create type, create version, create view, declare, delete, drop aggregate, drop database, drop function, drop index, drop operator, drop rule, drop table, drop type, drop view, end transaction, extend index, fetch, grant, insert, load, purge, revoke, select, update, vacuum

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

expressions สามารถเป็นได้ทั้ง arbitrary arithmetic, logical , functional expressions , aggregates.

2.4.2 PostGIS

PostGIS เป็นการเพิ่มเติมในส่วน of ฐานข้อมูลเชิงวัตถุสัมพันธ์ (Object-relational database system) ของ PostgreSQL ให้มีการรองรับวัตถุทางระบบสารสนเทศภูมิศาสตร์ (GIS object) เข้ามาเก็บไว้ในฐานข้อมูล มีการสนับสนุน GiST indexs และ R-tree indexs ซึ่งเป็นวิธีการค้นข้อมูลแบบดัชนี (Indexing) ที่ใช้ในฐานข้อมูลเชิงพื้นที่สำหรับฐานข้อมูลที่มีขนาดใหญ่ ซึ่ง PostGIS เองมีการกำหนดการใช้งานโดย OpenGIS ที่เป็นลักษณะพื้นฐานของ SQL (SFSQL)

2.4.2.1 ลักษณะทั่วไปของ PostGIS

ดังที่กล่าวมาแล้วว่า PostGIS มีการกำหนดการใช้งานที่เป็นลักษณะพื้นฐานโดย OpenGIS Consortium (OGC) ซึ่งเป็นสถาบันที่ศึกษาเพื่อการสร้างอินเทอร์เน็ตที่ทำโปรแกรมประยุกต์ซอฟต์แวร์ ให้มีการใช้งานได้กับระบบสารสนเทศภูมิศาสตร์ (GIS)

OpenGIS เป็นสิ่งกำหนดความชัดเจนในการเข้าถึงข้อมูลทางด้านพื้นที่ที่แตกต่างกัน และประมวลผลทางพื้นที่ของแหล่งทรัพยากรในสภาพแวดล้อมที่เป็นเครือข่าย (Network)

ซึ่งลักษณะที่มีการกำหนดโดย OpenGIS มีดังนี้

- OpenGIS SFSQL Objects เป็นการกำหนดในส่วน of วัตถุเชิงพื้นที่ เช่น
 - POINT
 - LINESTRING
 - POLYGON
 - MULTIPOINT
 - MULTILINESTRING
 - MULTIPOLYGON
 - GEOMETRYCOLLECTION

- OpenGIS SFSQL Representations เป็นการกำหนดในส่วน of การแสดงผล ซึ่งเป็นมาตรฐานในการแสดงผลของวัตถุเชิงพื้นที่ มี 2 รูปแบบคือ

1). Well – Known Text (WKT) Form

เป็นการแสดงผลในรูปแบบที่เป็น String เช่น

POINT(1 1)

MULTIPOINT(1 1, 3 4, -1 3)

LINESTRING(1 1, 2 2, 3 4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ POLYGON((0 0, 0 1, 1 1, 1 0, 0 0)) ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MULTIPOLYGON((0 0, 0 1, 1 1, 1 0, 0 0), (5 5, 5 6, 6 6, 6 5, 5 5))

MULTILINESTRING((1 1, 2 2, 3 4),(2 2, 3 3, 4 5))

2). Well – Known Binary (WKB) Form

เป็นการแสดงผลในรูปแบบที่เกี่ยวกับบิต ซึ่งจะดึงข้อมูลออกจากฐานข้อมูล โดยไม่มีการเปลี่ยนไปแสดงในรูปแบบที่เป็น String

2.4.2.2 การใช้รูปแบบมาตรฐาน OpenGIS

OpenGIS เป็นการระบุลักษณะพื้นฐานสำหรับ SQL ซึ่งจะกำหนดมาตรฐานระบบสารสนเทศภูมิศาสตร์ในส่วนที่เป็นชนิดของวัตถุ ฟังก์ชันที่ใช้ในการจัดการ และกลุ่มของตาราง

ตารางของ OpenGIS ที่ใช้มี 2 ตาราง ดังนี้

1) The SPATIAL_REF_SYS Table มีรูปแบบในการกำหนดดังต่อไปนี้

```
CREATE TABLE SPATIAL_REF_SYS (
  SRID          INTEGER NOT NULL PRIMARY KEY,
  AUTH_NAME     VARCHAR(256),
  AUTH_SRID     INTEGER,
  SRTEXT        VARCHAR(2048),
  PROJ4TEXT     VARCHAR(2048)
)
```

SRID

ตัวเลขที่ใช้ในการระบุระบบการอ้างอิงเชิงพื้นที่ภายในฐานข้อมูล

AUTH_NAME

ชื่อมาตรฐาน หรือ ส่วนที่เป็นมาตรฐาน สำหรับระบบการอ้างอิง

AUTH_SRID

หมายเลขของระบบการอ้างอิงเชิงพื้นที่ ซึ่งมีการกำหนดโดยการอ้างอิง

จาก AUTH_NAME

SRTEXT

เป็นการแสดงผลแบบ Well – known Text ของระบบการอ้างอิงเชิง

พื้นที่

PROJ4TEXT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับเป็น Proj4 library ซึ่งใช้กำหนด String ให้กับ SRID ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) The GEOMETRY_COLUMNS Table มีรูปแบบในการกำหนดดังต่อไปนี้

```
CREATE TABLE GEOMETRY_COLUMNS (
    F_TABLE_CATALOG    VARCHAR(256) NOT NULL,
    F_TABLE_SCHEMA     VARCHAR(256) NOT NULL,
    F_TABLE_NAME       VARCHAR(256) NOT NULL,
    F_GEOMETRY_COLUMN  VARCHAR(256) NOT NULL,
    COORD_DIMENSION    INTEGER NOT NULL,
    SRID               INTEGER NOT NULL,
    TYPE               VARCHAR(30) NOT NULL
)
```

F_TABLE_CATALOG, F_TABLE_SCHEMA, F_TABLE_NAME

ลักษณะพิเศษของชื่อตารางที่ประกอบด้วยคอลัมน์เชิงเรขาคณิต

F_GEOMETRY_COLUMN

ชื่อของคอลัมน์ทางเรขาคณิต ซึ่งมีลักษณะเป็นตาราง

COORD_DIMENSION

มิติเชิงพื้นที่ของคอลัมน์

SRID

หมายเลขของระบบอ้างอิงเชิงพื้นที่ ใช้สำหรับเรขาคณิตในตาราง ซึ่ง

เป็นคีย์นอกที่อ้างอิงถึง SPATIAL_REF_SYS

TYPE

ชนิดของวัตถุเชิงพื้นที่

2.4.2.3 การสร้าง Spatial table

การนำข้อมูล spatial มาสร้างลง table นั้นมีวิธีทำ 2 ขั้นตอน

1). สร้าง table ธรรมดาของ non-spatial ตัวอย่างเช่น

```
CREATE TABLE ROADS_GEOM ( ID int4, NAME varchar(25) )
```

2). เพิ่ม column ใน table ของ spatial โดยใช้ function OpenGIS

“AddGeometryColumn” ซึ่งจะมี syntax ดังตัวอย่างต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของ AddGeometryColumn(<schema_name>, <table_name>, ... นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<column_name>, <srid>, <type>,
<dimension>)

Or, using current schema:

AddGeometryColumn(<table_name>,
<column_name>, <srid>, <type>,
<dimension>)

2.5 ShapeFile

2.5.1 ทำไมต้องเป็น Shapefiles

Shapefile จะมีการเก็บแบบเรขาคณิตไม่คงที่ (Nontopological geometry) และ สำหรับ ลักษณะจุดเด่นในชุดข้อมูล ลักษณะเด่นของการเก็บลักษณะของข้อมูลจะเก็บเป็นแบบชุดของ ทิศทางพิกัด (Vector coordinates) เพราะ Shapefile จะไม่มีการประมวลผลส่วนเกินที่จำเป็นของ โครงสร้างข้อมูลรูปแบบเรขาคณิต ข้อดีของ Shapefile จะมีความสามารถตรงที่ความเร็วในการ วาดและการแก้ไขจะทำให้เร็วกว่าข้อมูลอื่น ๆ จุดเด่นอีกอย่างของ Shapefile จะสามารถ ซ้อนทับกัน หรือสามารถแยกเป็นชั้นได้ และงานต่อการอ่านและเขียนบน disk ที่มีพื้นที่น้อย

Shapefiles สามารถรองรับ จุด , เส้น และพื้นที่ลักษณะพิเศษต่าง ๆ ได้ โดยพื้นที่พิเศษนี้ อาจจะเป็นแบบ Polygon , Polyline ก็ได้ Attribute ถือว่าอยู่ในรูปแบบของ dBASE ซึ่งแต่ละ Attribute จะมีความสัมพันธ์เป็นแบบ หนึ่งต่อหนึ่ง เกี่ยวข้องกับการกำหนดรูปแบบที่บันทึก เอาไว้

สามารถสร้าง Shapefile ได้ 4 วิธีด้วยกันดังนี้

1. การนำออก (Export) Shapefile สามารถสร้างโดยการนำข้อมูลจาก source มาทำเป็น Shapefile โดยการใช้ ARC/INFO , PC ARC/INFO , Spatial Database Engine (SDE) , ArcView GIS หรือ BusinessMAP software
2. การเปลี่ยนให้เป็นตัวเลขโดยใช้เครื่องคอมพิวเตอร์ (Digitize) ซึ่ง Shapefiles สามารถสร้างได้โดยตรงโดยการเปลี่ยนรูปร่างให้เป็นตัวเลข โดยใช้ tool ในการ สร้างของ ArcView GIS
3. การเขียนโปรแกรม (Programming) โดยใช้ Avenue™ (ArcView GIS) , MapObjects™ , ARC Macro Language (AML™) (ARC/INFO) , หรือ Simple Macro Language (SML™) (PC ARC/INFO) software เราสามารถสร้าง Shapefile ภายในโปรแกรมเหล่านี้ได้
4. การเขียนโปรแกรมให้สร้าง Shapefile ได้โดยตรง

2.5.2 Shapefile Technical Description

ESRI shapefile จะประกอบไปด้วย main file , index file และตาราง dBASE โดย main file จะเข้าถึงโดยตรง variable-record-length file ในแต่ละ record จะบรรยายถึงรูปร่างร่วมกับรูปทรงเรขาคณิตต่าง ๆ ส่วน index file ในแต่ละ record จะประกอบไปด้วย offset ที่คล้ายกันกับ record ของ mainfile จากส่วนแรกของ main file ส่วนตาราง dBASE จะประกอบด้วย attribute ร่วมกับหนึ่ง record ต่อ feature ความสัมพันธ์แบบ one-to-one ระหว่าง geometry และ attribute เป็นแบบ base on record number โดย attribute record ใน dBASE file จะคล้ายกันกับ record ของ main file

2.5.3 Naming Convention

ชื่อทุก file ให้การสนับสนุน 8.3 naming convention โดย main file , index file และ dBASE file มีส่วนที่เติมหน้าคำเหมือนกัน ซึ่งส่วนที่เติมหน้าคำจำเป็นอย่างยิ่งร่วมกับที่ใช้ตัวหนังสือและตัวเลขเป็นสัญลักษณ์ (a-Z , 0-9) ติดตามโดยศูนย์ หรือมากกว่า 7 ตัวอักษร (a-Z , 0-9 , _ , -) ส่วนต่อท้ายคำสำหรับ main file จะเป็น .shp ส่วนต่อท้ายคำของ index file จะเป็น .shx สำหรับส่วนต่อท้ายคำของตาราง dBASE จะเป็น .dbf ชื่อของ file ทุก ๆ ตัวจะเป็นไปตามระบบปฏิบัติการ

Examples

Main file:	counties.shp
Index file:	counties.shx
dBASE table:	counties.dbf

2.5.4 Numeric Type

เกี่ยวกับชนิดของตัวเลข Shapefile จะเก็บเป็นแบบ integer และ double-precision โดยมันจะจำชนิดตามแบบเอกสารที่มันอ้างอิงดังนี้

Integer : Signed 32-bit integer (4 byte)

Double : Signed 64-bit IEEE double-precision floating point number (8 byte)

การแทนตัวเลขด้วยจุดทศนิยม (floating point) ทั้งฝั่งบวก , ฝั่งลบ และที่ไม่เป็นตัวเลขใน Shapefile จะไม่ยินยอม อย่างไรก็ตาม Shapefile จะรองรับแนวความคิดของ “no data” แต่ในปัจจุบันจะใช้สำหรับเปรียบเทียบเพียงอย่างเดียว ทุกการแทนตัวเลขด้วยจุดทศนิยม (floating point) จะน้อยกว่า -10^{38} เป็นการไต่ตรงโดย Shapefile reader แทนค่าของ “no data”

2.5.5 องค์ประกอบของ Main File

ส่วนประกอบ main file จะจำกัดความยาว (fixed-length) file ส่วนหัวโดยการบันทึก variable-length record แต่ละ variable-length record เป็นการสร้างขึ้นของ fixed-length record header ตามส่วนประกอบของ variable-length record ในรูปที่ 1 เป็นการแสดงองค์ประกอบของ main file

File Header	
Record Header	Record Contents
Record Header	Record Contents
Record Header	Record Contents
Record Header	Record Contents
...	
...	
Record Header	Record Contents

รูปที่ 2.8 องค์ประกอบของ main file

2.5.5.1 Byte Order

ทุกส่วนประกอบใน Shapefile สามารถแบ่งออกได้เป็น 2 ส่วนประกอบ ดังนี้

1. ข้อมูลที่สัมพันธ์กัน (Data related)
 - Main file record contents
 - Main file header's data description fields (Shape Type, Bounding Box, etc.)
2. การจัดการความสัมพันธ์ file (File management related)
 - File and record lengths
 - Record offsets, and so on

integers และ double-precision integers นั้นแทนการบรรยาย field ใน file ส่วนหัว (identified below) และบันทึกลงใน main file เป็นแบบ little endian ส่วน integers and double-precision floating point numbers จะแทนส่วนที่เหลือของ file และจะจัดการกับ file เป็นแบบ big endian

2.5.5.2 The Main File Header

ส่วนหัวของ main file จะมีความยาว 100 byte ในตารางที่ 1 จะแสดง fields ใน file header ร่วมกับตำแหน่งใน byte ต่าง ๆ ค่า (value) , ชนิด (type) และ byte order ในตารางนี้จะแสดงตำแหน่งความสัมพันธ์เริ่มต้นของ file

Position	Field	Value	Type	Byte Order
Byte 0	File Code	9994	Integer	Big
Byte 4	Unused	0	Integer	Big
Byte 8	Unused	0	Integer	Big
Byte 12	Unused	0	Integer	Big
Byte 16	Unused	0	Integer	Big
Byte 20	Unused	0	Integer	Big
Byte 24	File Length	File Length	Integer	Big
Byte 28	Version	1000	Integer	Little
Byte 32	Shape Type	Shape Type	Integer	Little
Byte 36	Bounding Box	Xmin	Double	Little
Byte 44	Bounding Box	Ymin	Double	Little
Byte 52	Bounding Box	Xmax	Double	Little
Byte 60	Bounding Box	Ymax	Double	Little
Byte 68*	Bounding Box	Zmin	Double	Little
Byte 76*	Bounding Box	Zmax	Double	Little
Byte 84*	Bounding Box	Mmin	Double	Little
Byte 92*	Bounding Box	Mmax	Double	Little

* Unused, with value 0.0, if not Measured or Z type

ตารางที่ 2.1 แสดงรายละเอียดของ main file header

ค่าสำหรับความยาวของ file เป็นผลรวมของความยาว file เป็น 16-bit words ทุกตำแหน่งที่ไม่เป็น null ใน Shapefile เป็นการร้องขอในขนาดของชนิดรูปร่างที่เหมือนกัน ค่าสำหรับชนิดรูปร่างจะเป็นดังต่อไปนี้

Value	Shape Type
0	Null Shape
1	Point
3	PolyLine
5	Polygon
8	MultiPoint
11	PointZ
13	PolyLineZ
15	PolygonZ
18	MultiPointZ
21	PointM
23	PolyLineM
25	PolygonM
28	MultiPointM
31	MultiPatch

ตารางที่ 2.2 แสดงค่าประจำตัวต่างๆ ใน Shape Type

Shape type ที่นอกเหนือจากนี้ (2, 4, 6, ect, และมากกว่า 33) เป็นตัวที่สำรองไว้ใช้ในอนาคต ซึ่งในปัจจุบัน Shape file ถูกจำกัดให้เหมือนกับชนิดของ shape นอกเหนือจากตัวพิเศษ ในอนาคต shapefile อาจจะถูกประกอบด้วยมากกว่าหนึ่งชนิด shape ถ้ารวมชนิดของ shape เป็นการดำเนินการ field shape type ในส่วนหัวต้องการตัวบ่งชี้ file ที่เป็นอะไรก็ตาม

Bounding Box ในส่วนหัวของ main file จะเก็บขนาดที่เป็นอยู่จริงของ shape ใน file ซึ่ง Bounding ที่น้อยที่สุดของรูปสี่เหลี่ยมมุมฉากเส้นตั้งฉากจะเป็นแกน X และ Y (และที่เป็นไปได้เป็น M และ Z) ทุกแกนที่ประกอบเป็นรูปร่าง ถ้า shapefile วางเปล่า (ไม่มี record) ค่าสำหรับ Xmin, Ymin, Xmax และ Ymax คือไม่ได้ระบุไว้ ส่วน Mmin และ Mmax สามารถจะประกอบไปด้วย ค่าของ “no data” สำหรับ shapefile ของการควบคุมโดย shape type นั้นจะไม่ถูกควบคุม

2.5.5.3 Record Header

สำหรับส่วนหัวแต่ละ record จะเก็บตัวเลขและ ขนาดความยาวสำหรับ record ส่วนหัวของ record จะมีการกำหนดขนาดเป็น 8 byte ในตารางที่ 3 จะแสดงความสัมพันธ์ของ file header กับตำแหน่ง byte, value, type และ byte order ในตารางจะแสดงตำแหน่งความสัมพันธ์ของ record เริ่มต้น

Position	Field	Value	Type	Byte Order
Byte 0	Record Number	Record Number	Integer	Big
Byte 4	Content Length	Content Length	Integer	Big

ตารางที่ 2.3 แสดงรายละเอียดของ Main File Record Headers

ตัวเลขในการ record จะเริ่มต้นที่เลข 1 ขนาดความยาว record มีขนาดเป็นความยาวของขนาด record ที่ตัดออกเป็น 16-bit words ซึ่งแต่ละ record ดังนั้นให้เป็น (4+ขนาดความยาว) 16-bit words ไปยังผลรวมขนาดของ file ที่เก็บที่ 24 byte ในส่วนหัว

2.5.5.4 Main File Record Content

Shapefile จะเก็บขนาดที่ประกอบไปด้วย shape type ตามด้วยข้อมูล geometric สำหรับรูปร่าง ขนาดของความจุ record ขึ้นอยู่กับตัวเลขของเส้นแบ่ง และจุดตัดของเส้นใน shape สำหรับแต่ละชนิดของรูปร่างเราจะอธิบายรูปร่างในส่วนที่เป็น mapping ส่วนประกอบ record บน disk ในตารางที่ 4 จาก 16 ตำแหน่งร่วมกับความสัมพันธ์ของส่วนประกอบ record

2.5.5.5 Null Shapes

ชนิดของรูปร่างของ 0 แสดงเป็น null shape คือไม่มีข้อมูล geometric สำหรับรูปร่าง แต่ละลักษณะชนิด (point , line , polygon , ect) จะรองรับ nulls มันเป็นการพบว่ามีจุดและ ไม่มีจุดที่เหมือนกับ shapefile ถ้ามี null shape บ่อย ๆ เราจะทำการสร้าง shapefile และร่วมกับข้อมูล geometry ก่อนหลังจากที่เราสร้าง

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	0	Integer	1	Little

ตารางที่ 2.4 แสดงความจุ Null Shape Record

2.5.5.6 Shape Type in X , Y Space

Point จะประกอบไปด้วยคู่ของ double-precision พิกัดใน X , Y

Point

{

Double X //X coordinate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาดูเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Double Y //Y coordinate
}

```

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	1	Integer	1	Little
Byte 4	X	X	Double	1	Little
Byte 12	Y	Y	Double	1	Little

ตารางที่ 2.5 Point Record Contents

Multipoint multipoint จะถูกแทนด้วย set ของจุดดังข้างล่างนี้

```

MultiPoint
{
    Double[4] Box // Bounding Box
    Integer NumPoints // Number of Points
    Point[NumPoints] Points // The Points in the Set
}

```

Bounding Box จะเก็บ Xmin , Ymin , Xmax , Ymax

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	S	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumPoints	NumPoints	Integer	1	Little
Byte 40	Points	Points	Point	NumPoints	Little

ตารางที่ 2.6 Multipoint Record Contents

PolyLine polyline เป็นชุดของจุดตัดของเส้นที่ประกอบไปด้วย หนึ่งหรือมากกว่า หนึ่งเส้น เส้นเป็นการติดต่อลำดับของจุดสองจุด หรือมากกว่านั้น เพราะรายละเอียดไม่อนุญาต ให้เป็นจุดที่เรียงต่อเนื่องกัน ร่วมกันเหมือนเป็นพิกัด ตัวอ่าน shapefile

```

PolyLine
{

```

```

    Double[4] Box // Bounding Box

```

```

    Integer NumParts // Number of Parts

```

```

    Integer NumPoints // Total Number of Points

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะในวงวิชาการเท่านั้น ไม่ควรเผยแพร่ให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Integer[NumParts] Parts // Index to First Point in Part

Point[NumPoints] Points // Points for All Parts

}

สำหรับแต่ละ field ของ polyline จะอธิบายรายละเอียดดังนี้

Box Bounding Box สำหรับ PolyLine จะเก็บค่าของ Xmin, Ymin, Xmax, Ymax

NumParts เป็นตัวเลขของเส้นใน PolyLine

Numpoints เป็นตัวเลขผลรวมของจุดสำหรับทุก ๆ เส้น

Parts เป็น array ของความยาว NumParts จะเก็บสำหรับแต่ละ PolyLine โดย index ของจุดแรกในจุด array ส่วน array indexes ร่วมกับความสัมพันธ์ ศูนย์

Point เป็น array ของความยาว NumPoint จุดสำหรับแต่ละเส้นใน PolyLine เป็นการเก็บแบบ end to end จุดสำหรับ Part 2 ตามด้วยจุดสำหรับ Part 1 และที่เหลือ

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	3	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little

Note: $X = 44 + 4 * NumParts$

ตารางที่ 2.7 PolyLine Record Contents

Polygon polygon จะประกอบไปด้วย หนึ่งหรือมากกว่าหนึ่งวงแหวนวงแหวนเป็นลำดับที่ต่อกันไปเรื่อย ๆ สี่จุดหรือมากกว่าสี่จุดในรูปแบบปิด Polygon อาจประกอบด้วยวงแหวนด้านนอกหลายวง ส่วนจุดตัดหรือการกำหนดทิศทางสำหรับวงแหวนแสดงส่วนไหนของวงแหวนเป็นภายในของ polygon บริเวณใกล้เคียง อาจจะมีหลาย polygon ได้

โครงสร้างของ polygon เป็นเหมือนกันทุกประการกับโครงสร้างของ PolyLine ดังข้างล่างนี้

Polygon

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Double[4] Box // Bounding Box

Integer NumParts // Number of Parts

Integer NumPoints // Total Number of Points

Integer[NumParts] Parts // Index to First Point in Part

Point[NumPoints] Points // Points for All Parts

}

Box Bounding Box สำหรับ Polygon จะเก็บค่าของ Xmin, Ymin, Xmax, Ymax

NumParts เป็นตัวเลขของวงแหวนใน Polygon

Numpoints เป็นตัวเลขผลรวมของจุดสำหรับทุก ๆ วงแหวน

Parts เป็น array ของความยาว NumParts จะเก็บสำหรับแต่ละวงแหวน โดย index ของจุดแรกในจุด array ส่วน array indexes ร่วมกับความสัมพันธ์ ศูนย์

Point เป็น array ของความยาว NumPoint จุดสำหรับแต่ละวงแหวนใน Polygon เป็นการเก็บแบบ end to end วงแหวนสำหรับ Ring 2 ตามด้วยวงแหวนสำหรับ Ring 1 และที่เหลือ

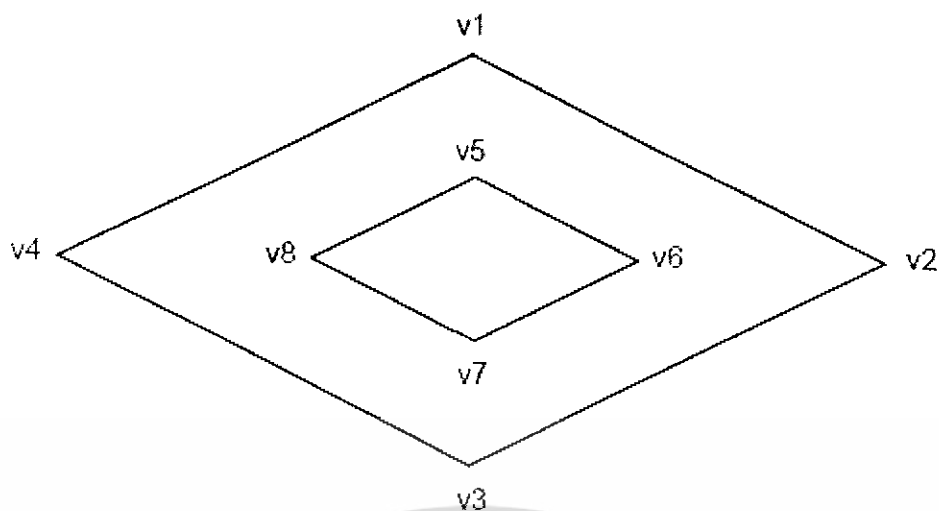
ในรูปที่ 2.9 จะแสดงให้เห็น ถึงแนวคิดกว้าง ๆ การแทนที่ของ Polygon รูปที่แสดงนี้ polygon กับหนึ่งช่อง และผลรวมของจุดตัดทั้ง 8

ข้างล่างนี้เป็นส่วนสำคัญที่เก็บรูปร่างของ Polygon

- วงแหวนปิด
- วงแหวนในจุด array ไม่เป็นที่น่าสนใจ
- Polygon จะเก็บใน shapefile เป็นสิ่งที่สำคัญอย่างดี

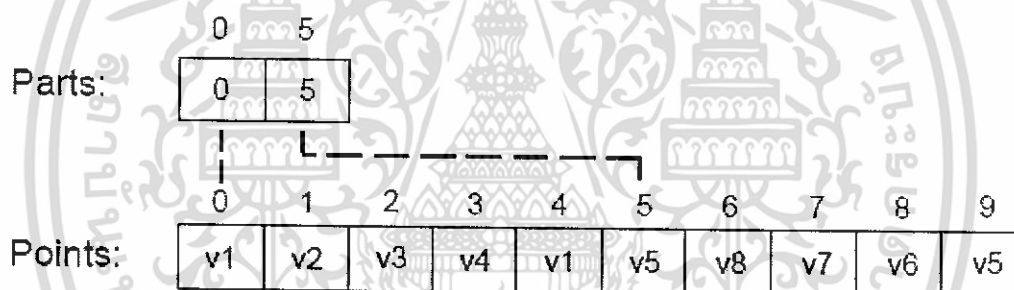
1. ไม่มีจุดตัดของตัวเอง หมายถึงอาจจะมีเพียงหนึ่งวงหรือ อาจจะซ้อนทับกับหลายวงก็ได้ วงแหวน polygon สามารถสัมผัสแต่ละจุดตัดได้ แต่ไม่เป็นไปในทิศทางเดียวกัน

2. Polygon มีภายในที่ถูกต้องเป็นเส้นที่กำหนดขึ้นมา



รูปที่ 2.9 แสดงตัวอย่างของรูป Polygon

สำหรับตัวอย่างนี้ NumParts เท่ากับ 2 และ NumPoint เท่ากับ 10 ข้อสังเกตของจุดสำหรับรูปร่างของ Polygon เป็นดังต่อไปนี้



รูปที่ 2.10 แสดงการเก็บของ Parts และ Points

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	5	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little

Note: $X = 44 + 4 * \text{NumParts}$

ตารางที่ 2.8 Polygon Record Contents

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.5.7 Measured Shape Type in X,Y Space

ชนิดของรูปร่างนี้มีการเพิ่มฟิลด์ M ข้อสังเกตที่ “no data” ถ้าของมันสามารถระบุค่าของ M (ให้ดูที่ Numeric Type)

PointM pointM จะประกอบไปด้วยกลุ่มของฟิลด์ double-precision ใน X , Y บวกการวัด M

```
PointM
{
    Double X // X coordinate
    Double Y // Y coordinate
    Double M // Measure
}
```

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	21	Integer	1	Little
Byte 4	X	X	Double	1	Little
Byte 12	Y	Y	Double	1	Little
Byte 20	M	M	Double	1	Little

ตารางที่ 2.9 PointM Record Contents

MultipointM multipointM แทนด้วยเซตของ PointM ดังนี้

```
MultiPointM
{
    Double[4] Box // Bounding Box
    Integer NumPoints // Number of Points
    Point[NumPoints] Points // The Points in the Set
    Double[2] M Range // Bounding Measure Range
    Double[NumPoints] M Array // Measures
}
```

สำหรับ field ของ multipointM จะเป็นดังนี้

Box Bounding Box สำหรับ MultipointM จะเก็บค่าของ Xmin, Ymin, Xmax, Ymax

NumParts เป็นตัวเลขของ Points

Numpoints เป็น array ของ point ของความยาว NumPoint

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

M Range	ปริมาณที่น้อยสุดและมากที่สุด สำหรับ MultipointM ซึ่งจะเก็บใน Mmin , Mmax
M Array	เป็น array ของปริมาณของความยาว NumPoints

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	28	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumPoints	NumPoints	Integer	1	Little
Byte 40	Points	Points	Point	NumPoints	Little
Byte X*	Mmin	Mmin	Double	1	Little
Byte X+8*	Mmax	Mmax	Double	1	Little
Byte X+16*	Marray	Marray	Double	NumPoints	Little

Note: X = 40 + (16 * NumPoints)
* optional

ตารางที่ 2.10 MultiPointM Record Contents

PolyLineM shapefile ของ PolyLineM จะประกอบไปด้วยเส้นหนึ่งหรือมากกว่าหนึ่งเส้น ซึ่งเส้นนั้นจะเรียงติดต่อกันของจุดสองจุดหรือมากกว่า

PolyLineM

{

Double[4] Box // Bounding Box

Integer NumParts // Number of Parts

Integer NumPoints // Total Number of Points

Integer[NumParts] Parts // Index to First Point in Part

Point[NumPoints] Points // Points for All Parts

Double[2] M Range // Bounding Measure Range

Double[NumPoints] M Array // Measures for All Points

}

สำหรับ field ของ PolyLineM จะเป็นดังนี้

Box Bounding Box สำหรับ PolyLineM จะเก็บค่าของ Xmin, Ymin, Xmax, Ymax

NumParts เป็นตัวเลขของเส้นใน PolyLineM

Numpoints เป็นผลรวมของ point สำหรับทุกเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Parts เป็น array ของความยาว NumParts จะเก็บสำหรับแต่ละเส้น โดย index ของจุดแรกใน array ส่วน array indexes รวมกับความสัมพันธ์เป็นศูนย์
- Point เป็น array ของความยาว NumPoint จุดสำหรับแต่ละเส้นใน PolyLineM เป็นการเก็บแบบ end to end Points สำหรับ parts 2 ตามด้วย point สำหรับ part 1 และที่เหลือ
- M Range ปริมาณที่น้อยสุดและมากที่สุด สำหรับ PolyLineM ซึ่งจะเก็บใน Mmin , Mmax
- M Array เป็น array ของความยาว NumPoints ปริมาณของแต่ละเส้นใน PolyLineM เป็นการเก็บแบบ end to end Parts สำหรับ parts 2 ตามด้วย point สำหรับ part 1 และที่เหลือ

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	23	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little
Byte Y*	Mmin	Mmin	Double	1	Little
Byte Y + 8*	Mmax	Mmax	Double	1	Little
Byte Y + 16*	Marray	Marray	Double	NumPoints	Little

Note: $X = 44 + (4 * \text{NumParts})$, $Y = X + (16 * \text{NumPoints})$
 * optional

ตารางที่ 2.11 PolyLineM Record Contents

PolygonM PolygonM จะประกอบไปด้วยตัวเลขของวงแหวน เป็นรูปแบบวงแหวนปิด ข้อสังเกตการคำนวณในพื้นที่ X, Y ไม่ใช่ในพื้นที่ X, Y, M ซึ่ง PolygonM จะประกอบไปด้วยวงแหวนภายนอกหลาย ๆ วง ซึ่งวงแหวนของ PolygonM เป็นการอ้างอิงถึงเส้นของมันเอง

โครงสร้างของ PolygonM เป็นเหมือนกันทุกประการกับโครงสร้าง PolyLineM

PolygonM

{

Double[4] Box // Bounding Box

Integer NumParts // Number of Parts

Integer NumPoints // Total Number of Points

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น เมื่อผู้ยูเซ่เห็นหน้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Integer[NumParts] Parts // Index to First Point in Part
Point[NumPoints] Points // Points for All Parts
Double[2] M Range // Bounding Measure Range
Double[NumPoints] M Array // Measures for All Points
}

```

สำหรับ field ของ PolygonM จะเป็นดังนี้

Box Bounding Box สำหรับ PolygonM จะเก็บค่าของ Xmin, Ymin, Xmax, Ymax

NumParts เป็นตัวเลขของเส้นใน PolygonM

Numpoints เป็นผลรวมของ point สำหรับทุกวงแหวน

Parts เป็น array ของความยาว NumParts จะเก็บสำหรับแต่ละเส้น โดย index ของจุดแรกใน array ส่วน array indexes ร่วมกับความสัมพันธ์เป็นศูนย์

Point เป็น array ของความยาว NumPoint จุดสำหรับแต่ละเส้นใน PolyLineM เป็นการเก็บแบบ end to end Points สำหรับ Ring 2 ตามด้วย point สำหรับ Ring 1 และที่เหลือ

M Range ปริมาณที่น้อยสุดและมากที่สุด สำหรับ PolyLineM ซึ่งจะเก็บใน Mmin , Mmax

M Array เป็น array ของความยาว NumPoints ปริมาณของแต่ละเส้นใน PolyLineM เป็นการเก็บแบบ end to end Parts สำหรับปริมาณ Ring 2 ตามด้วยปริมาณสำหรับ Ring 1 และที่เหลือ

ข้างล่างนี้เป็นส่วนสำคัญที่เก็บรูปร่างของ Polygon

- วงแหวนปิด
- วงแหวนในจุด array ไม่เป็นที่น่าสนใจ

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	25	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little
Byte Y*	Mmin	Mmin	Double	1	Little
Byte Y + 8*	Mmax	Mmax	Double	1	Little
Byte Y + 16*	Marray	Marray	Double	NumPoints	Little

Note: $X = 44 + (4 * NumParts)$, $Y = X + (16 * NumPoints)$
 * optional

ตารางที่ 2.12 PolygonM Record Contents

2.5.5.8 Shape Types in X,Y,Z Space รูปร่างของชนิดนี้มีชนิดเป็นพิกัดที่เลือกได้ M ข้อสังเกตคือ “no data” ค่าของมันสามารถระบุค่าได้ สำหรับ M (ดูในเรื่อง Numeric Type)

PointZ pointZ จะประกอบไปด้วย 3 ตัวของพิกัด double-precision ใน X, Y, Z บวกปริมาณ

```
PointZ
{
  Double X // X coordinate
  Double Y // Y coordinate
  Double Z // Z coordinate
  Double M // Measure
}
```

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	11	Integer	1	Little
Byte 4	X	X	Double	1	Little
Byte 12	Y	Y	Double	1	Little
Byte 20	Z	Z	Double	1	Little
Byte 28	Measure	M	Double	1	Little

ตารางที่ 2.13 PointZ Record Contents

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MultiPointZ ถูกแทนด้วยเซตของ PointZ ดังข้างล่างนี้

MultiPointZ

```
{
    Double[4] Box // Bounding Box
    Integer NumPoints // Number of Points
    Point[NumPoints] Points // The Points in the Set
    Double[2] Z Range // Bounding Z Range
    Double[NumPoints] Z Array // Z Values
    Double[2] M Range // Bounding Measure Range
    Double[NumPoints] M Array // Measures
}
```

- Bounding Box สำหรับ PolygonM จะเก็บค่าของ Xmin , Ymin , Xmax , Ymax
- Bounding Z Range จะเก็บค่าของ Zmin , Zmax ส่วน Bounding M Range จะเก็บค่าของ Mmin , Mmax

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	18	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumPoints	NumPoints	Integer	1	Little
Byte 40	Points	Points	Point	NumPoints	Little
Byte X	Zmin	Zmin	Double	1	Little
Byte X+8	Zmax	Zmax	Double	1	Little
Byte X+16	Zarray	Zarray	Double	NumPoints	Little
Byte Y*	Mmin	Mmin	Double	1	Little
Byte Y+8*	Mmax	Mmax	Double	1	Little
Byte Y+16*	Marray	Marray	Double	NumPoints	Little

Note: X = 40 + (16 * NumPoints); Y = X + 16 + (8 * NumPoints)
* optional

ตารางที่ 2.14 MultiPointZ Record Contents

PolyLineZ PolyLineZ จะประกอบไปด้วยเส้นหนึ่งหรือมากกว่าหนึ่งเส้น ซึ่งเส้นนั้นจะเรียงติดต่อกันของจุดสองจุดหรือมากกว่าสองจุด

PolyLineZ

```
{
    Double[4] Box // Bounding Box
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Integer NumParts // Number of Parts
Integer NumPoints // Total Number of Points
Integer[NumParts] Parts // Index to First Point in Part
Point[NumPoints] Points // Points for All Parts
Double[2] Z Range // Bounding Z Range
Double[NumPoints] Z Array // Z Values for All Points
Double[2] M Range // Bounding Measure Range
Double[NumPoints] M Array // Measures
}

```

สำหรับ fields ของ PolyLineZ จะอธิบายรายละเอียดดังข้างล่างนี้

Box Bounding Box สำหรับ PolygonM จะเก็บค่าของ Xmin, Ymin, Xmax, Ymax

NumParts เป็นตัวเลขของเส้นใน PolyLineZ

Numpoints เป็นผลรวมของ point สำหรับทุกเส้น

Parts เป็น array ของความยาว NumParts จะเก็บสำหรับแต่ละเส้น โดย index ของจุดแรกใน array ส่วน array indexes ร่วมกับความสัมพันธ์เป็นศูนย์

Point เป็น array ของความยาว NumPoint จุดสำหรับแต่ละเส้นใน PolyLineM เป็นการเก็บแบบ end to end Points สำหรับ Part 2 ตามด้วย point สำหรับ Part 1 และที่เหลือ

Z Range ค่า Z ที่น้อยสุดและมากที่สุด สำหรับ PolyLineZ ซึ่งจะเก็บใน Zmin , Zmax

Z Array เป็น array ของความยาว NumPoints ปริมาณของแต่ละเส้นใน PolyLineM เป็นการเก็บแบบ end to end Parts ค่า Z สำหรับ Part 2 ตามด้วยค่า Z สำหรับ Part 1 และที่เหลือ

M Range ปริมาณที่น้อยสุดและมากที่สุด สำหรับ PolyLineZ ซึ่งจะเก็บใน Mmin , Mmax

M Array เป็น array ของความยาว NumPoints ปริมาณของแต่ละเส้นใน PolyLineZ เป็นการเก็บแบบ end to end ปริมาณสำหรับ Part 2 ตามด้วยปริมาณสำหรับ Part 1 และที่เหลือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	13	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little
Byte Y	Zmin	Zmin	Double	1	Little
Byte Y + 8	Zmax	Zmax	Double	1	Little
Byte Y + 16	Zarray	Zarray	Double	NumPoints	Little
Byte Z*	Mmin	Mmin	Double	1	Little
Byte Z+8*	Mmax	Mmax	Double	1	Little
Byte Z+16*	Marray	Marray	Double	NumPoints	Little

Note: $X = 44 + (4 * NumParts)$, $Y = X + (16 * NumPoints)$, $Z = Y + 16 + (8 * NumPoints)$
 * optional

ตารางที่ 2.15 PolyLineZ Record Contents

PolygonZ PolygonZ จะประกอบไปด้วยตัวเลขของวงแหวน เป็นรูปแบบวงแหวนปิด ซึ่ง PolygonZ จะประกอบไปด้วยวงแหวนภายนอกหลาย ๆ วง ซึ่งวงแหวนของ PolygonZ เป็นการอ้างอิงถึงเส้นของมันเอง

โครงสร้างของ PolygonZ เป็นเหมือนกันทุกประการกับโครงสร้าง PolyLineZ

PolygonZ

```
{
  Double[4] Box // Bounding Box
  Integer NumParts // Number of Parts
  Integer NumPoints // Total Number of Points
  Integer[NumParts] Parts // Index to First Point in Part
  Point[NumPoints] Points // Points for All Parts
  Double[2] Z Range // Bounding Z Range
  Double[NumPoints] Z Array // Z Values for All Points
  Double[2] M Range // Bounding Measure Range
  Double[NumPoints] M Array // Measures
}
```

เอกสารนี้เป็นเอกสารสำหรับ field ของ PolygonZ จะเป็นดังนี้ หากท่านนั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Box	Bounding Box สำหรับ PolygonM จะเก็บค่าของ Xmin, Ymin, Xmax, Ymax
NumParts	เป็นตัวเลขของวงแหวนใน PolygonZ
Numpoints	เป็นผลรวมของ point สำหรับทุกวงแหวน
Parts	เป็น array ของความยาว NumParts จะเก็บสำหรับแต่ละเส้น โดย index ของจุดแรกใน array ส่วน array indexes ร่วมกับความสัมพันธ์เป็นศูนย์
Point	เป็น array ของความยาว NumPoint จุดสำหรับแต่ละเส้นใน PolyLineM เป็นการเก็บแบบ end to end Points สำหรับ Ring 2 ตามด้วย point สำหรับ Ring 1 และที่เหลือ
Z Range	ค่า Z ที่น้อยสุดและมากที่สุด สำหรับเก็บใน Zmin , Zmax
Z Array	เป็น array ของความยาว NumPoints ค่าของ Z สำหรับแต่ละวงแหวนใน PolygonZ เป็นการเก็บแบบ end to end ค่า Z สำหรับ Ring 2 ตามด้วยค่า Z สำหรับ Ring 1 และที่เหลือ
M Range	ปริมาณที่น้อยสุดและมากที่สุด สำหรับ PolygonZ เก็บใน Mmin , Mmax
M Array	เป็น array ของความยาว NumPoints ปริมาณสำหรับแต่ละวงแหวนใน PolygonZ เป็นการเก็บแบบ end to end ปริมาณสำหรับ Ring 2 ตามด้วยค่า Z สำหรับ Ring 1 และที่เหลือ

ข้างล่างนี้เป็นส่วนสำคัญที่เกี่ยวกับรูปร่างของ PolygonZ

- วงแหวนปิด
- วงแหวนในจุด array ไม่เป็นที่น่าสนใจ

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	15	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little
Byte Y	Zmin	Zmin	Double	1	Little
Byte Y+8	Zmax	Zmax	Double	1	Little
Byte Y+16	Zarray	Zarray	Double	NumPoints	Little
Byte Z*	Mmin	Mmin	Double	1	Little
Byte Z+8*	Mmax	Mmax	Double	1	Little
Byte Z+16*	Marray	Marray	Double	NumPoints	Little

Note: $X = 44 + (4 * \text{NumParts})$, $Y = X + (16 * \text{NumPoints})$, $Z = Y + 16 + (8 * \text{NumPoints})$
 * optional

ตารางที่ 2.16 PolygonZ Record Contents

MultiPatch MultiPatch จะประกอบไปด้วยตัวเลขของพื้นผิว ซึ่งพื้นผิวของ MultiPatch เป็นการอ้างอิงถึงเส้นของมัน และชนิดของเส้นที่ควบคุมของเส้นตัดของ MultiPatch เส้นของ MultiPatch สามารถแบ่งเป็นชนิดได้ดังนี้

1. **Triangle Strip** เป็นการเชื่อมโยงของสี่เหลี่ยมมุมฉาก ที่ทุก ๆ จุดตัดของเส้นที่สมบูรณ์เป็นสี่เหลี่ยมมุมฉากใหม่ ซึ่งสี่เหลี่ยมมุมฉากใหม่เป็นรูปแบบของการติดต่อกันของจุดตัดใหม่กับทั้งสองที่กะทันหันแทนตำแหน่งเดิม

2. **Triangle Fan** เป็นการเชื่อมโยงของสี่เหลี่ยมรูปพัด ที่ทุก ๆ จุดตัดของเส้นที่สมบูรณ์เป็นสี่เหลี่ยมใหม่ ซึ่งสี่เหลี่ยมใหม่เป็นรูปแบบของการติดต่อกันของจุดตัดใหม่กับทั้งสองที่กะทันหันแทนตำแหน่งเดิม และจุดตัดแรกของเส้น

3. **Outer Ring** ด้านนอกวงแหวนของ Polygon

4. **Inner Ring** ภายในวงแหวนของ Polygon

5. **First Ring** วงแหวนวงแรกของ Polygon ของชนิดที่ไม่ได้ระบุไว้

6. **Ring** วงแหวนของ Polygon ของชนิดที่ไม่ได้ระบุไว้

สี่เหลี่ยมมุมฉากรูปเดียว หรือสี่เหลี่ยมรูปพัดจะถูกแทนด้วยพื้นผิวหนึ่งพื้นผิวดังรูปที่ 3 เป็นตัวอย่างของชนิดของเส้น

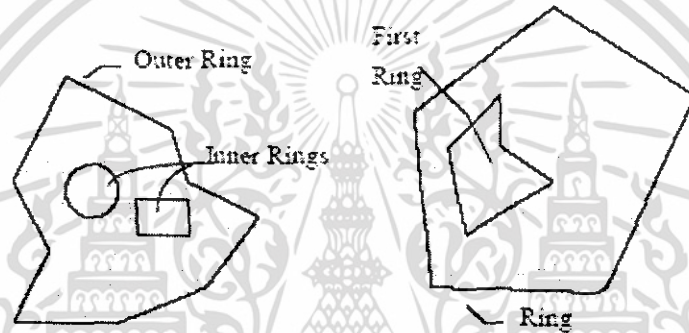
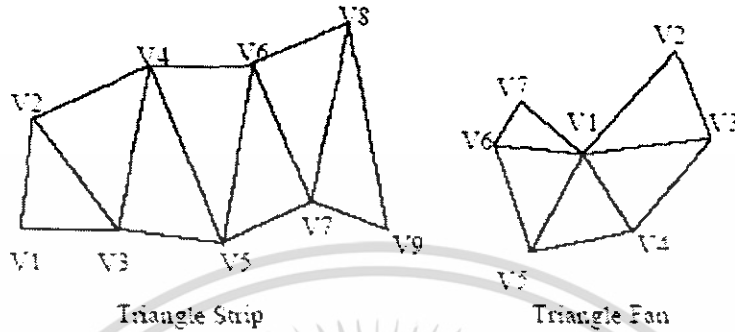
ลำดับของเส้นที่เป็นวงแหวนสามารถอธิบายเป็นพื้นผิวแบบ Polygon ได้โดยจะประกอบไปด้วยชนิดของ Outer Ring แทนด้วยเขตแดนด้านนอกของ patch ตามด้วย

ตัวเลขของ Inner Ring แทนด้วยช่องเพื่อให้เห็นความแตกต่างของชนิดวงแหวนเราจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แทนที่ด้วย พื้นที่หลายเหลี่ยม กับช่อง ลำดับแรกคือ First Ring ตามด้วยตัวเลขของ Ring ลำดับของ Ring จะไม่สำคัญกว่า First Ring จะรักษาลำดับของ Outer Ring ไว้ ด้านนอกช่อง



รูปที่ 2.11 แสดงตัวอย่างของชนิดของเส้น MultiPatch

ค่าที่ใช้สำหรับ encoding ชนิดของเส้นเป็นดังนี้

Value	Part Type
0	Triangle Strip
1	Triangle Fan
2	Outer Ring
3	Inner Ring
4	First Ring
5	Ring

MultiPatch

{

Double[4] Box // Bounding Box

Integer NumParts // Number of Parts

Integer NumPoints // Total Number of Points

Integer[NumParts] Parts // Index to First Point in Part

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Integer[NumParts] PartTypes // Part Type
Point[NumPoints] Points // Points for All Parts
Double[2] Z Range // Bounding Z Range
Double[NumPoints] Z Array // Z Values for All Points
Double[2] M Range // Bounding Measure Range
Double[NumPoints] M Array // Measures

```

}

สำหรับแต่ละ field ของ MultiPatch เป็น

Box	Bounding Box สำหรับ PolygonM จะเก็บค่าของ Xmin, Ymin, Xmax, Ymax
NumParts	เป็นตัวเลขของเส้นใน MultiPatch
Numpoints	เป็นผลรวมของ point สำหรับทุกเส้น
Parts	เป็น array ของความยาว NumParts จะเก็บสำหรับแต่ละเส้น โดย index ของจุดแรกใน array ส่วน array indexes ร่วมกับความสัมพันธ์เป็นศูนย์
Point	เป็น array ของความยาว NumPoint จุดสำหรับแต่ละเส้นใน PolyLineM เป็นการเก็บแบบ end to end Points สำหรับ Ring 2 ตามด้วย point สำหรับ Ring 1 และที่เหลือ
Z Range	ค่า Z ที่น้อยสุดและมากที่สุด สำหรับเก็บใน Zmin, Zmax
Z Array	เป็น array ของความยาว NumPoints ค่า Z ของแต่ละเส้นใน MultiPatch เป็นการเก็บแบบ end to end ค่า Z สำหรับ Part 2 ตามด้วยค่า Z สำหรับ Part 1 และที่เหลือ
M Range	ปริมาณที่น้อยสุดและมากที่สุด สำหรับ MultiPatch จะเก็บใน Mmin, Mmax
M Array	เป็น array ของความยาว NumPoints ปริมาณของแต่ละเส้นใน MultiPatch เป็นการเก็บแบบ end to end ปริมาณสำหรับ Part 2 ตามด้วยปริมาณสำหรับ Part 1 และที่เหลือ

ข้างล่างนี้เป็นส่วนสำคัญที่เก็บเกี่ยวกับรูปร่างของ MultiPatch

- ถ้าเส้นเป็นวงแหวนจะเป็นวงแหวนปิด
- ลำดับของเส้นที่เป็นวงแหวนใน point array มีความหมายเป็น Inner Ring ตามด้วยพวก Outer Ring ลำดับของ Ring แทนด้วยพื้นผิวหนึ่งส่วนร่วมกับ วงแหวนของชนิด First Ring
- เส้นต่าง ๆ สามารถเป็นเส้นแบ่งเขตร่วมได้ แต่เส้นเหล่านี้จะไม่ตัดขวาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายใน และไม่เผยแพร่สู่สาธารณะโดยไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

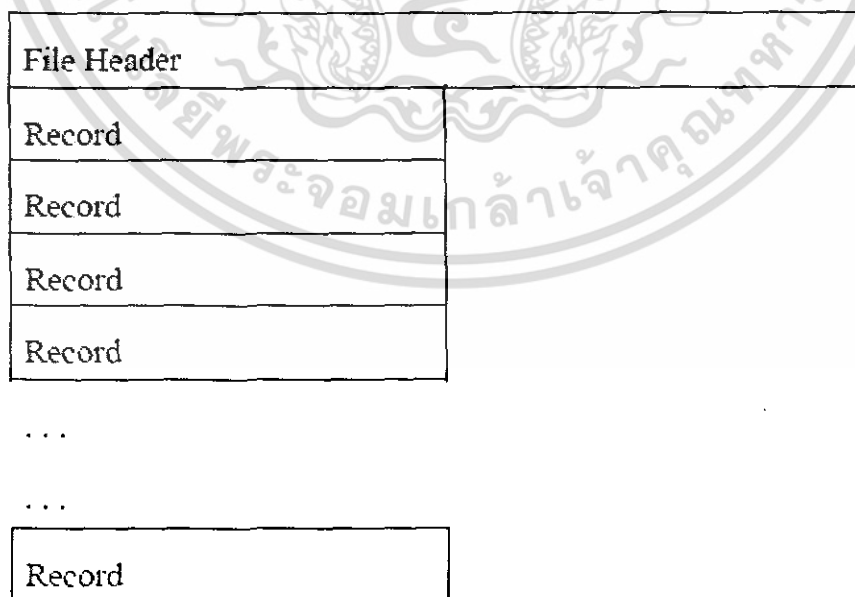
Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	31	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte W	PartTypes	PartTypes	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little
Byte Y	Zmin	Zmin	Double	1	Little
Byte Y+8	Zmax	Zmax	Double	1	Little
Byte Y+16	Zarray	Zarray	Double	NumPoints	Little
Byte Z*	Mmin	Mmin	Double	1	Little
Byte Z+8*	Mmax	Mmax	Double	1	Little
Byte Z+16*	Marray	Marray	Double	NumPoints	Little

Note: $W = 44 - (4 * \text{NumParts})$, $X = W - (4 * \text{NumParts})$, $Y = X + (16 * \text{NumPoints})$,
 $Z = Y - 16 + (8 * \text{NumPoints})$
 * optional

ตารางที่ 2.17 MultiPatch Record Contents

2.5.7 องค์ประกอบของ Index File

Index file (.shx) จะประกอบไปด้วยส่วนหัวที่มีขนาด 100byte ตามด้วย 8 byte จำกัดความยาวของ record ดังรูปที่ 4 จะแสดงถึงส่วนประกอบต่างๆ ของ index file



รูปที่ 2.12 ส่วนประกอบของ Index File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.7.1 The Index File Header

ส่วนหัวของ Index file มีส่วนประกอบเหมือนกับ header ของ Main file ยิ่งไปกว่านั้น ขนาดของการเก็บ file ใน Index file header มีผลรวมขนาดของ Index file เป็น 16-bit words

2.5.7.2 Index Records

ในหนึ่ง record ใน Index file จะเก็บ Offset และขนาดของ Content แต่สำหรับหนึ่ง record ของ main file ในตารางที่ 18 จะแสดงรายละเอียดของ field ใน file header

Position	Field	Value	Type	Byte Order
Byte 0	Offset	Offset	Integer	Big
Byte 4	Content Length	Content Length	Integer	Big

ตารางที่ 2.18 รายละเอียดของ Index Records

Offset ของ record ใน main file เป็นตัวเลขของ 16-bit words จาก byte เริ่มต้นของ main file ถึง byte แรกของ record header ดังนั้น offset สำหรับ record แรกใน main file เป็น 50 ให้ header มี 100 byte

ความยาว Content จะเก็บใน index record เป็นเหมือนกับค่าที่เก็บใน main file record header

2.5.8 องค์ประกอบของ dBASE File

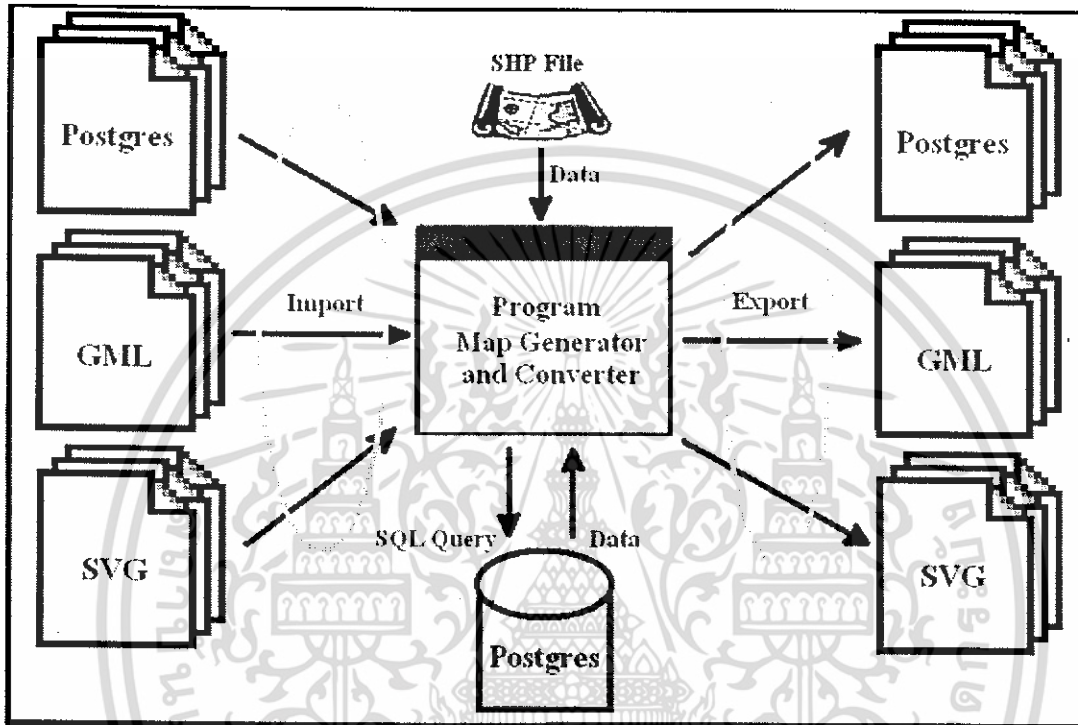
dBASE file (.dbf) จะประกอบไปด้วยจุดเด่นของ attribute หรือ key ของ attribute ซึ่งสามารถ joined กับตารางอื่นได้ มันเป็นรูปแบบมาตรฐาน DBF file ที่ใช้มากในโปรแกรมประยุกต์ของ Windows™ และ DOS. ชุดของ field สามารถแทนในตารางมีสามสิ่งที่จำเป็นดังนี้

1. ชื่อของ file เป็นเหมือนส่วนที่เติมข้างหน้า shape และ index file ส่วนที่ต่อท้ายจะเป็น .dbf (สามารถดูในตัวอย่างเรื่อง Naming Conventions)
2. ในตารางจะประกอบไปด้วยหนึ่ง record ต่อ ลักษณะของรูปร่าง
3. ใน record จะเหมือนกับลักษณะของรูปร่างใน main file (.shp)
4. ค่าของ year ใน dBASE header จะเริ่มตั้งแต่ 1900

บทที่ 3

หลักการทำงานของโปรแกรม

3.1 โครงสร้างและการออกแบบของโปรแกรม



รูปที่ 3.1 แสดง โครงสร้างและการออกแบบของ โปรแกรม

ตัวโปรแกรมสามารถที่จะนำข้อมูลแผนที่จาก SHP File, Postgres, GML และ SVG มาเก็บลงฐานข้อมูลของตัวเอง โดยจะทำการสร้าง Table ขึ้นมาใหม่ เพื่อใช้จัดการเกี่ยวกับการแสดงแผนที่หรือรายละเอียดต่างๆของไฟล์ที่นำเข้า จากนั้นก็จะนำข้อมูลจากฐานข้อมูลมาแปลงเป็น Postgres, GML และ SVG ตามความต้องการของผู้ใช้งาน

3.2 การนำข้อมูลจัดเก็บลงฐานข้อมูล

ฐานข้อมูลที่ใช้ในการเก็บรายละเอียดข้อมูลแผนที่ของโครงการนี้คือ PostgreSQL เป็นระบบจัดการฐานข้อมูล (Database Management System: DBMS) เพื่อจัดเก็บข้อมูล 2 ประเภท คือ Spatial Data และ Non-spatial Data นอกจากนี้ยังได้ติดตั้ง PostGIS เพื่อเป็นการเพิ่มประสิทธิภาพในการจัดการข้อมูลเชิงพื้นที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การนำข้อมูลจาก shapeFile เก็บลงฐานข้อมูล

ใน shapeFile นั้นจะประกอบไปด้วยไฟล์ 3 ไฟล์ ดังนี้

1. Mainfile: มีนามสกุล (.shp) Mainfile เป็นไฟล์ที่เก็บข้อมูลรายละเอียดทั้งหมดของแผนที่ไว้ เช่น field, detail, The_Geom เป็นต้น
2. Indexfile: มีนามสกุล (.shx) เป็นไฟล์ที่เก็บขนาดของแต่ละ field ใน Mainfile
3. dBASE: มีนามสกุล (.dbf) เป็นไฟล์ที่เก็บลักษณะของ attribute หรือ keys ที่ใช้ join กับตารางอื่น

ข้อมูลเชิงภูมิศาสตร์ที่ใช้ทำการทดลองคือ ข้อมูลแผนที่จังหวัดระยองและ ข้อมูลแผนที่บริเวณรอบๆสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เป็นตัวอย่างในการทดลอง ในฐานข้อมูลจะมี Field ที่เหมือนกันของทั้ง Postgres, GML และ SVG แต่ในขั้นตอนต่อไปจะเป็นการนำข้อมูลที่ได้จาก SHP File มาเก็บลงฐานข้อมูล ซึ่งใน SHP File ประกอบด้วย field ที่สำคัญๆดังนี้

ขอบเขตของข้อมูล (Field)	ความหมาย
GID	เป็นค่า ID ของ object ข้อมูลแผนที่ (เป็น primary key)
"name"	เป็นชื่อและรายละเอียดต่างๆของ object ข้อมูลแผนที่
The_geom	ข้อมูลทางด้านภูมิศาสตร์

ตารางที่ 3.1 แสดงขอบเขตของข้อมูลและคำอธิบาย

โดยข้อมูลที่นำมาใช้สามารถแยกประเภทชนิดของข้อมูลได้ 3 ประเภท ได้แก่

1. ข้อมูลแบบจุด ได้แก่ point
2. ข้อมูลแบบเส้น ได้แก่ linestring, multilinestring
3. ข้อมูลแบบโพลีกอน ได้แก่ polygon, multipolygon

ซึ่งจะเห็นได้ตามตัวอย่างในฐานข้อมูลดังต่อไปนี้

1. ตาราง T_point เป็นข้อมูลประเภท point

```
pgAdmin III Query - postgres on localhost:5432 - [C:\point.sql]
File Edit Query Help
postgres on localhost:5432
CREATE TABLE "t_point" (gid serial, "point" int4, "name" varchar, "x" int4, "y" int4, "จำนวนครั้ง" int8, "จำนวนเงิน" int8, "จำนวนสาย" int8, "อุบัติเหตุ" int8, "จำนวนรถ" int8, "the_geom" geometry(Point, 2));
SELECT AddGeometryColumn('','t_point','the_geom','-1','POINT',2);
COPY "t_point" (gid,"point","name","x","y","จำนวนครั้ง","จำนวนเงิน","จำนวนสาย","อุบัติเหตุ","จำนวนรถ","the_geom") FROM stdin;
0 1 ซิงขากของทาง 743415 1402855 0 0 1 0.2996 10 SRID=-1;POINT(743415 1402855)
1 2 ขนขลุ่ยข้างลำพง4 747017 1402761 9 5 0 0.2451 17 SRID=-1;POINT(747017 1402761)
2 3 ซิงขากของทาง 739207 1435348 5 6 0 1.3483 14 SRID=-1;POINT(739884.325702726 14353
3 4 ซิงขากของทาง 773985 1431182 5 5 1 0.2491 8 SRID=-1;POINT(773985 1431182)
4 5 ซิงขากของทาง 739152 1404136 7 4 0 0.1586 11 SRID=-1;POINT(739152 1404136)
5 6 ซิงขากของทาง 743755 1405261 11 10 4 0.4180 21 SRID=-1;POINT(743755 1405261)
6 8 ซิงขากของทาง36 748884 1404107 6 7 0 0.1792 9 SRID=-1;POINT(748884 1404107)
7 10 ซิงขากของทาง 1402782 14 12 0 0.2730 10 SRID=-1;POINT(1402782 14)
```

รูปที่ 3.2 แสดงขอบเขตและชนิดของข้อมูลแบบ point

2. ตาราง T_road เป็นข้อมูลประเภท multilinestring

```
pgAdmin III Query - postgres on localhost:5432 - [C:\road.sql]
File Edit Query Help
postgres on localhost:5432
CREATE TABLE "t_road" (gid serial, "fnode" int8, "tnode" int8, "lpoly" int8, "rpoly" int8, "length" float8, "trans" int8, "trans_id" int8, "tr_type" int8, "tr_name" varchar, "the_geom" geometry(MultiLineString, 2));
SELECT AddGeometryColumn('','t_road','the_geom','-1','MULTILINESTRING',2);
COPY "t_road" (gid,"fnode","tnode","lpoly","rpoly","length","trans","trans_id","tr_type","tr_name","the_geom") FROM stdin;
0 128229 128041 0 0 10157.970000 1 23 23 1 SRID=-1;MULTILINESTRING((780402.1675 1452584.6
1 128365 128381 0 0 84.515720 2 23 23 1 SRID=-1;MULTILINESTRING((780546.875 1450832.37
2 128250 128406 0 0 3471.551000 3 23 23 1 SRID=-1;MULTILINESTRING((780727.6875 1452137.2
3 128381 128494 0 0 1024.654000 4 23 23 1 SRID=-1;MULTILINESTRING((780804.5 1450759.25
4 128895 128899 0 0 635.175200 5 43 43 1 SRID=-1;MULTILINESTRING((740001.625 1446547.12
5 128937 128899 0 0 3501.432000 6 43 43 1 SRID=-1;MULTILINESTRING((736615.375 1446057.75
6 128937 128911 0 0 506.807000 7 12 12 1 SRID=-1;MULTILINESTRING((736615.375 1446087.75
7 128938 128934 0 0 157.676900 8 22 22 1 SRID=-1;MULTILINESTRING((735535.175 1446081.25
```

รูปที่ 3.3 แสดงขอบเขตและชนิดของข้อมูลแบบ Line

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ตาราง T_amphoe เป็นข้อมูลประเภท multipolygon

```

pgAdmin III Query - postgres on localhost:5432 - [C:\amphoe.sql]
File Edit Query Help
postgres on localhost:5432
CREATE TABLE "t_amphoe" (gid serial, "area" float8, "perimeter" float8, "amphoe_" int8, "amphoe_id" int8, "amp_code" varchar, "amp_name"
SELECT AddGeometryColumn('t_amphoe', 'the_geom', 4, 'MULTIPOLYGON', 2);
COPY "t_amphoe" (gid,"area", "perimeter", "amphoe_", "amphoe_id", "amp_code", "amp_name", "sp_nrd2c", "prov_code", "prov_name", the_geom) FROM std:
0 507122200 136090.400000 2 4707 4707 อ.เมืองระยอง 14707 47 ระยอง SRID=-1 ;MULTIPOLYGON(((777603.121
1 542347100 131493.500000 3 4704 4704 อ.เขาชะเมา 14704 47 ระยอง SRID=-1 ;MULTIPOLYGON(((740743.056
2 394688500 104000.600000 4 4706 4706 อ.วังน้ำเย็น 14706 47 ระยอง SRID=-1 ;MULTIPOLYGON(((774144.5625 14464
3 473513800 116416.100000 5 4703 4703 อ.บ้านค่าย 14703 47 ระยอง SRID=-1 ;MULTIPOLYGON(((762310.1675 14349
4 703251900 172967.200000 6 4702 4702 อ.ฉะเชิงเทรา 14702 47 ระยอง SRID=-1 ;MULTIPOLYGON(((905031.9375 14263
5 243797900 83763.600000 7 4708 4708 อ.บ้านฉาง 14708 47 ระยอง SRID=-1 ;MULTIPOLYGON(((726788.37
6 663449800 174208.000000 8 4701 4701 อ.เมืองระยอง 14701 47 ระยอง SRID=-1 ;MULTIPOLYGON(((769831.93
7 163444100 87671.200000 5 4705 4705 อ.บ้านค่าย 14705 47 ระยอง SRID=-1 ;MULTIPOLYGON(((773056.25 141699

```

รูปที่ 3.4 แสดงขอบเขตและชนิดของข้อมูลแบบ Polygon

3.4 การนำข้อมูลจาก GML เก็บลงฐานข้อมูล

การนำข้อมูลจาก GML นั้น ในโปรแกรม C#.NET จะมี XML serializer เป็นตัวอ่านข้อมูลไฟล์ GML โดยจะทำการอ่าน tag เฉพาะ Element ที่เกี่ยวข้องกับ field มาสร้างเป็นแอตทริบิวต์ในฐานข้อมูล และค่าที่อยู่ใน tag ก็จะเก็บลงตาม column name แต่จะมี Tag ที่ชื่อว่า <TheGeometry> ที่จะต้องมาดูชนิดของ object ว่าเป็นชนิดไหน เช่น point, line, polygon, polyline เป็นต้น โดย Element ที่จำเป็นใน GML มีดังนี้

GML	ACTION
<gml:featureCollection>	เป็น Tag ที่ใช้ในการกำหนดพื้นที่ทั้งหมดของภาพและเป็น Tag ที่เก็บ Tag ต่างๆของ GML
<gml:boundedBy>	ใช้ในการกำหนดพื้นที่ของภาพ
<gml:coordinates>	ใช้ในการแสดงข้อมูลพิกัด
<gml:featureMember>	ใช้ในการรวม Tag ต่างๆให้อยู่ในกลุ่มเดียวกัน

ตารางที่ 3.2 แสดง Element ของ GML

เอกสาร amphoe.gml ซึ่งเป็นข้อมูลชนิด polygon เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<?xml version="1.0"?>
<wfs:FeatureCollection xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns
<gml:description>description for FeatureCollection.</gml:description>
<gml:boundedBy>
  <gml:Box>
    <gml:coordinates>715838,1456514.75 807625.1875,1384225.625 </gml:
  </gml:Box>
</gml:boundedBy>
<gml:featureMember>
  <amphoe fid="amphoe_0">
    <AREA>507122200</AREA>
    <PERIMETER>138090.4</PERIMETER>
    <AMPHOE_>2</AMPHOE_>
    <AMPHOE_ID>4707</AMPHOE_ID>
    <AMP_CODE>4707</AMP_CODE>
    <AMP_NAME>!Th'M.'`R`P`AR</AMP_NAME>
    <AP_NRD2C>14707</AP_NRD2C>
    <PROV_CODE>47</PROV_CODE>
    <PROV_NAME>(.CPBM'</PROV_NAME>
    <TheGeometry>
      <gml:Polygon>
        <gml:outerBoundaryIs>
          <gml:LinearRing>
            <gml:coordinates>777603.125,1446301.75 777623.125,1446316
          </gml:LinearRing>
        </gml:outerBoundaryIs>
      </gml:Polygon>
    </TheGeometry>
  </amphoe>
</gml:featureMember>
</wfs:FeatureCollection>

```

รูปที่ 3.5 แสดงไฟล์เอกสาร amphoe.gml

AREA	PERIMETER	AMPHOE_	AMPHOE_I	AMP_CODE	AMP_NAME	AP_NRD2C	PROV_CODE	PROV_NAME	TheGeometry
507122200	138090.4	2	4707	4707	!Th'M.'`R`P`AR	14707	47	(.CPBM'	PolygonProp
542347100	131493.5	3	4704	4704	M.EGla4'	14704	47	(.CPBM'	PolygonProp
394688600	104000.8	4	4706	4706	M.GQ1Q97	14706	47	(.CPBM'	PolygonProp
478613800	116416.1	5	4703	4703	M.RS\$hrB	14703	47	(.CPBM'	PolygonProp
705251300	172967.2	6	4702	4702	M.aE'	14702	47	(.CPBM'	PolygonProp
249797900	83762.63	7	4708	4708	!Th'M.9T\$A	14708	47	(.CPBM'	PolygonProp
662449800	174209	8	4701	4701	M.'AWM'CP	14701	47	(.CPBM'	PolygonProp
163944100	57671.28	9	4705	4705	M.'RSJR'	14705	47	(.CPBM'	PolygonProp
233267.7	2985.855	10	4702	4702	M.aE'	14702	47	(.CPBM'	PolygonProp
35487.47	1082.194	11	4701	4701	M.'AWM'CP	14701	47	(.CPBM'	PolygonProp
152410.4	1960.389	12	4702	4702	M.aE'	14702	47	(.CPBM'	PolygonProp
17341.91	575.6382	13	4701	4701	M.'AWM'CP	14701	47	(.CPBM'	PolygonProp
12196.56	493.3925	14	4701	4701	M.'AWM'CP	14701	47	(.CPBM'	PolygonProp
95271.34	1442.262	15	4701	4701	M.'AWM'CP	14701	47	(.CPBM'	PolygonProp
5170695	19155.38	16	4701	4701	M.'AWM'CP	14701	47	(.CPBM'	PolygonProp
9316.129	370.517	17	4701	4701	M.'AWM'CP	14701	47	(.CPBM'	PolygonProp
156605.3	1717.881	18	4702	4702	M.aE'	14702	47	(.CPBM'	PolygonProp
64681.92	1060.46	19	4702	4702	M.aE'	14702	47	(.CPBM'	PolygonProp
50347.15	1097.131	20	4701	4701	M.'AWM'CP	14701	47	(.CPBM'	PolygonProp

รูปที่ 3.6 แสดงฐานข้อมูลที่ได้จากไฟล์ amphoe.gml

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การสร้างแผนที่ในรูปแบบเอกสาร GML

การสร้างเอกสาร GML นั้นก็จะทำงานคล้ายๆกับการนำข้อมูลลงฐานข้อมูล แต่การสร้างนั้นจะต้องเขียนให้ครบ Element ของเอกสาร GML และต้องตรงกับ Schema ไวยากรณ์และโครงสร้างของเอกสาร GML โดยจะทำการอ่านข้อมูลจากฐานข้อมูลซึ่งเป็นข้อมูลที่อยู่ในรูปของ Spatial Data และ Non-spatial Data มาทีละตาราง และนำข้อมูลที่ได้จากฐานข้อมูลเหล่านั้นมาสร้างเป็นเอกสาร GML ตามโครงสร้างที่ถูกต้อง ซึ่งข้อมูลที่ได้ผลลัพธ์จะถูกส่งออกไปให้แก่ผู้ใช้นั้นเป็นเอกสารซึ่งมีนามสกุล .gml

ขั้นตอนในการสร้างเอกสาร GML มีดังนี้

1. เป็นการประกาศนามสเปสต่างๆ ที่เกี่ยวข้องในการสร้างเอกสาร GML ซึ่งในโครงงานนี้เราจะสร้างเอกสาร XML เวอร์ชัน 1.0

```
<?xml version="1.0" encoding="utf-8"?>
<gml:featureCollection xmlns:gml=http://www.opengis.net/gml
xmlns:xlink="http://www.w3.org/1999/xlink">
```

2. ในส่วนถัดมาเป็นส่วนที่กำหนดขอบเขตของข้อมูลแผนที่ ซึ่งจะถูกสร้างอยู่ใน Tag ที่ชื่อว่า <gml:boundedBy> ซึ่งสามารถคำนวณได้จากข้อมูลที่ได้รับมาจากฐานข้อมูล โดยตัวโปรแกรมจะนำเอาพิกัดที่มีค่าพิกัดในแกน X,Y น้อยที่สุดมาใส่ไว้ใน Tag <gml:coordinates> โดยเรียงลำดับดังนี้ (Xmin,Ymin,Xmax,Ymax) มาจากข้อมูลทั้งหมดที่ได้รับมาจากฐานข้อมูล

```
<gml:boundedBy>
  <gml:Box>
    <gml:coordinates>715838,1456514.75 807625.1875,1384225.625 </gml:coordinates>
  <gml:Box>
</gml:boundedBy>
```

3. จากโปรแกรมจะอ่านข้อมูลทั้งหมดทั้ง Spatial Data และ Non-spatial Data และนำข้อมูลเหล่านั้นมาใส่ใน Tag ที่สร้างไว้ โดยเรียงลำดับจากคอลัมน์ในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<gml:featureMember>
  <amphoe fid="amphoe_0">
    <AREA>507122200</AREA>
    <PERIMETER>138090.4</PERIMETER>
    <AMPHOE_>2</AMPHOE_>
    <AMPHOE_ID>4707</AMPHOE_ID>
    <AMP_CODE>4707</AMP_CODE>
    <AMP_NAME>!Th'M.'R*P'AR</AMP_NAME>
    <AP_NRD2C>14707</AP_NRD2C>
    <PROV_CODE>47</PROV_CODE>
    <PROV_NAME>(CPBM'</PROV_NAME>
    <TheGeometry>
    <gml:Polygon>
      <gml:outerBoundaryIs>
        <gml:LinearRing>
          <gml:coordinates>777603.125,1446301.75 777623.125,1446316.125
777686.0625,1446350.5 777789.0625,1446412.75 777882.25,1446469.125
777999.6875,1446646.25 778156.125,1446913.25 778306.6875,1447153.375
          </gml:coordinates>
        </gml:LinearRing>
      </gml:outerBoundaryIs>

```

3.6 การนำข้อมูลจาก SVG ลงฐานข้อมูล

การนำข้อมูล SVG ลงฐานข้อมูลนั้นจะมีวิธีการที่คล้ายกับของ GML แต่จะต่างกันตรงที่ Element ของ SVG จะบอกรายละเอียดเฉพาะค่าที่ใช้ในการแสดงผล จะมีรายละเอียดได้ไม่เยอะ เหมือนกับ GML และ Attribute ในฐานข้อมูลก็จะน้อยกว่าฐานข้อมูลของ Shapefile และ GML เพราะ SVG เน้นที่การแสดงผลภาพแผนที่ รายละเอียดอื่นๆจึงถูกตัดทิ้งไป ซึ่ง Attribute หลักๆ ของ SVG ที่เก็บลงฐานข้อมูลคือ

- ID คือ ค่า ID ของ object แต่ละตัว
- Type คือ ค่าของชนิด object ว่าเป็นแบบไหน เช่น point, polyline, polygon
- The_geom คือค่าที่เก็บ coordinate ของ object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Detail คือจะเก็บค่ารายละเอียดอื่นๆของ object
- Class คือค่าของ style ของแต่ละ object

เอกสาร amphoe.svg ซึ่งเป็นข้อมูลชนิด polygon

```
<?xml version="1.0"?>
<svg xmlns:xlink="http://www.w3.org/1999/xlink" x="0" y="0" width="100%" h
<style type="text/css">.style-amphoe-0 {fill:RGB(76,58,102);fill-opacity
<g transform="matrix(1 0 0 -1 0 2840740.375 )" >
  <g id="amphoe">
    <path id="amphoe_f0" class="style-amphoe-0" d="M 777603.125,1446301.
    <path id="amphoe_f1" class="style-amphoe-0" d="M 740743.0625,1446401
    <path id="amphoe_f2" class="style-amphoe-0" d="M 774144.5625,1446419
    <path id="amphoe_f3" class="style-amphoe-0" d="M 762510.1875,1434956
    <path id="amphoe_f4" class="style-amphoe-0" d="M 805031.9375,1426375
    <path id="amphoe_f5" class="style-amphoe-0" d="M 726788.375,1429876.
    <path id="amphoe_f6" class="style-amphoe-0" d="M 769801.9375,1423409
    <path id="amphoe_f7" class="style-amphoe-0" d="M 722456.25,1416992.7
    <path id="amphoe_f8" class="style-amphoe-0" d="M 792429.6875,1396010
    <path id="amphoe_f9" class="style-amphoe-0" d="M 773677.125,1394479.
    <path id="amphoe_f10" class="style-amphoe-0" d="M 792974.4375,139435
    <path id="amphoe_f11" class="style-amphoe-0" d="M 772955.1875,139324
    <path id="amphoe_f12" class="style-amphoe-0" d="M 773132.125,1393326
    <path id="amphoe_f13" class="style-amphoe-0" d="M 772951.5,1392451.6
    <path id="amphoe_f14" class="style-amphoe-0" d="M 766649.75,1391934
    <path id="amphoe_f15" class="style-amphoe-0" d="M 773150.375,1391825
    <path id="amphoe_f16" class="style-amphoe-0" d="M 793703.125,1390830
    <path id="amphoe_f17" class="style-amphoe-0" d="M 779655,1389176 L 7
    <path id="amphoe_f18" class="style-amphoe-0" d="M 765698.1875,138422
  </g>
</g>
</svg>
```

รูปที่ 3.7 แสดงไฟล์เอกสาร amphoe.svg

SVG	Action
<svg>	เป็น Tag ที่ใช้ในการกำหนดพื้นที่ทั้งหมดของภาพและเป็น Tag ที่เก็บ Tag ต่างๆของ SVG
<path>	ใช้ในการวาดเส้น
<ellips>	ใช้ในการวาดวงรี
<g>	ใช้ในการรวม Tag ต่างๆให้อยู่กลุ่มเดียวกัน
<defs>	ใช้ในการกำหนดค่าต่างๆ
<script>	ในการเขียน script จะเขียนใน Tag

ตารางที่ 3.3 แสดงรายละเอียดของ Element ที่สำคัญในเอกสาร SVG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Attribute	Action
fill	จัดการสีของ object นั้นๆ
id	ใช้ในการแยกแยะ object ต่างๆ
Xmlns	ใช้ในการอ้างอิงถึง namespace
Visibility	กำหนดการปรากฏอยู่ของ object
viewBox	กำหนดขนาดในการแสดงผลของภาพ
Stroke_width	กำหนดขนาดของเส้น
Stroke	กำหนดสีของเส้น
Transform	เคลื่อนย้ายหรือหมุนหรืออัตราส่วนของภาพ
Xlink:href	เป็นการอ้างอิงถึง URL

ตารางที่ 3.4 แสดงรายละเอียดของ Attribute ที่สำคัญในเอกสาร SVG

3.7 การสร้างแผนที่ในรูปแบบเอกสาร SVG

การสร้างเอกสาร SVG นั้นก็จะต้องตรงกับ Schema ไวยากรณ์และ โครงสร้างของเอกสาร SVG โดยจะทำการอ่านข้อมูลจากฐานข้อมูลซึ่งเป็นข้อมูลที่อยู่ในรูปของ Spatial Data และ Non-spatial Data มาทีละตาราง และนำข้อมูลที่ได้จากฐานข้อมูลเหล่านั้นมาสร้างเป็นเอกสาร SVG ตามโครงสร้างที่ถูกต้อง ซึ่งข้อมูลที่ได้ผลลัพธ์จะถูกส่งออกไปให้แก่ผู้ใช้งานนั้น เป็นเอกสารซึ่งมีนามสกุล .svg

ขั้นตอนในการสร้างเอกสาร SVG มีดังนี้

4. เป็นการประกาศ เนมสเปสต่าง ๆ ที่เกี่ยวข้องในการสร้างเอกสาร SVG โดยในส่วนนี้จะรวมไปถึงค่าขอบเขตของ SVG viewBox ที่จะใช้แสดงรูปแผนที่ (ในแท็ก svg) โดยมีวิธีการดังนี้
 - หาค่า Xmin จากข้อมูลทั้งหมดที่ได้รับจากฐานข้อมูล
 - หาค่า Ymin จากข้อมูลทั้งหมดที่ได้รับจากฐานข้อมูล
 - คำนวณค่าความกว้างของขอบเขตโดยนำค่า $X_{max} - X_{min}$: width
 - คำนวณค่าความยาวของขอบเขตโดยนำค่า $Y_{max} - Y_{min}$: height

```
<?xml version="1.0"?>
<svg xmlns:xlink="http://www.w3.org/1999/xlink" x="0" y="0" width="100%" height="100%"
viewBox="715838 1384225.625 91787.1875 72289.125" xmlns="http://www.w3.org/2000/svg">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. จากนั้นโปรแกรมจะอ่านข้อมูลทั้งหมดทั้ง Spatial Data และ Non-spatial Data และนำข้อมูลเหล่านั้นมาใส่ในแท็ก<g> โดยมีแอตทริบิวต์ต่างๆดังนี้

```
<g id="amphoe" visibility="visible">
  <path id="amphoe_f0" class="style-amphoe-0" d="M 777603.125,1446301.75
  <path id="amphoe_f1" class="style-amphoe-0" d="M 740743.0625,1446401.75
  <path id="amphoe_f2" class="style-amphoe-0" d="M 774144.5625,1446419.875
  <path id="amphoe_f3" class="style-amphoe-0" d="M 762510.1875,1434956.375
  ...
  <path id="amphoe_f18" class="style-amphoe-0" d="M 765698.1875,1384225.625
</g>
```

- ชื่อของตารางของข้อมูลโพลีกอนมาใส่ไว้ในแอตทริบิวต์ id
- ถ้าต้องการให้ข้อมูลนี้ถูกแสดงอยู่ในแผนที่ให้กำหนดค่าในแอตทริบิวต์ visibility เป็น visible แต่ถ้าไม่ต้องการให้กำหนดเป็น hidden
- ในแท็ก <path> นำชื่อของสถานที่นั้นมาใส่ไว้ในแอตทริบิวต์ id
- ค่าพิกัดมาใส่ในแอตทริบิวต์ d ซึ่งในตัวอย่างนี้เป็นตัวอย่างข้อมูลแบบโพลีกอนจะทำการเก็บข้อมูลพิกัดในรูปของ d="M (แล้วตามด้วยคู่ลำดับพิกัดต่างๆ)"

นอกจากทำการนำข้อมูลต่างๆที่ได้รับมาจากฐานข้อมูลมาใส่ในแท็กต่างๆแล้ว ยังมี ส่วนของการคำนวณคุณสมบัติต่างๆ เพื่อใช้ในการแสดงผลซึ่งคำนวณได้จากค่าต่างๆในแท็ก <svg> หรือค่า svg ViewBox นั้นเอง ยกตัวอย่าง เช่น

- stroke-width คือ ความหนาของเส้นขอบของโพลีกอน ได้มาจากการเปรียบเทียบค่าระหว่างค่าความกว้าง (width) และ ค่าความยาว (height) ใน svg ViewBox นำค่าที่มากกว่ามาคำนวณดังนี้

stroke-width = 0.5 div MAX ; เมื่อค่า MAX เป็นค่าที่มากกว่าค่า

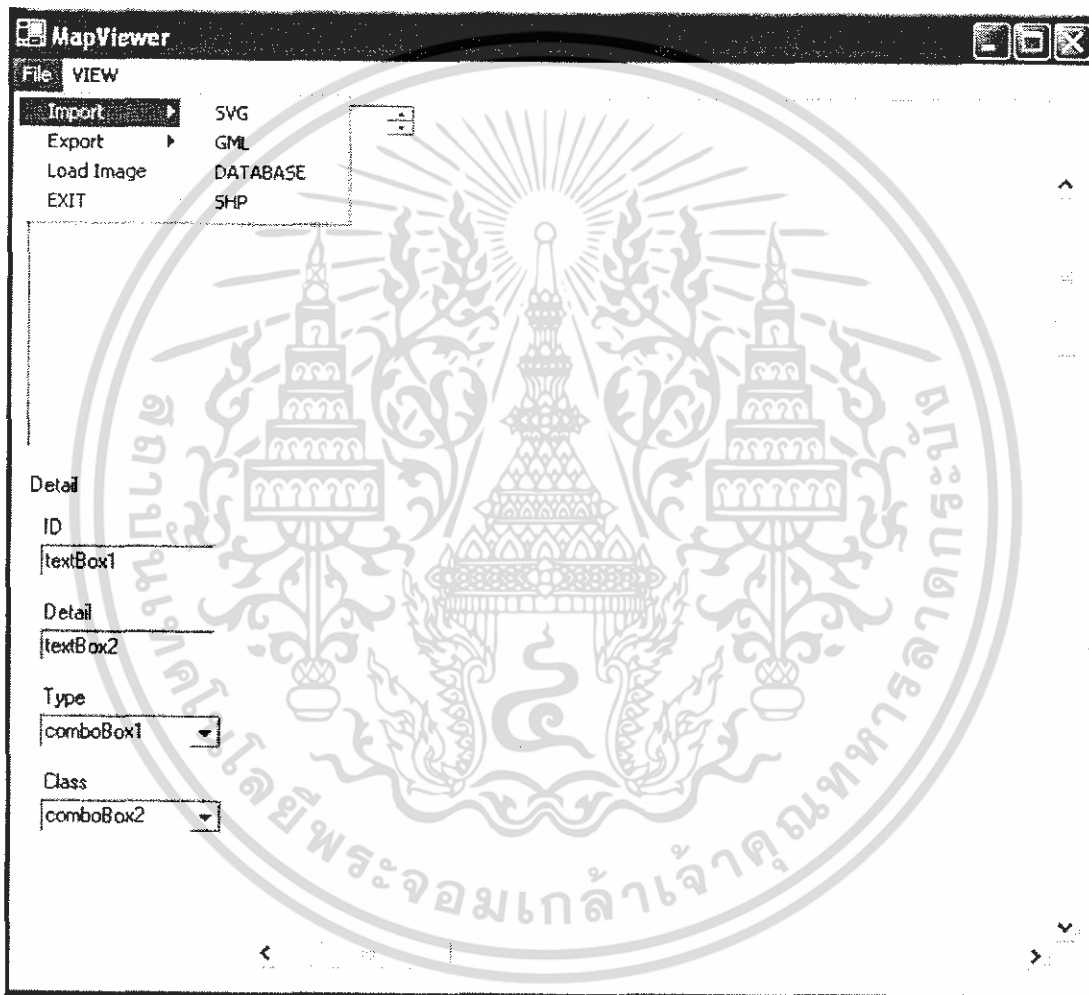
ความกว้างและความสูง

บทที่ 4

การทดลองและผลการทดลอง

4.1 Input

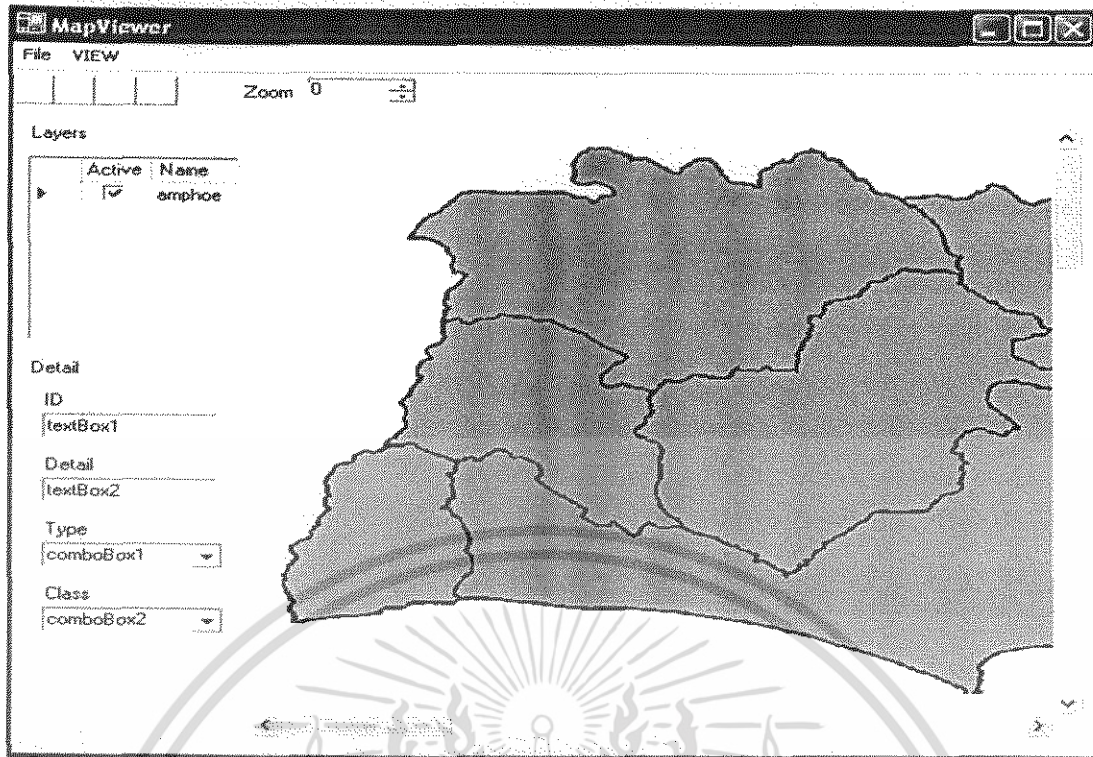
ในส่วนของ input นั้น ตัวโปรแกรม Map Generator and Converter นั้นจะสามารถรับ input ได้ 4 รูปแบบ คือ SVG, GML, Database, ShapeFile มาเก็บลงฐานข้อมูล



รูปที่ 4.1 แสดงการ Import ไฟล์เข้ามาโปรแกรม

จากนั้นตัวโปรแกรมจะไปทำการ query ในฐานข้อมูลแล้วขึ้นมาแสดงบน view ในที่นี้ทำได้ทำการทดลองจากไฟล์ amphoe.shp และในส่วนของการ view ก็จะสามารถที่จะ zoom ขนาดภาพของแผนที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



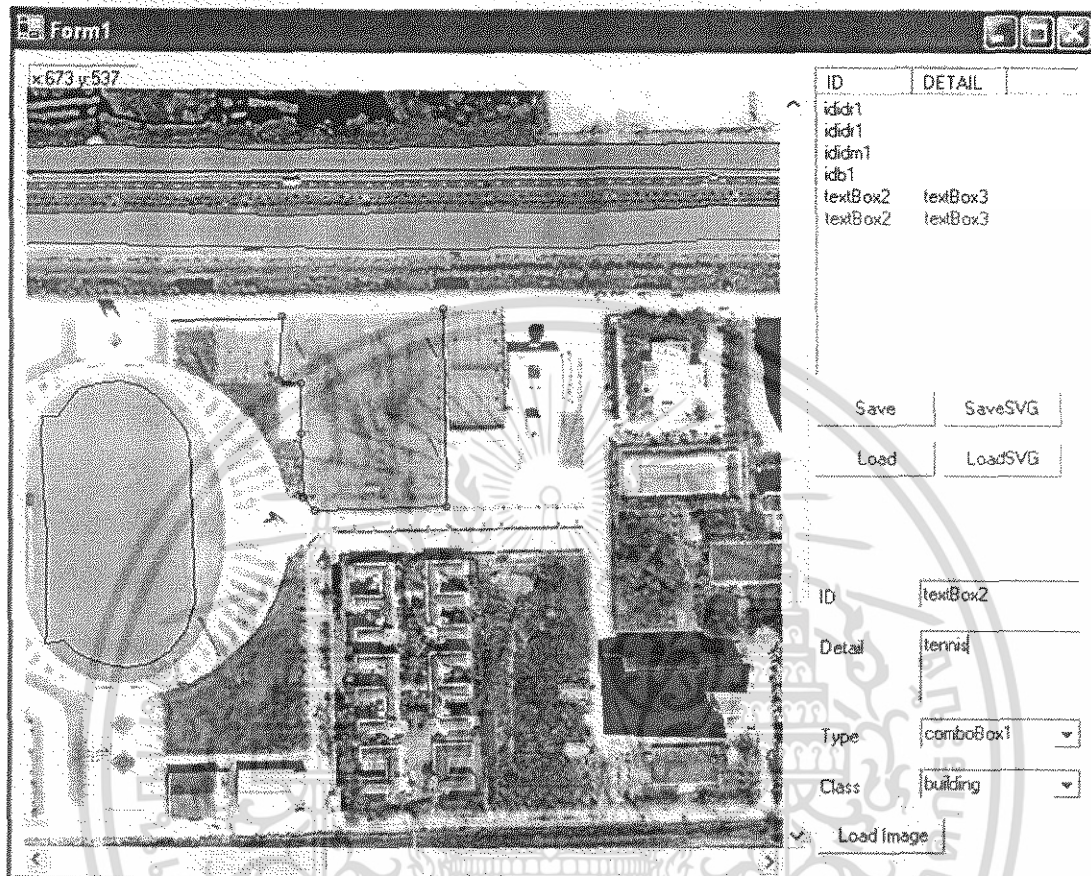
รูปที่ 4.2 แสดงภาพแผนที่ที่ถูก Import มาจาก shape file

เมื่อต้องการที่จะดู table ของ attribute ต่างๆของไฟล์ amphoe.shp

Active	Name	AREA	PERIMETER	AMPHOE_ID	AMPHOE_NAME	AMP_CODE	AA
<input checked="" type="checkbox"/>	amphoe	507122200	138090.4	2	4707	4707	ITP
		542347100	131493.5	3	4704	4704	M..
		394688600	104000.8	4	4706	4706	M.I
		478613800	116416.1	5	4703	4703	M..
		705251300	172967.2	6	4702	4702	M..
		249797900	83762.63	7	4708	4708	ITP
		662449800	174209	8	4701	4701	M..
		163944100	57671.28	9	4705	4705	M..
		233267.7	2985.855	10	4702	4702	M..
		35487.47	1082.194	11	4701	4701	M..
		152410.4	1960.389	12	4702	4702	M..
		17341.91	575.6382	13	4701	4701	M..
		12196.56	493.3925	14	4701	4701	M..
		95271.34	1442.262	15	4701	4701	M..
		5170695	19155.38	16	4701	4701	M..
		9316.129	370.517	17	4701	4701	M..
		156605.3	1717.881	18	4702	4702	M..
		64681.92	1060.46	19	4702	4702	M..
		50347.15	1097.131	20	4701	4701	M..

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 4.3 แสดงฐานข้อมูลของ Shapefile ชาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของการ edit ถ้าต้องการสร้างแผนที่ขึ้นมาเองให้ใกล้เคียงกับความเป็นจริง ถ้าผู้ใช้งานมีรูปภาพแผนที่ที่สามารถขึ้นมา เพื่อที่จะสร้างแผนที่ให้ใกล้เคียงกับความเป็นจริง และข้อมูลที่ได้ก็จะเก็บลงฐานข้อมูลเพื่อที่จะมาทำการแปลงไฟล์ตามความต้องการของผู้ใช้งาน

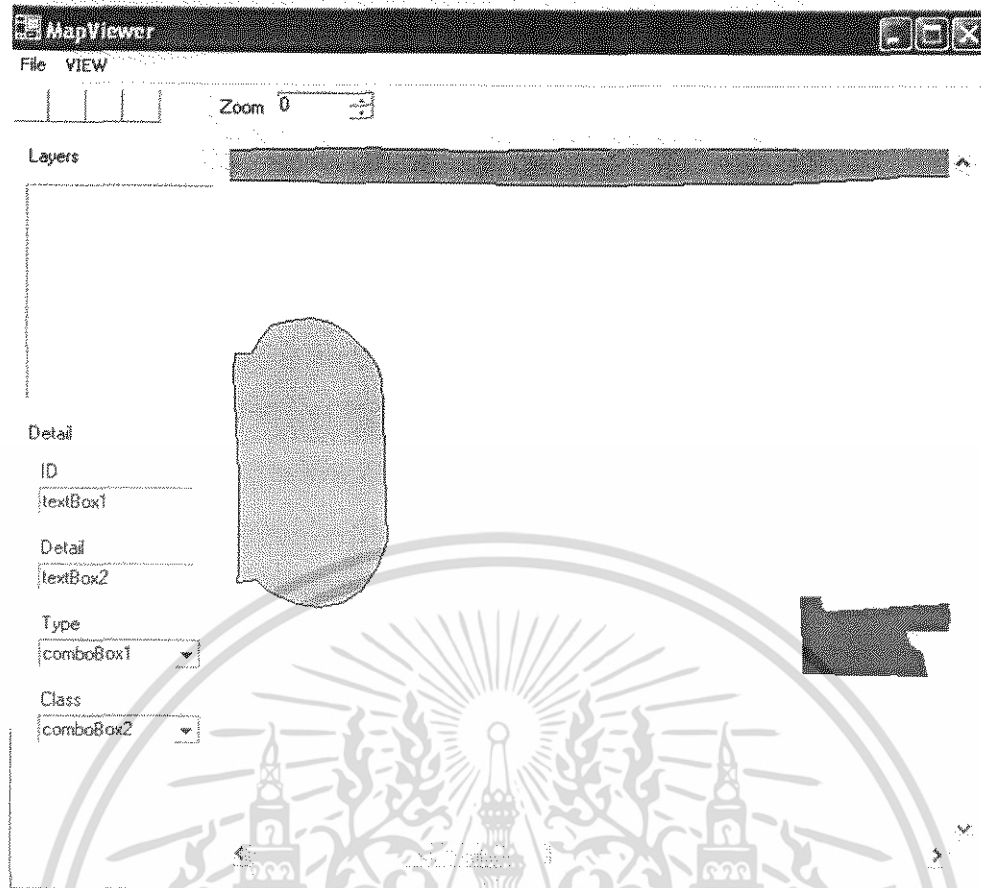


รูปที่ 4.4 แสดงการสร้างข้อมูลแผนที่

4.2 Output

Output ที่ได้จากโปรแกรม Map Generator and Converter จะสามารถ export ได้มาเป็นฐานข้อมูล GML และ SVG และไม่ว่าจะเป็นการนำข้อมูลเข้ามาหรือสร้างขึ้นใหม่ ก็จะสามารถ export ออกมาเป็นได้ตามความต้องการของผู้ใช้งาน ดังตัวอย่างดังรูปที่ 4.5 มาจากการสร้างภาพแผนที่ในรูปที่ 4.4 และ export โดยโปรแกรมได้มาเป็นไฟล์ SVG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงภาพที่ได้จากการสร้างภาพแผนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 บทสรุป

ความต้องการใช้แผนที่ร่วมกับอุปกรณ์อำนวยความสะดวกต่าง ๆ เริ่มมีมากขึ้น รวมทั้งเทคโนโลยีในปัจจุบันนี้ก้าวไปอย่างรวดเร็ว เราจึงต้องหา Software ที่ก้าวให้ทันพร้อมกันกับเทคโนโลยีที่นับวันยิ่งล้ำสมัยมากขึ้น พร้อมทั้งต้องมี Software ที่ต้องรองรับกับอุปกรณ์ที่ใช้จากอดีตมาจนถึงปัจจุบันได้ สำหรับโปรแกรม Map Generator and Converter ก็เป็นอีกทางเลือกหนึ่ง เพราะภาษาที่ใช้เป็นภาษาที่ยอมรับกันทั่วโลกด้วยมาตรฐานเดียวกัน เพราะว่าปัจจุบันภาษา XML เริ่มเข้ามามีบทบาทกับการทำงานด้านเอกสารมากขึ้น และภาษา XML ก็มีพัฒนาอย่างต่อเนื่อง และมีความยืดหยุ่นต่อการสร้างและนำไปใช้งาน

เราจะเห็นได้ว่าตัวโปรแกรม Map Generator and converter มีความสามารถในการสร้างและแปลงแผนที่ให้อยู่ในรูปแบบของ SVG (Scalable Vector Graphic) หรือ GML (Geography Markup Language) และให้เราได้รู้ถึงวิธีการเก็บข้อมูลในรูปแบบต่าง ๆ ที่เป็นมาตรฐาน เช่น SVG, GML และ Postgres

5.2 วิจารณ์สิ่งที่ได้จากปริญาานิพนธ์

ได้ช่วยในการสร้างงานด้านเอกสารให้สามารถที่จะทำการสร้างได้โดยง่าย โดยที่ผู้ใช้งานอาจไม่จำเป็นต้องรู้เรื่องโครงสร้างและไวยากรณ์ของภาษา GML และ SVG เลย หรือจะแก้ไขเอกสารก็ตามที่ และยังได้ข้อมูลเกี่ยวกับโครงสร้างของทั้งภาษา GML และ SVG นำมาเปรียบเทียบกัน เพื่อนำมาศึกษาและพัฒนาต่อไปให้เกิดประโยชน์สูงสุดได้

5.3 ปัญหาอุปสรรคและแนวทางในการแก้ไข

1. การแก้ไขแผนที่นั้นอาจจะมีผลคลาดเคลื่อนจากพิกัดโลกอยู่ เพราะในการแก้ไขนั้นเราจะใช้เมาส์เป็นตัวกำหนดจุดที่ต้องการแก้ไขในแต่ละจุด ซึ่งถ้าเป็น Line นั้นเส้นอาจจะหนาเกินกว่าที่เป็นพิกัดโลกจริง ซึ่งแนวทางการแก้ปัญหานี้คือ จะต้องใช้เทคโนโลยีที่สามารถติดต่อกับดาวเทียมเข้ามาช่วยในการ ยืนยันในตำแหน่งจริงที่ต้องการแก้ไขในแผนที่แล้วทำการกำหนดจุดตามตำแหน่งจริงของพิกัดโลก โดยใช้ดาวเทียมเป็นตัวส่งสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การที่ภาษา GML มีโครงสร้างที่หลากหลายทำให้การยึดหยุ่นในด้านของภาษา GML มีข้อจำกัดพอสมควร เพราะในขณะที่ภาษา GML มีโครงสร้างที่หลากหลาย ทำให้เรื่องยากที่จะเขียน XSLT ขึ้นมารองรับ schema ทั้งหมดของภาษา GML

5.4 แนวทางการพัฒนาต่อ

1. พัฒนาในส่วนของการแก้ไขแผนที่ให้มีความเที่ยงตรง โดยอ้างอิงจากพิภพโลก
2. พัฒนาในส่วนของ XSLT ให้รองรับ schema ได้เพิ่มขึ้น
3. สามารถเพิ่มเติมในส่วนของฟังก์ชันต่างๆของแผนที่ ให้มีความสามารถมากขึ้น เช่น การค้นหาสถานที่ หรือเส้นทางเดินรถ
4. เพิ่มความสามารถในด้านการตอบสนองการใช้งานเช่น เพิ่มเติมภาพเคลื่อนไหวในส่วนต่างให้มากขึ้น หรือเพิ่มการแสดงผลที่หลากหลายโดยใช้ความสามารถของภาษา SVG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] “GML_SVG”[Online]. Available :http://202.28.94.50/upload/GML_SVG.doc
- [2] “OpenGIS® Geography Markup Language (GML) Implementation Specification”[online]. Available :<http://www.opengis.org/legal/ipr.htm>
- [3] “PostgreSQL”[Online]. Available :http://iwis.pcd.go.th/TWIS/document/other/postgres.htm#_Toc63300
- [4] ศุภชัย สมพานิช 2546 คู่มือการเขียนโปรแกรมและใช้งาน Visual C#.NET ฉบับสมบูรณ์ สำนักพิมพ์ด้านสุทธาการพิมพ์ กรุงเทพฯ
- [5] ชวลิต จิตรทีปติสุนทร 2544 XML กับการพัฒนาโซลูชันระดับสูง สำนักพิมพ์สามย่าน.com กรุงเทพฯ
- [6] พงษ์ศักดิ์ ตียนันท์ “การนำเสนอข้อมูลสารสนเทศภูมิศาสตร์บนระบบ internet โดยใช้เทคโนโลยี SVG”[online]. Available : http://svg.hpcc.nectec.or.th/documents/SVG_Paper.pdf
- [7] สุเพชร จิระจรกุล “ลักษณะข้อมูลเชิงพื้นที่ (Spatial Characteristics)” [Online]. Available : <http://www.gis2mc.com>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้