รายการอ้างอิง

ภาษาไทย

คณะกรรมการประสานและส่งเสริมการพัฒนาระบบสารสนเทศทางภูมิศาสตร์. 2540. **แผนปฏิบัติการพัฒนาระบบสารสนเทศภูมิศาสตร์ในช่วงแผนพัฒนาเศรษฐกิจและสังคมแห่งชาติ ฉบับที่ 8 (พ.ศ.2540 - 2544)** : ส่วนที่ 2.

คณะกรรมการอนุกรรมการจัดทำ(ร่าง)นโยบายและแผนการพัฒนาระบบสารสนเทศทางภูมิศาสตร์ในคณะกรรมการประสานและส่งเสริมการพัฒนาระบบสารสนเทศภูมิศาสตร์. 2538. **เอกสาร นโยบายและแผนการพัฒนาระบบสารสนเทศทางภูมิศาสตร์ในช่วงแผนพัฒนาเศรษฐกิจและสังคมแห่งชาติ ฉบับที่ 8 (พ.ศ.2540 - 2544)** : ภาคผนวก 2.

พ.อ.เจนวิทย์ เหลืองอร่าม. 2538. **การเขียนโปรแกรมสำหรับ Applicaitons และ Applets ด้วย JAVA**. บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน) : V-VI

สรรเพชญ ชื้อนิธิไพศาล. ระบบสารสนเทศภูมิศาสตร์กับอินเตอร์เนต. **วิศวกรรมสาร ฉบับ ว.ส.ท. เทคโนโลยี** ปีที่ 52 เล่มที่ 3 (มีนาคม 2542) : 61-67

ภาษาอังกฤษ

Environmental System Research Institute, Inc. 19 April 1997. ESRI's GIS and the Internet [slides]. A Symposium on "GIS and the WWW", Department of Geomatics, The University of Melbourne.

Environmental System Research Institute, Inc. July 1998. ESRI Shapefile Technical Description. <http://www.esri.com/library/whitepapers/addl_lit.html>

Environmental System Research Institute, Inc. 1997. How ESRI uses the Internet/Intranet to deliver GIS on-line. <http://www.esri.com/library/whitepapers/addl_lit.html>

Environmental System Research Institute, Inc. 1997. Introduction to ARC/INFO : 2-22 – 2-30

Environmental System Research Institute, Inc. 1997. The future of GIS on the Internet . <http://www.esri.com/library/whitepapers/addl_lit.html>

Iestyn Polley. 19 April 1997. A Review of The Current Situation in Regards to GIS and the Internet [slides]. A Symposium on "GIS and the WWW", Department of Geomatics, The University of Melbourne.

Intergraph Corporation. GeoMedia Web Map. <http://www.intergraph.com/software/geo_map/geo_web.asp>

Intergraph Corporation. 19 April 1997. GeoMedia Web Map [slides]. A Symposium on "GIS and the WWW", Department of Geomatics, The University of Melbourne.

James D.Foley, Andries van Dam, Steven K.Feiner, John F.Hughes. 1990. Computer Graphics. Addison-Wesley Publishing Company, Inc : 34

Jamie Jaworski. 1996. JAVA DEVELOPER'S GUIDE. Sams.net Publishing : 16-19

John December, Mark Ginsburg. 1995. HTML & CGI. Sams.net Publishing : 383

MapInfo Corporation. MapInfo MapXtreme. <http://www.mapinfo.com/mapxtreme/index.html>

ภาคผนวก

ภาคผนวก ก

โครงสร้างฐานข้อมูล Shapefile

Shapefile Format.

Shapefile[1] คือรูปแบบหนึ่งของฐานข้อมูลสารสนเทศภูมิศาสตร์ ประกอบด้วยข้อมูลทางกราฟฟิกเก็บอยู่ในชุดของค่าพิกัด X,Y รวมทั้งข้อมูลเชิงอรรถาธิบาย โดยไม่มี Topology ทำให้จัดการกับข้อมูลได้ง่ายและสะดวก การแสดงผลรวดเร็ว และเพิ่มความสามารถในการแก้ไขข้อมูลได้ด้วยตัวเองจากการสร้างโปรแกรมประยุกต์ต่างๆ shapfile ยังสนับสนุนข้อมูลกราฟฟิกประเภทจุด เส้น และพื้นที่ ข้อมูลอรรถาธิบายเก็บอยู่ในรูปแบบ dbf มีการเชื่อมโยงแบบ หนึ่ง ต่อ หนึ่ง (one to one)

การสร้าง Shapefile

โดยทั่วไปสามารถสร้างได้ 4 วิธี ดังนี้

1) Export คือการใช้คำสั่ง export จากแหล่งข้อมูลอื่นผ่าน software เช่น ARC/INFO , PC ARC/INFO , Spatial Database Engine , Arcview GIS หรือ software อื่นๆที่เกี่ยวข้อง
2) Digitize เป็นวิธีสร้างขึ้นโดยตรงจากการ digitize ผ่านทาง Arcview GIS
3) Programming โดยใช้ภาษา macro ที่มีในซอฟท์แวร์ เช่น Avenue ใน Arcview GIS , Arc Macro Language (AML) ใน ARC/INFO , Simple Macro Language (SML) ใน PC ARC/INFO
4) การเขียน shapefile โดยตรงจากการโปรแกรมภาษาต่าง เช่น ภาษา C , Basic , Pascal , Java เป็นต้น ซึ่งต้องรู้รูปแบบโครงสร้างของ shapefile

---

[1] Environmental System Research Institute, Inc. ESRI Shapefile Technical Description. (http://www.esri.com/library/whitepapers/addl_lit.html)

คุณลักษณะของ Shapefile

shapefile ประกอบด้วย file มาตรฐาน 3 file ได้แก่

- Main File เป็นตัวเก็บข้อมูลกราฟฟิก โดยแต่ละ record เก็บชุดข้อมูลของจุดในแต่ละ feature
- Index File เก็บข้อมูลตำแหน่ง byte เริ่มต้นของแต่ละ record ใน Main file จากตำแหน่งเริ่มต้น (byte 0) ของ Main file และ ความยาวในแต่ละ record ใน Main file
- ตาราง dBase ประกอบด้วยข้อมูลอรรถาธิบาย 1 record ในตารางคือ 1 feature เป็นความสัมพันธ์แบบ 1 ต่อ 1 การเชื่อมโยงระหว่างข้อมูลกราฟฟิกกับข้อมูลอรรถาธิบาย อ้างอิงถึงกันจากหมายเลข record โดย record ของแต่ละ feature ใน Main file และข้อมูลอรรถาธิบายของ feature นั้นๆ ในตาราง dBase จะมีหมายเลข record เดียวกัน

ทั้ง Main file , Index file และ dBase file จะมีชื่อแฟ้มเดียวกัน โดยเริ่มจากตัวอักษร (a-Z,0-9) ตามด้วยตัวอักษรหรืออักขระพิเศษ (a-Z,0-9,_,-) แต่นามสกุลของแต่ละแฟ้มจะต่างกันตามชนิดของแฟ้ม ดังนี้

Main file	มีนามสกุลเป็น shp
Index file	มีนามสกุลเป็น shx
DBase file	มีนามสกุลเป็น dbf

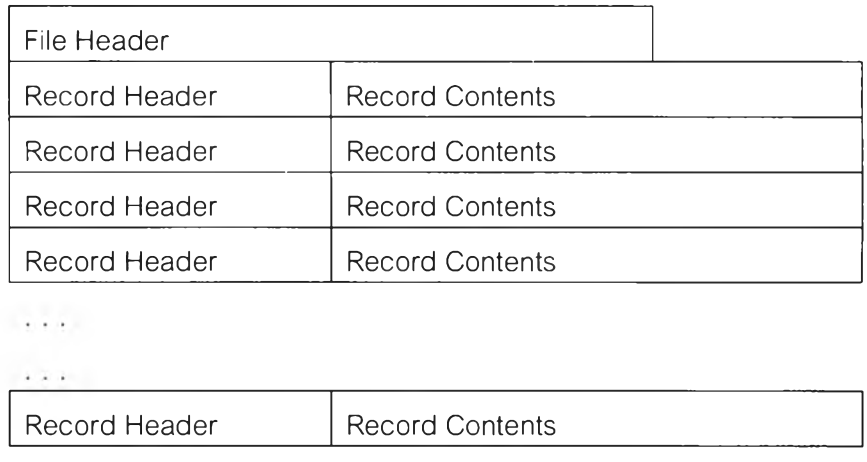
รูปแบบการเก็บข้อมูลใน shapefile จะอยู่ในรูปของตัวเลข 2 ชนิด คือ

- Integer : Signed 32-bit integer (4 bytes)
- Double : Signed 64-bit IEEE double-precision floating point number (8 bytes)

การเรียงตัวของ byte (Byte Order) จะมีทั้ง little endian (PC or Intel) byte order และ big endian (Sun or Motorola) byte order

โครงสร้างของ Main File

Main file ประกอบด้วย 2 ส่วน คือ File Header และ variable-length record โดยที่ File Header จะเก็บข้อมูลรายละเอียดของ shapefile โดยมีขนาดเนื้อที่ที่ใช้ในการเก็บข้อมูลคงที่ 100 bytes และ variable-length record ประกอบด้วย record header และ record content โดยที่ record header ใช้เนื้อที่ในเก็บข้อมูลคงที่ 8 bytes ส่วน record content จะใช้เนื้อที่ไม่แน่นอนขึ้นกับชนิดของ feature และจำนวนจุดของแต่ละ feature นั้นๆ



รูปที่ ก.1 โครงสร้างของ Main File

Main File Header : ใช้เนื้อที่ในการเก็บข้อมูลคงที่ 100 bytes ประกอบด้วยรายละเอียดของ shapefile ต่างๆ ดังตารางที่ ก.1

ตารางที่ ก.1 รายละเอียดของ Main File Header

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
Byte16	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
* ไม่ได้ใช้ จะมีค่า = 0 แต่จะสำรองเนื้อที่ไว้สำหรับโครงสร้างใหม่ในอนาคต				

File Length : เก็บค่าขนาดเนื้อที่ที่ใช้ในเก็บข้อมูลของ main file ทั้งหมด อยู่ในหน่วย 16-bit เช่น ถ้า geology.shp มีขนาดแฟ้ม 480 byte file length จะเท่ากับ 240

Shape Type : แสดงถึงประเภทข้อมูลกราฟฟิก ของ shapefile นั้นๆ ได้แก่

ค่า	ประเภทข้อมูล
0	ไม่มีค่า (Null Shape)
1	จุด (Point)
3	เส้น (PolyLine)
5	รูปปิด (Polygon)
8	หลายจุด (MultiPoint)

Bounding Box : เก็บค่าพิกัดขอบเขตข้อมูลกราฟฟิก โดยเก็บค่า Xmin , Ymin , Xmax , Ymax

Record Header : ใช้เนื้อที่ในการเก็บข้อมูลคงที่ 8 bytes โดยเก็บค่าหมายเลข record และเนื้อที่ที่ใช้ในการเก็บข้อมูลของแต่ละ record ดังตารางที่ ก.2

ตารางที่ ก.2 รายละเอียดของ Main File Record Header

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

หมายเลข record แรกจะเริ่มต้นที่ 1 เสมอ

Content Length : เก็บค่าขนาดเนื้อที่ที่ใช้ในเก็บข้อมูลของแต่ละ record อยู่ในหน่วย 16-bit เช่น ถ้า record แรกมีขนาดเนื้อที่ที่ใช้ 30 bytes content length จะเท่ากับ 15 bytes

Main File Record Header : ในส่วนนี้จะใช้เนื้อที่ในการเก็บไม่คงที่ ขึ้นอยู่กับประเภทข้อมูลของ shapefile และจำนวนจุดของแต่ละ feature นั้นๆ ดังนี้

Null Shapes : shapefile ที่มีค่า shape type = 0 ไม่มีค่าพิกัด ซึ่งจะเกิดขึ้นจากการที่สร้าง shapefile ขึ้นมาใหม่ โดยยังไม่ได้ลงข้อมูลกราฟฟิกใดๆ หรือข้อมูลที่สร้างขึ้นเพื่อเตรียมที่จะ digitize

ตารางที่ ก.3 Null Shape Record Contents

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

Point : shapefile ที่มีค่า shape type = 1 เก็บข้อมูลประเภทจุด โดย 1 record ต่อ 1 จุด ข้อมูลที่เก็บอยู่ในรูปของพิกัด XY โดยใช้ชนิดตัวเลขแบบ Double ในเก็บค่าพิกัด X และ Y ดังนี้

```
Point
{
        Double   X    // X coordinate
        Double   Y    // Y coordinate
}
```

ตารางที่ ก.4 Point Record Contents

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

MultiPoint : shapefile ที่มีค่า shape type = 8 เก็บข้อมูลประเภทหลายจุด ใน 1 record อาจมีมากกว่า 1 จุด โดยเก็บเป็นชุดข้อมูลของจุดที่มีข้อมูลเชิงอรรถาธิบายเหมือนกันใน record เดียวกัน

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

Box : เก็บข้อมูลขอบเขตของจุดในแต่ละ record นั้น โดยเก็บค่า Xmin , Ymin , Xmax , Ymax

ตารางที่ ก.5 MultiPoint Record Contents

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	8	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumPoints	NumPoints	Integer	1	Little
Byte 40	Points	Points	Point	NumPoints	Little

PolyLine : shapefile ที่มีค่า shape type = 3 เก็บข้อมูลประเภทเส้น ใน 1 record อาจมีมากกว่า 1 เส้น โดยเก็บเป็นชุดข้อมูลของจุดในแต่ละเส้นที่มีข้อมูลเชิงอรรถาธิบายเหมือนกันใน record เดียวกัน

```
PolyLine
{
    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 : เก็บข้อมูลขอบเขตของเส้นในแต่ละ record นั้น โดยเก็บค่า Xmin , Ymin , Xmax , Ymax

NumParts : จำนวนเส้นในแต่ละ record

NumPoints : จำนวนจุดทั้งหมดในแต่ละ record

Parts : ค่าดัชนีตำแหน่งเริ่มต้นของแต่ละเส้น โดยค่าเริ่มต้นของเส้นแรก = 0

Points : ชุดของจำนวนจุดของแต่ละเส้น

ตารางที่ ก.6 PolyLine Record Contents

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	8	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

หมายเหตุ : $X = 44 + 4 \times NumParts$

Polygon : shapefile ที่มีค่า shape type = 5 เก็บข้อมูลประเภท polygon ใน 1 record อาจมีมากกว่า 1 polygon โดยเก็บเป็นชุดข้อมูลของจุดในแต่ละ polygon ที่มีข้อมูลอรรถาธิบายเหมือนกันใน record เดียวกัน

```
Polygon
{
    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 : เก็บข้อมูลขอบเขตของ polygon ในแต่ละ record นั้น โดยเก็บค่า Xmin , Ymin , Xmax , Ymax

NumParts : จำนวน polygon ในแต่ละ record

NumPoints : จำนวนจุดทั้งหมดในแต่ละ record

Parts : ค่าดัชนีตำแหน่งเริ่มต้นของแต่ละ polygon โดยค่าเริ่มต้นของรูปปิดแรก = 0

Points : ชุดของจำนวนจุดของแต่ละ polygon

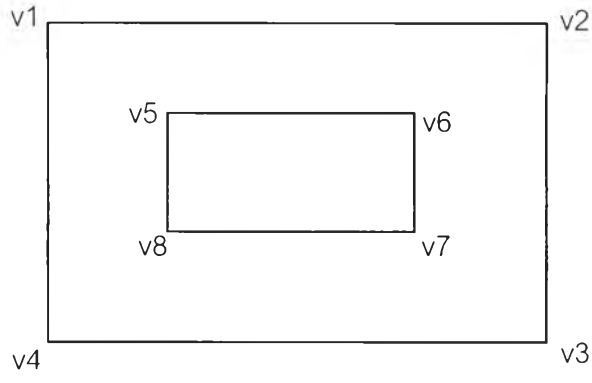
ตารางที่ ก.7 Polygon Record Contents

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	8	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

หมายเหตุ : $X = 44 + 4 \times NumParts$

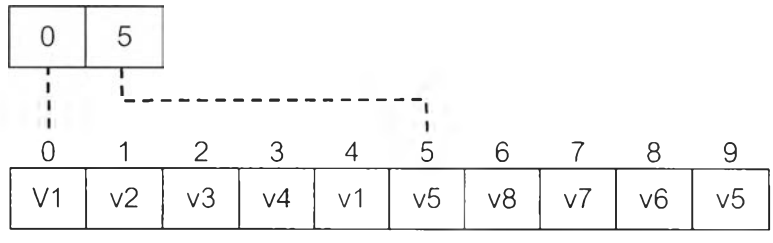
คุณสมบัติของ Polygon ใน shapefile

1. polygon ต้องปิด กล่าวคือ จุดเริ่มต้นและจุดสิ้นสุดของ polygon ต้องเป็นจุดเดียวกัน
2. การจัดลำดับของ part ใน polygon ที่มีมากกว่า 1 part จะจัดอย่างไรก็ได้ เช่น polygon หนึ่งประกอบด้วย 3 parts คือ 1 , 2 , 3 อาจเก็บชุดพิกัดของ 3 2 1 หรือ 2 3 1 หรือ 1 3 2 ก็ได้
3. polygon แต่ละ polygon จะไม่ตัดกัน ส่วนของ polygon ใดจะไม่อยู่ในส่วนของ polygon อื่นๆ จะสัมผัสกันได้เฉพาะจุดใน polygon เท่านั้น
4. การบันทึกชุดพิกัดของจุดใน polygon จะยึดหลักที่ว่า polygon ที่สนใจจะอยู่ทางขวามือของเส้นรอบรูป ดังนั้นทิศทางของจุดที่บันทึกจะเป็นไปในทิศทางตามเข็มนาฬิกา แต่ถ้า polygon มีมากกว่า 1 part ทิศทางของ polygon ในจะมีทิศทางทวนเข็มนาฬิกา



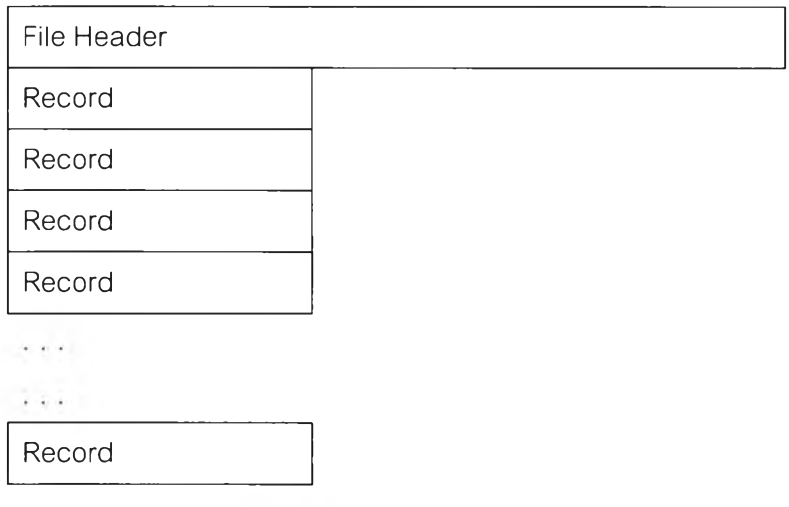
รูปที่ ก.2 ตัวอย่างการบันทึกข้อมูลของ Polygon

จากรูป polygon ดังกล่าวมี 2 parts และ จำนวนจุดทั้งหมด 10 จุด



โครงสร้างของ Index File

Index file ประกอบด้วย 2 ส่วนคือ File Header และ Index record โดย File Header มีขนาดเนื้อที่ที่ใช้เก็บข้อมูลคงที่ 100 และมีโครงสร้างเช่นเดียวกับ File Header ของ Main File



รูปที่ ก.3 โครงสร้างของ Index File

Index Record : ในแต่ละ record จะเก็บค่า offset และ ความยาวของแต่ละ record ใน Main File ดังตาราง

ตารางที่ ก.8 Description of Index Records

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

Offset : ตัวเลขของตำแหน่ง byte แรกของแต่ละ record ใน Main File โดยนับจาก byte ที่ 0 ของ Main File ค่า offset อยู่ในหน่วยตัวเลข 16-bit ดังนั้น ค่า offset ของ record แรก จะเท่ากับ 50 เนื่องจาก file header มีความยาว 100 bytes

Content length : ค่าความยาวของแต่ละ record ใน Main File ซึ่งจะมีค่าเดียวกับค่า content length ใน Main File Record Header

ตาราง dBase

ตาราง dBase เป็นฐานข้อมูลสำหรับเก็บข้อมูลเชิงอรรถาธิบายของ feature ต่างๆ หรือเก็บข้อมูลที่ไว้ใช้สำหรับเชื่อมโยงไปยังฐานข้อมูลตารางอื่นๆที่เกี่ยวข้อง โครงสร้างฐานข้อมูลตาราง dBase เป็นโครงสร้างมาตราฐาน ที่ใช้ในโปรแกรมประยุกต์ทั่วไปที่ทำงานกับฐานข้อมูลลักษณะตารางดังกล่าว ใน Windows และ Dos

ลักษณะเฉพาะของตาราง dBase ใน shapefile มี 3 ข้อดังนี้

1) ชื่อแฟ้มข้อมูลจะต้องมีชื่อเช่นเดียวกับ main file และ index file
2) ความสัมพันธ์ของข้อมูลเชิงตำแหน่งกับข้อมูลเชิงอรรถาธิบายเป็นแบบ 1 ต่อ 1 หรือ ใน 1 record ต่อ 1 feature
3) ลำดับของ record ในตาราง dBase จะต้องมีลำดับเช่นเดียวกันใน main file และ index file

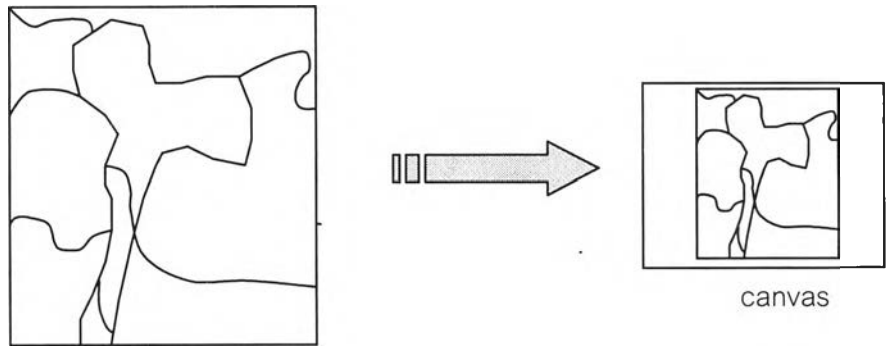
ภาคผนวก ข

สูตร และ อัลกอริทึมที่ใช้ในคำสั่งพื้นฐานการแสดงผล

คำสั่งพื้นฐานการแสดงผลข้อมูลเชิงตำแหน่งและรับค่าพิกัดต่างๆจะอยู่ที่ canvas พร้อมทั้ง Class ที่เก็บค่าพิกัดของชั้นข้อมูลต่างๆ โดย Class ดังกล่าวคือ Polygon ซึ่งเป็น Class มาตรฐานของ Java โดยเก็บเป็นชุดของค่าพิกัดทำให้เข้าใจง่าย เนื่องจากจะมีลักษณะการเก็บเช่นเดียวกับใน shapefile ในงานวิจัยได้จัดสร้างคำสั่งพื้นฐานไว้ 5 คำสั่งโดยมีอัลกอริทึมในการคำนวณค่าพิกัดจากระบบพิกัดโลกเป็นระบบพิกัด Canvas ดังนี้

fullView

ค่าพิกัดที่ใช้ในฐานข้อมูล GIS จะเป็นค่าพิกัดโลก การแสดงผลจึงต้องทำการทอนค่าพิกัดให้อยู่ในระบบพิกัดเดียวกับหน้าจอแสดงผล ดังรูปที่ ข.1

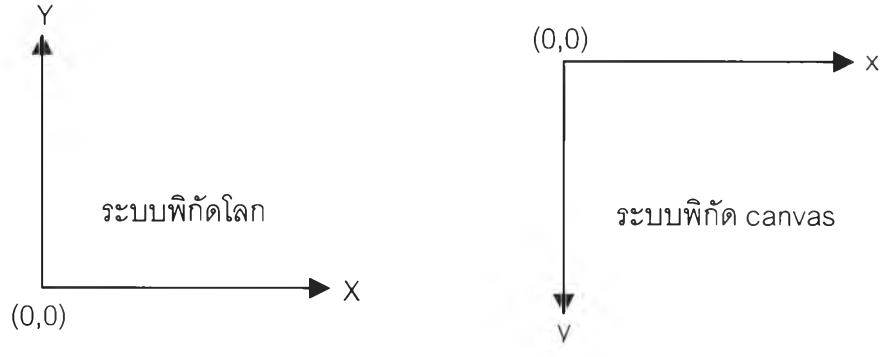


ค่าพิกัดโลก

รูปที่ ข.1 การทอนค่าพิกัดโลกเป็นพิกัดของ canvas

การแสดงผลข้อมูลเชิงตำแหน่งจากระบบพิกัดโลกมาเป็นระบบพิกัดของ canvas กระทำโดยการหาจุดศูนย์กลางของข้อมูลพิกัดโลก และจุดศูนย์กลางของ canvas เป็นจุดโยงยึดสำหรับการถ่ายระบบพิกัด หลังจากนั้นคำนวณหามาตราส่วนระหว่างความกว้างของขอบเขตข้อมูลพิกัดโลกกับความกว้างของ canvas และ ความยาวของขอบเขตข้อมูลพิกัดโลกกับความยาวของ canvas โดยเลือกใช้ค่าที่น้อยที่สุด เพื่อให้ข้อมูลแสดงบน canvas ได้ครบถ้วน

fullView เป็นคำสั่งที่ใช้แสดงผลข้อมูลเชิงตำแหน่งทั้งหมดของแผนที่ ที่หน้าจอแสดงผลของ canvas โดยต้องทำการย่อขยายค่าพิกัดโลกเพื่อให้สามารถนำมาแสดงผลใน canvas ได้ทั้งหมด พร้อมทั้งแปลงค่าระบบพิกัดให้มาอยู่ในระบบเดียวกับ canvas



รูปที่ ข.2 พิกัดเริ่มต้นและทิศทางของแกนพิกัด X , Y

สมการที่ใช้ในการแปลงค่าพิกัดจากระบบพิกัดโลกซึ่งเป็นพิกัด X,Y เป็นพิกัดของ canvas คือ

$$x = m(X - X_b) + x_b \quad \text{_____ (1)}$$

$$y = \Delta y - [m(Y - Y_b) + y_b] \quad \text{_____ (2)}$$

$$m = \text{minimum}\left(\frac{\Delta x}{\Delta X}, \frac{\Delta y}{\Delta Y}\right) \quad \text{_____ (3)}$$

โดยที่ ΔX = ค่าขอบเขตความยาวของข้อมูลพิกัดโลก

ΔY = ค่าขอบเขตความกว้างของข้อมูลพิกัดโลก

Δx = ค่าความยาวของ canvas

Δy = ค่าความกว้างของ canvas

m = ตัวคูณมาตราส่วน

x,y = พิกัดของ canvas

x_b,y_b = พิกัดกึ่งกลางของ canvas

X,Y = พิกัดโลก

X_b,Y_b = พิกัดกึ่งกลางของข้อมูลเชิงตำแหน่งโดยหาจากค่าขอบเขต (boundary)

จาก (1) และ (2) จะได้ $x = mX + a$ _____ (4)

$y = b - mY$ _____ (5)

โดยที่ $a = mX_b + x_b$ _____ (6)

$b = y_b - mY_b$ _____ (7)

จากสมการ (3) และ (4) จะได้ค่าพิกัดที่ใช้ใน canvas และในทางกลับสามารถหาค่าพิกัดโลกจากตำแหน่งใดใน canvas ได้ดังนี้

$$X = \frac{(x - a)}{m} \quad \text{_____ (8)}$$

$$Y = \frac{(b - y)}{m} \quad \text{_____ (9)}$$

zoomIn

เป็นคำสั่งที่ใช้ขยายภาพ หรือทำการลดขอบเขตข้อมูลเชิงตำแหน่งของพิกัดโลกให้เล็กลง เมื่อเทียบกับขอบเขตเดิมที่ทำการแสดงผลอยู่บน canvas ในขณะนั้น ผู้ใช้สามารถเลือกบริเวณที่ต้องการแสดงผลได้ด้วยการเลือกขอบเขตพิกัดที่แสดงผลใหม่

จากการกำหนดตำแหน่งจากตัวชี้ตำแหน่ง (cursor) ได้ตำแหน่งเป็น (x_1,y_1) กับ (x_2,y_2) และจากสมการ (7) และ (8) ได้ค่าพิกัดโลกเป็น (X_1,Y_1) กับ (X_2,Y_2) ตามลำดับ

$$\Delta x = | x_2 - x_1 | \quad \text{_____ (10)}$$

$$\Delta y = | y_2 - y_1 | \quad \text{_____ (11)}$$

$$\Delta X = | X_2 - X_1 | \quad \text{_____ (12)}$$

$$\Delta Y = | Y_2 - Y_1 | \quad \text{_____ (13)}$$

$$x_b = \frac{(x_1 + x_2)}{2} \quad \text{_____ (14)}$$

$$X_b = \frac{(X_1 + X_2)}{2} \quad \text{_____ (15)}$$

$$y_b = \frac{(y_1 + y_2)}{2} \quad \text{_____ (16)}$$

$$Y_b = \frac{(Y_1 + Y_2)}{2} \quad \text{_____ (17)}$$

นำค่าที่ได้จากสมการ (10) (11) (12) และ (13) แทนค่าในสมการ (3) จะได้ค่าตัวคูณมาตราส่วนใหม่ (m) และนำค่าที่ได้จากสมการ (14) (15) (16) (17) และ m แทนค่าในสมการ (6) และ (7) จะได้ค่าพารามิเตอร์ a และ b หลังจากนั้นนำมาคำนวณหาค่าพิกัด canvas ใหม่ด้วยสมการ (4) และ (5)

zoomOut

เป็นคำสั่งที่ใช้ย่อภาพ หรือทำการเพิ่มขอบเขตข้อมูลเชิงตำแหน่งของพิกัดโลกให้มากขึ้นเมื่อเทียบกับขอบเขตเดิมที่ทำการแสดงผลอยู่บน canvas ในขณะนั้น โดยหลักการคือการลดค่าตัวคูณมาตราส่วนลงนั่นเอง เช่นกำหนดให้ลดลง 0.8 จากค่าเดิม แล้วนำไปแทนค่าในสมการ (6) และ (7) จะได้ค่าพารามิเตอร์ของ a และ b ใหม่ หลังจากนั้นนำไปแทนค่าในสมการ (4) และ (5) จะได้ค่าพิกัด canvas ใหม่ ผลลัพธ์ทำให้ภาพที่แสดงมีขนาดเล็กลง

pan

เป็นคำสั่งที่ใช้เลื่อนตำแหน่งของภาพที่แสดงอยู่ ผู้ใช้ทำการกำหนดทิศทางและระยะทางในการเลื่อนตำแหน่งแสดงผลของภาพโดยใช้ตัวชี้ตำแหน่งกำหนดจุดเริ่มต้น และกำหนดจุดสุดท้ายเพื่อกำหนดระยะทางและทิศทาง จากกำหนดจุดสองจุดดังกล่าว จะได้

$$dx = x_2 - x_1$$

$$dy = y_2 - y_1$$

จาก dx และ dy จะได้ทำให้ค่า X_b , Y_b และค่าพารามิเตอร์ a และ b เปลี่ยนไปดังนี้

$$a = a + dx$$

$$b = b + dy$$

$$X_b = \frac{(x_b - a)}{m}$$

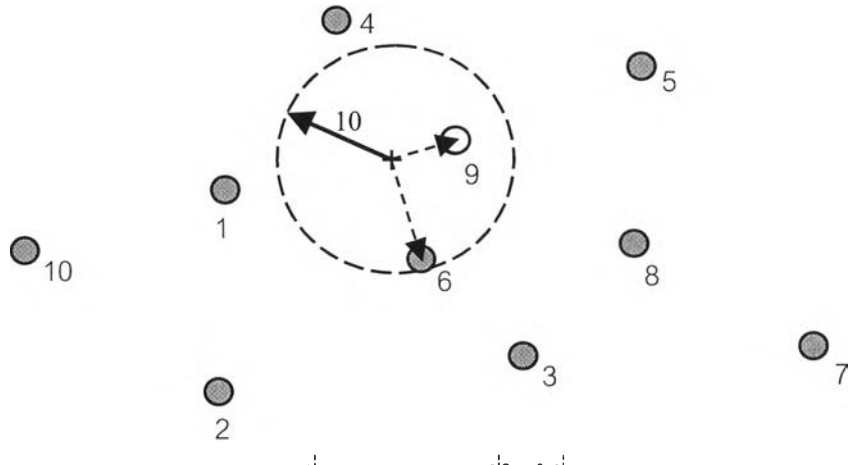
$$Y_b = \frac{(b - y_b)}{m}$$

หลังจากนั้นคำนวณค่าพิกัด canvas ด้วยสมการ (4) และ (5)

identify

นอกจากคำสั่งที่ใช้เกี่ยวข้องกับการแสดงผลข้อมูลเชิงตำแหน่งแล้ว คำสั่งพื้นฐานหนึ่งที่สำคัญทาง GIS คือจะต้องทำการค้นคืนข้อมูลจากฐานข้อมูลได้ ในงานวิจัยนี้ได้สร้างคำสั่ง identify และกำหนดระยะ buffer ของการเลือกแต่ละ feature ไว้เท่ากับ 10 หน่วย Pixel เพื่อให้ผู้ใช้ทำการค้นคืนข้อมูลอรรถาธิบายจาก features ที่ต้องการ ด้วยการกำหนดตำแหน่ง feature ที่สนใจจากตัวชี้ตำแหน่ง โดยแบ่งเป็น 3 features ดังนี้

- Point :
  - กำหนดระยะ buffer ของการเลือกไว้ 10 หน่วย ถ้าไม่มี feature อยู่ในรัศมี 10 หน่วยจากจุดที่เลือกค่าพิกัด ถือว่าไม่มีการเลือก feature ดังกล่าว
  - หาระยะทางระหว่างจุดพิกัดที่ผู้ใช้กำหนดกับพิกัดของจุดทุกๆจุดในชั้นข้อมูล ถ้ามีระยะน้อยกว่า 10 หน่วยจะเก็บค่า record ของ feature นั้น และใช้ค่าระยะทางเป็นค่า buffer ใหม่
  - หาระยะทางจากจุดอื่นๆที่เหลือ ถ้ามีค่าน้อยกว่าระยะ buffer จะใช้ค่า record ของ feature อันใหม่แทน
  - ทำการตรวจสอบเช่นนี้จนครบทุกจุดในชั้นข้อมูล จะได้จุดที่อยู่ใกล้ที่สุดและ record ของ feature ดังกล่าว
  - หลังจากนั้นส่งค่า record ดังกล่าวไปค้นคืนในฐานข้อมูล dbf ด้วย CGI และนำผลลัพธ์ที่ได้มาแสดงผล



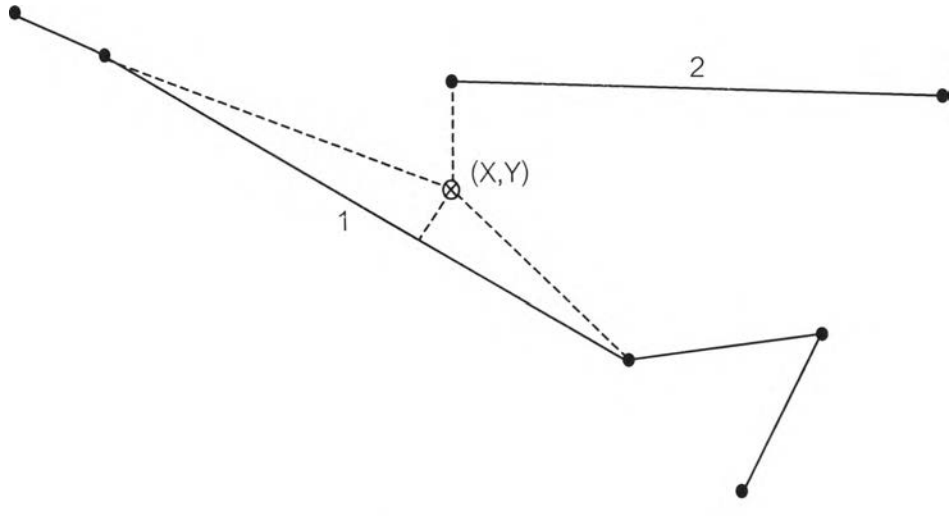
รูปที่ ข.3 การหาจุดที่ใกล้ที่สุด

♦ Line :

— กำหนดระยะ buffer ของการเลือกไว้ 10 หน่วย

— ตรวจสอบค่าพิกัดที่เลือกว่าอยู่ในขอบเขตของเส้นหรือไม่ ถ้าอยู่ในขอบเขตให้หาระยะทางที่สั้นที่สุดจากจุดถึงเส้นนั้น ถ้าไม่ ตรวจสอบกับเส้นต่อไป

— หาระยะทางระหว่างจุดที่เลือกกับจุดเริ่มต้น (จุดแรก) และจุดต่อไป (จุดสอง) ของเส้นนั้น บันทึกค่าระยะทางที่สั้นที่สุดจาก 2 ค่าที่ได้ เป็น d



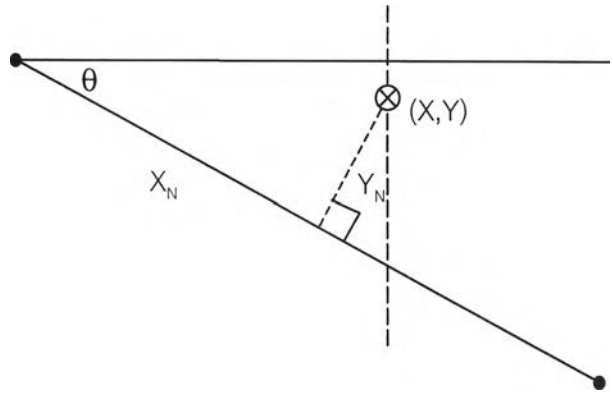
รูปที่ ข.4 การหาเส้นที่ใกล้ที่สุด

— หาค่าทิศทางของช่วงเส้นจาก จุดแรกถึงจุดสอง (θ)

— ทำการแปลงค่าพิกัดของจุดพิกัดที่เลือกโดยผู้ใช้ ไปอยู่ระบบพิกัดใหม่ของช่วงเส้นดังกล่าว โดยให้ทิศทางของช่วงเส้นนั้นเป็นแกน X

— จุดพิกัดใหม่ที่ได้คือ $X_N = X \cos \theta + Y \sin \theta$ และ $Y_N = Y \cos \theta - X \sin \theta$

— ตรวจสอบ X_N อยู่ในช่วงของระยะทางระหว่างจุดแรกกับจุดที่สองในช่วงเส้นที่พิจารณา หรือไม่



รูปที่ ข.5 การหาระยะทางเส้นตั้งฉาก

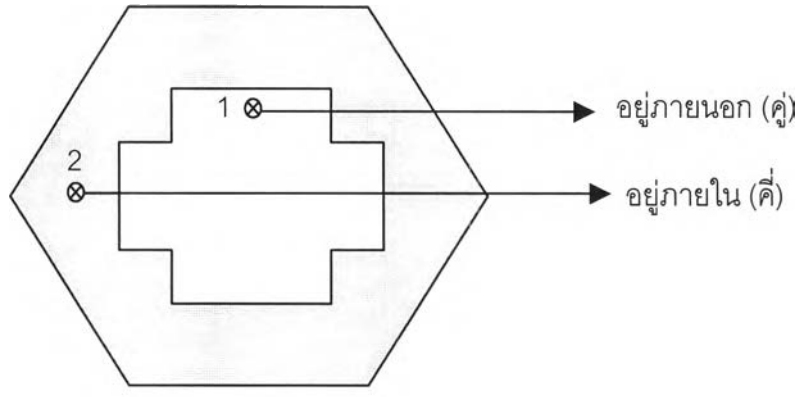
- ถ้าอยู่ ค่าระยะทางที่สั้นที่สุดคือค่าของ Y_N ถ้าไม่ ค่าระยะทางที่สั้นที่สุดคือ d และตรวจสอบว่ามีค่าน้อยกว่าระยะ buffer หรือไม่ ถ้าน้อยกว่าให้บันทึกเป็นค่า buffer ใหม่ และบันทึกค่า record ของ feature นั้น ถ้าไม่ทำการเปรียบเทียบกับจุดที่เหลือในเส้นนั้นต่อไป
- เปรียบเทียบจนครบทุกเส้น นำค่า record ที่ได้ ส่งผ่าน CGI เพื่อไปค้นคืนข้อมูลอรรถาธิบายที่ต้องการมาแสดงผล

◆ Polygon :

- การหาจุดพิกัดที่สนใจอยู่ใน polygon หรือไม่
- พิจารณาจุดพิกัดที่สนใจ ลากเส้นตรงจากจุดดังกล่าวตัด polygon แล้วหาจำนวนจุดตัดว่าเป็นเลขคู่หรือ เลขคี่
- ถ้าเป็นเลขคู่แสดงว่าจุดพิกัดที่สนใจอยู่ภายนอก polygon ถ้าเป็นเลขคี่แสดงว่าอยู่ภายใน polygon
- ใน object polygon มี function ในการหาจุดว่าอยู่ภายนอกหรือภายใน polygon โดยใช้ even-odd insideness rule (odd-parity rule[1])
- ในกรณีที่ polygon เป็นลักษณะวงแหวนหรือมีมากกว่า 1 วง ให้ตรวจสอบดูว่าจุดที่สนใจอยู่ภายใน polygon มากกว่า 1 วงหรือไม่

---

[1] Foley, VanDam, Feiner, Hughes. Computer Graphics Principles and Practice. (Addison-Wesley Publishing Compan, Inc):34



รูปที่ ข.6 การหาจุดอยู่ภายใน polygon

- ถ้ามากกว่าแสดงว่าอยู่ภายนอก polygon
- ถ้าพบว่าจุดที่สนใจอยู่ภายใน polygon และมีเพียง 1 วง บันทึกค่า record ของ feature หยุดการค้นหา
- ส่งค่า record เพื่อไปค้นคืนข้อมูลอรรถาธิบายโดยใช้ CGI

ภาคผนวก ค

Source Code

test.html

```
<HTML> <HEAD> <TITLE>Test MapClass</TITLE> </HEAD> <BODY>
<applet code="startApp" width=650 height=400 name="startApp">
<PARAM NAME=amount VALUE="5">
<PARAM NAME=maxRecord VALUE="4322">
<PARAM NAME=maxPart VALUE="1">
<PARAM NAME=Xmin VALUE="642350">  <PARAM NAME=Ymin VALUE="1482500">
<PARAM NAME=Xmax VALUE="710550">  <PARAM NAME=Ymax VALUE="1543550">

<PARAM NAME=name0 VALUE="School">
<PARAM NAME=file0 VALUE="bma_edu">
<PARAM NAME=type0 VALUE="1">
<PARAM NAME=rec0 VALUE="1243">
<PARAM NAME=part0 VALUE="1">
<PARAM NAME=symbol0 VALUE="T">

<PARAM NAME=name1 VALUE="Hospital">
<PARAM NAME=file1 VALUE="bma_hos">
<PARAM NAME=type1 VALUE="1">
<PARAM NAME=rec1 VALUE="153">
<PARAM NAME=part1 VALUE="1">
<PARAM NAME=symbol1 VALUE="+">

<PARAM NAME=name2 VALUE="Office">
```

```
<PARAM NAME=file2 VALUE="bma_off">
<PARAM NAME=type2 VALUE="1">
<PARAM NAME=rec2 VALUE="33">
<PARAM NAME=part2 VALUE="1">
<PARAM NAME=symbol2 VALUE="F">

<PARAM NAME=name3 VALUE="Street">
<PARAM NAME=file3 VALUE="bma_str">
<PARAM NAME=type3 VALUE="3">
<PARAM NAME=rec3 VALUE="2005">
<PARAM NAME=part3 VALUE="1">
<PARAM NAME=symbol3 VALUE="-">

<PARAM NAME=name4 VALUE="Bangkok">
<PARAM NAME=file4 VALUE="bma">
<PARAM NAME=type4 VALUE="5">
<PARAM NAME=rec4 VALUE="38">
<PARAM NAME=part4 VALUE="1">
<PARAM NAME=symbol4 VALUE="-">

This Web page requires a <B>Java</B>-enabled Web Browser.
</applet> </BODY> </HTML>
```

startApplet.java

```
import java.applet.*;
import java.awt.*;
import java.net.*;
import java.io.*;
import aCanvas;
```

```
import layerPanel;
import symantec.itools.awt.ImageButton;

public class startApplet extends Applet {
    ImageButton imgZoom,imgZLast,imgPan,imgExt,imgId;
    Button clear;
    TextField test;
    Choice iden;
    aCanvas c;
    layerPanel pl;
    String xc, yc, str[], strshp;

    InputStream dat[];
    int type[],rec[],part[],ntype[];
    int n,mrec,mprt;
    int Xmin,Ymin,Xmax,Ymax;

    public void init() {
        setup();
        setGUI();
        setShape();
        readShape();
        c.update();
        c.fullView();
    }

    public void setup() {
        n = Integer.parseInt(getParameter("amount"));
        mrec = Integer.parseInt(getParameter("maxRecord"));
        mprt = Integer.parseInt(getParameter("maxPart"));
```

```
        Xmin = Integer.parseInt(getParameter("Xmin"));
        Ymin = Integer.parseInt(getParameter("Ymin"));
        Xmax = Integer.parseInt(getParameter("Xmax"));
        Ymax = Integer.parseInt(getParameter("Ymax"));
        str = new String[n];
        dat = new InputStream[n];
        type = new int[n];
        ntype = new int[n];
        rec = new int[n];
        part = new int[n];

        try {
            for (int i=0;i<n;i++) {
                str[i] = getParameter("file"+String.valueOf(i));
                URL url = new URL(getDocumentBase(),str[i]+".shp");
                dat[i] = url.openStream();
                type[i] = Integer.parseInt(getParameter("type"+String.valueOf(i)));
                ntype[i] = Integer.parseInt(getParameter("ntype"+String.valueOf(i)));
                rec[i] = Integer.parseInt(getParameter("rec"+String.valueOf(i)));
                part[i] = Integer.parseInt(getParameter("part"+String.valueOf(i)));
        }} catch (IOException e) {
            System.out.println("File open error: " + e.getMessage());
        }
    }

    public void setGUI() {
        c = new aCanvas();
        Panel pn = new Panel();
        pl = new LayerPanel(n,c);
        test = new TextField("test");
```

```
imgZoom = new ImageButton();        imgZLast = new ImageButton();
imgPan = new ImageButton();         imgExt = new ImageButton();
imgId = new ImageButton();

clear = new Button("Clear");
iden = new Choice();
for (int i=0;i<n;i++) {
    pl.addName(i,getParameter("name"+String.valueOf(i)));
    iden.addItem(getParameter("name"+String.valueOf(i)));
}
iden.setBackground(Color.lightGray);
pn.setBackground(Color.yellow);
pn.resize(150,25);

pn.add(imgZoom);   imgZoom.reshape(1,1,22,22);
pn.add(imgZLast);  imgZLast.reshape(25,1,22,22);
pn.add(imgPan);    imgPan.reshape(50,1,22,22);
pn.add(imgExt);    imgExt.reshape(75,1,22,22);
pn.add(imgId);     imgId.reshape(100,1,22,22);
pn.add(iden);     pn.add(clear);   pn.add(test);
try {
    imgZoom.setImageURL(new URL(getDocumentBase(),"zoomin.gif"));
    imgZLast.setImageURL(new URL(getDocumentBase(),"zoomout.gif"));
    imgPan.setImageURL(new URL(getDocumentBase(),"pan.gif"));
    imgExt.setImageURL(new URL(getDocumentBase(),"zoomlayer.gif"));
    imgId.setImageURL(new URL(getDocumentBase(),"id.gif"));
} catch (java.net.MalformedURLException error) {
}
imgZoom.setScaleMode(true);
```

```
        imgZLast.setScaleMode(true);
        imgPan.setScaleMode(true);
        imgExt.setScaleMode(true);
        imgId.setScaleMode(true);

        setLayout(new BorderLayout());
        pl.updatePanel();
        add("North",pn);
        add("Center",c);
        add("West",pl);
        resize(600,450);
    }

    public void setShape() {
        // set CGI Path
        c.setCGIPath("http://161.200.86.68/project/webdb.dll");
        // set color
        Color color[] = new Color[6];
        color[0] = Color.yellow;
        color[1] = Color.blue;
        color[2] = Color.green;
        color[3] = Color.red;
        color[4] = Color.magenta;
        color[5] = Color.pink;

        // set variable of canvas.
        c.setNLayer(n);
        c.setType(type);
        c.setRecord(rec);
        c.setPart(part);
```

```
        c.setMax(mrec,mprt);
        c.setMape(Xmin,Ymin,Xmax,Ymax);
        for (int i=0; i<n; i++)
            c.addColor(i,color[i]);

        // initialize.
        for (int i=0;i<n;i++) {
            c.addSymbol(i,getParameter("symbol"+String.valueOf(i)));
            c.addFile(i,str[i]);
        }
    }

    public void readShape(){
        try{
        for (int i=0;i<n;i++)
            c.setPoly(i,dat[i],type[i],rec[i],part[i]);
        }
        catch (IOException e){
        System.out.println("File open error: " + e.getMessage());
        }
    }

    public boolean action(Event ev,Object obj) {
        if (ev.target instanceof Button) {
            if(ev.target == clear) {
                for (int i=0;i<n;i++)
                    pl.setChkState(i,false);
                c.clear();
            }
        }
```

```
        else if(ev.target instanceof ImageButton) {
            if(ev.target == imgZoom) {
                c.zoomIn();
            }
            else if(ev.target == imgZLast) {
                c.zoomOut();
            }
            else if(ev.target == imgPan) {
                c.panDone();
            }
            else if(ev.target == imgExt) {
                c.fullView();
            }
            else if(ev.target == imgId) {
                int k = iden.getSelectedIndex();
                String txt = String.valueOf(k) + " "
                    + String.valueOf(iden.getSelectedItem());
                test.setText(txt);
                if (pl.getChkState(k) == true)
                    c.identify(k);
            }
        }

        return true;
    }
}
```

layerPanel.java

```
import java.awt.*;
```

```
import java.awt.event.*;
import aCanvas;

public class layerPanel extends Panel implements ItemListener {
    private Checkbox chkLayer[];
    private int n;
    aCanvas c;

    public layerPanel(int i,aCanvas target) {
        n = i;
        c = target;
        setBackground(Color.lightGray);
        chkLayer = new Checkbox[i];
    }

    public void addName(int i,String name) {
        chkLayer[i] = new Checkbox(name);
        chkLayer[i].addItemListener(this);
    }

    public void setChkState(int i,boolean tbf) {
        chkLayer[i].setState(false);
    }

    public boolean getChkState(int i) {
        return chkLayer[i].getState();
    }

    public void updatePanel() {
        GridBagLayout gridbag = new GridBagLayout();
```

```
        GridBagConstraints gc = new GridBagConstraints();
        setLayout(gridbag);
        gc.fill = GridBagConstraints.HORIZONTAL;
        gc.weightx = 0.0;
        gc.gridwidth = GridBagConstraints.REMAINDER;
        for (int i=0; i<n; i++) {
            gridbag.setConstraints(chkLayer[i],gc);
            add(chkLayer[i]);
        }
        gc.gridwidth = GridBagConstraints.REMAINDER;
        Panel empty = new Panel();
        gc.gridheight = GridBagConstraints.REMAINDER;
        gc.fill = GridBagConstraints.BOTH;
        gc.weighty = 0.1;
        empty.setBackground(Color.lightGray);
        gridbag.setConstraints(empty,gc);
        add(empty);
    }

    public void itemStateChanged(ItemEvent e) {
        if (e.getSource() instanceof Checkbox) {
            for (int i=0; i<n; i++)
            c.setSelect(i, chkLayer[i].getState());
            c.update();
        }
    }
}
```

aCanvas.java

```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import symantec.itools.awt.MultiList;
import java.net.*;
import java.util.StringTokenizer;

public class aCanvas extends Canvas implements MouseListener, MouseMotionListener
{
    private Color color[];
    private String Psym[];
    private boolean pan, zoomW, iden, select[];
    private String file[];
    private Wqry wqry;
    private DataInputStream dat;

    private int n, type[], rec[], maxPart[], part[][];
    private Polygon polys[][][];
    private Polygon polyn[][][];
    private int Xmin,Ymin,Xmax,Ymax;

    private static int Gwidth, Gheight;
    private static int startX, endX, startY, endY;
    private static int Xgb, Ygb;    // X,Y mean of ground.
    private static int Xcb, Ycb;    // X,Y mean of canvas.
    private static int xConst, yConst;  // X,Y constant to use in transformation.
    private static double mScale;   // transformation Scale.
    private int qryLayer;           // Number of Layer to query.
    double scaleOut;
```

```
public aCanvas() {
    setBackground(Color.black);
    pan = false;
    zoomW = false;
    iden = false;
    scaleOut = 0.8;
    addMouseMotionListener(this);
    addMouseListener(this);
    wqry = new Wqry();
}

public void setMape(int xmi, int ymi, int xmx, int ymx) {
    Xmin = xmi;
    Ymin = ymi;
    Xmax = xmx;
    Ymax = ymx;
    Gwidth = Xmax - Xmin;
    Gheight = Ymax - Ymin;
}

public void setNLayer(int j) {
    n = j;
    type = new int[n];
    rec = new int[n];
    maxPart = new int[n];
    file = new String[n];
    color = new Color[n];
    select = new boolean[n];
    Psym = new String[n];
    for (int i=0; i<n; i++)
```

```
            select[i]=false;
    }

    public void setType(int types[]) {
        type = types;
    }

    public void setRecord(int recs[]) {
        rec = recs;
    }

    public void setPart(int prt[]) {
        maxPart = prt;
    }

    public void setMax(int mrec,int mprt) {
        part = new int[n][mrec];
        polys = new Polygon[n][mrec][mprt];
        polyn = new Polygon[n][mrec][mprt];
        for(int i=0; i<n; i++) {
            switch (type[i]) {
            case 1:
            case 8:
                polys[i][0][0] = new Polygon();
                break;
            default:
                for(int j=0; j<rec[i]; j++) {
                    for(int k=0; k<mprt; k++) {
                        polys[i][j][k] = new Polygon();
                        polyn[i][j][k] = new Polygon();
```

```
                    }
                }
                break;
            } // End switch
        } // End for i
}

public void addRecPart(int i,int j,int prt) {
    part[i][j] = prt;
}

private void addPoly(int i,int j,int k,Polygon poly){
    polys[i][j][k] = poly;
}

public void setPoly(int numP,InputStream shapeInput,int typeP
    ,int recP,int partP) throws IOException {

    dat = new DataInputStream(shapeInput);
    int x,y,pnts,prts;
    int parts[] = new int[recP];
    int prt[] = new int[partP+1];
    dat = new DataInputStream(shapeInput);

    dat.skipBytes(100);
    switch (typeP){
    case 1:
        for (int i=0;i<recP;i++) {
            dat.skipBytes(12);
            x = (int)(swapDouble());
```

```
            y = (int)(swapDouble());
            polys[numP][0][0].addPoint(x,y);
        }
        dat.close();
        break;
    case 3:
    case 5:
        for (int i=0;i<recP;i++) {
            dat.skipBytes(44);
            prts = swapInt();
            addRecPart(numP,i,prts);
            pnts = swapInt();
            for (int j=0;j<prts;j++)
                prt[j] = swapInt();
            prt[prts] = pnts;
            for (int j=0;j<prts;j++) {
                int np = prt[j+1]-prt[j];
                for (int k=0;k<np;k++) {
                    x = (int)(swapDouble());
                    y = (int)(swapDouble());
                    polys[numP][i][j].addPoint(x,y);
                }
            }
        }
        dat.close();
        break;
    case 8:
        for (int i=0;i<recP;i++) {
            dat.skipBytes(44);
            pnts = swapInt();
```

```
                //parts[i] = pnts;
                addRecPart(numP,i,pnts);
                for (int j=0;j<pnts;j++) {
                    x = (int)(swapDouble());
                    y = (int)(swapDouble());
                    polys[numP][0][0].addPoint(x,y);
                }
            }
            dat.close();
            break;
        default:
            dat.close();
            break;
        }
        shapeInput.close();
    }

    private int swapInt(){
        try{
            int i = dat.readInt();
            int b1 = (i << 24) & 0xFF000000;
            int b2 = (i << 8) & 0x00FF0000;
            int b3 = (i >> 8) & 0x0000FF00;
            int b4 = (i >> 24) & 0x000000FF;
            return (b1 | b2 | b3 | b4);
        }
        catch (IOException e) {
            System.out.println("File Data read error: "+e.getMessage());
        }
        return -1;
```

```
}

private double swapDouble() {
    try {
        double dd = dat.readDouble();
        long lb = Double.doubleToLongBits(dd);
        long lb1 = (lb << 56) & 0xFF00000000000000L;
        long lb2 = (lb << 40) & 0x00FF000000000000L;
        long lb3 = (lb << 24) & 0x0000FF0000000000L;
        long lb4 = (lb << 8) & 0x000000FF00000000L;
        long lb5 = (lb >> 8) & 0x00000000FF000000L;
        long lb6 = (lb >> 24) & 0x0000000000FF0000L;
        long lb7 = (lb >> 40) & 0x000000000000FF00L;
        long lb8 = (lb >> 56) & 0x00000000000000FFL;
        long lbb = (lb1|lb2|lb3|lb4|lb5|lb6|lb7|lb8);
        return Double.longBitsToDouble(lbb);
    }
    catch (IOException e) {
        System.out.println("File Data read error: " + e.getMessage());
    }
    return -1.111111;
}

public void addColor(int i, Color cl) {
    color[i] = cl;
}

private int getcHeight() {
    return getSize().height;
}
```

```
private int getcWidth() {
    return getSize().width;
}

public void setCGIPath(String path) {
    wqry.setCGI(path);
    //path = "http://161.200.86.68/project/gisquery.dll"
}

public void paint(Graphics g) {
    for (int i=0; i<n; i++) {
        if(select[i] == true) {
            g.setColor(color[i]);
            switch (type[i]) {
            case 1:  // Point
            case 8:  // MultiPoint
                for (int j=0; j<rec[i]; j++)
                g.drawString(Psym[i],polyn[i][0][0].xpoints[j],polyn[i][0][0].ypoints[j]);
                break;
            case 3:   //arc
                for(int j=0; j<rec[i]; j++) {
                    for(int k=0; k<part[i][j]; k++) {
                        g.drawPolyline(polyn[i][j][k].xpoints
                            ,polyn[i][j][k].ypoints, polyn[i][j][k].npoints);
                    }
                }
                break;
            case 5:  //Polygon
                for (int j=0; j<rec[i]; j++) {
```

```
                    for(int k=0; k<part[i][j]; k++)
                        g.drawPolygon(polyn[i][j][k]);
                }
                break;
            default:
                break;
            } // End switch.
        } // End If clause.
    } // End Loop i
}

private double getScale() {
    return Math.min((double)getcHeight()/(double)Gheight
        , (double)getcWidth()/(double)Gwidth);
}

public void update(){
    repaint();
}

public void setSelect(int i,boolean sel) {
    select[i] = sel;
}

public void addSymbol(int i,String str) {
    Psym[i] = str;
}

public void addFile(int i, String str) {
    file[i] = str;
```

```
}

public void clear() {
    for (int i=0 ; i<n; i++)
        select[i] = false;
    update();
}

public void identify(int i) {
    iden = true;
    qryLayer = i;
    this.setCursor(new Cursor(Cursor.HAND_CURSOR));
}

public void panDone() {
    pan = true;
    this.setCursor(new Cursor(Cursor.MOVE_CURSOR));
}

public void zoomIn() {
    zoomW = true;
    this.setCursor(new Cursor(Cursor.CROSSHAIR_CURSOR));
}

public void zoomOut() {
    Polygon poly = new Polygon();
    int x,y;
    mScale = scaleOut * mScale;
    xConst = (int)(Xcb - mScale * Xgb);
    yConst = (int)(Ycb + mScale * Ygb);
```

```
for (int i=0; i<n; i++) {
    switch (type[i]) {
    case 1:
    case 8:
        poly = new Polygon();
        for (int j=0; j<polyn[i][0][0].npoints; j++) {
            x = (int)(mScale * polys[i][0][0].xpoints[j] + xConst);
            y = (int)(yConst - mScale * polys[i][0][0].ypoints[j]);
            poly.addPoint(x,y);
        }
        polyn[i][0][0] = poly;
        break;
    case 3:
    case 5:
        for (int j=0; j<rec[i]; j++) {
            for (int k=0; k<part[i][j]; k++) {
                poly = new Polygon();
                for (int l=0; l<polyn[i][j][k].npoints; l++) {
                    x = (int)(mScale * polys[i][j][k].xpoints[l] + xConst);
                    y = (int)(yConst - mScale * polys[i][j][k].ypoints[l]);
                    poly.addPoint(x,y);
                }
                polyn[i][j][k] = poly;
            }
        }
        break;
    default:
        break;
    } // End switch type[].
```

```
    } // End for loop.
    repaint();
}

public void fullView() {
    this.setCursor(new Cursor(Cursor.DEFAULT_CURSOR));
    mScale = getScale();
    Xgb = (Xmax + Xmin) / 2;
    Ygb = (Ymax + Ymin) / 2;
    Xcb = getcWidth() / 2;
    Ycb = getcHeight() / 2;
    xConst = (int)(Xcb - (mScale * Xgb));
    yConst = (int)(Ycb + (mScale * Ygb));
    int x,y;

    for (int i=0; i<n; i++) {
        switch (type[i]) {
        case 1:
        case 8:
            polyn[i][0][0] = new Polygon();
            for (int j=0; j<polys[i][0][0].npoints; j++) {
                x = (int)(mScale * polys[i][0][0].xpoints[j] + xConst);
                y = (int)(yConst - mScale * polys[i][0][0].ypoints[j]);
                polyn[i][0][0].addPoint(x,y);
            }
            break;
        case 3:
        case 5:
            for (int j=0; j<rec[i]; j++) {
                for (int k=0; k<part[i][j]; k++) {
```

```
                    polyn[i][j][k] = new Polygon();
                    for (int l=0; l<polys[i][j][k].npoints; l++) {
                        x = (int)(mScale * polys[i][j][k].xpoints[l] + xConst);
                        y = (int)(yConst - mScale * polys[i][j][k].ypoints[l]);
                        polyn[i][j][k].addPoint(x,y);
                    }
                }
            }
            break;
        default:
            break;
        } // End switch type[].
    } // End for loop.
    repaint();
}

private void panUpdate(int Xdist, int Ydist) {
    xConst = xConst + Xdist;
    yConst = yConst + Ydist;
    Xgb = (int)((Xcb - xConst) / mScale);
    Ygb = (int)((yConst - Ycb) / mScale);

    for (int i=0; i<n; i++) {
        switch (type[i]) {
        case 1:
        case 8:
            polyn[i][0][0].translate(Xdist,Ydist);
            break;
        case 3:
        case 5:
```

```
                for (int j=0; j<rec[i]; j++) {
                    for (int k=0; k<part[i][j]; k++)
                        polyn[i][j][k].translate(Xdist,Ydist);
                }
                break;
            default:
                break;
            } // End switch type[].
        } // End for loop.
        repaint();
    }

    private void zoomInUpdate(int x1, int y1, int x2, int y2) {
        Polygon poly;
        int x,y;
        int X1 = (int)(Xgb + (x1 - Xcb)/mScale);
        int Y1 = (int)(Ygb + (Ycb - y1)/mScale);
        int X2 = (int)(Xgb + (x2 - Xcb)/mScale);
        int Y2 = (int)(Ygb + (Ycb - y2)/mScale);
        mScale = Math.min((double)getcHeight()/(double)(Y1-Y2)
            , (double)getcWidth()/(double)(X2-X1));
        Xgb = (X1 + X2) / 2;
        Ygb = (Y1 + Y2) / 2;
        Xcb = getcWidth() / 2;
        Ycb = getcHeight() / 2;
        xConst = (int)(Xcb - (mScale * Xgb));
        yConst = (int)(Ycb + (mScale * Ygb));

        for (int i=0; i<n; i++) {
            switch (type[i]) {
```

```
            case 1:
            case 8:
                poly = new Polygon();
                for (int j=0; j<polyn[i][0][0].npoints; j++) {
                    x = (int)(mScale * polys[i][0][0].xpoints[j] + xConst);
                    y = (int)(yConst - mScale * polys[i][0][0].ypoints[j]);
                    poly.addPoint(x,y);
                }
                polyn[i][0][0] = poly;
                break;
            case 3:
            case 5:
                for (int j=0; j<rec[i]; j++) {
                    for (int k=0; k<part[i][j]; k++) {
                        poly = new Polygon();
                        for (int l=0; l<polyn[i][j][k].npoints; l++) {
                            x = (int)(mScale * polys[i][j][k].xpoints[l] + xConst);
                            y = (int)(yConst - mScale * polys[i][j][k].ypoints[l]);
                            poly.addPoint(x,y);
                        }
                        polyn[i][j][k] = poly;
                    }
                }
                break;
            default:
                break;
            } // End switch type[].
        } // End for loop.
        repaint();
    }
```

```
private void query(int x,int y) {
    int rec1 = -1;
    int prt1 = 0;
    double dist = 10;
    double x1,y1,d1,x2,y2,d2,d12;
    double x3,y3,d3;
    Graphics g = this.getGraphics();
    boolean sel = false;
    g.setColor(Color.white);

    switch (type[qryLayer]) {
    case 1:
    case 8:
        for (int i=0;i<polyn[qryLayer][0][0].npoints;i++) {
            x1 = (double)polyn[qryLayer][0][0].xpoints[i];
            y1 = (double)polyn[qryLayer][0][0].ypoints[i];
            d1 = Math.sqrt( (x1-x)*(x1-x) + (y1-y)*(y1-y));
            if (d1 < dist) {
                rec1 = i;
                dist = d1;
                sel = true;
            }
        }
        break;
    case 3:
        for (int i=0; i<rec[qryLayer]; i++) {
            for (int j=0; j<part[qryLayer][i]; j++) {
                if (polyn[qryLayer][i][j].getBounds().contains(x,y)==true) {
                    for (int k=0;k<polyn[qryLayer][i][j].npoints-1;k++) {
```

```
                    x1 = (double)polyn[qryLayer][i][j].xpoints[k];
                    y1 = (double)polyn[qryLayer][i][j].ypoints[k];
                    x2 = (double)polyn[qryLayer][i][j].xpoints[k+1];
                    y2 = (double)polyn[qryLayer][i][j].ypoints[k+1];
                    d12 = Math.sqrt((x1-x2)*(x1-x2) + (y1-y2)*(y1-y2));
                    d1 = Math.sqrt((x1-x)*(x1-x) + (y1-y)*(y1-y));
                    d2 = Math.sqrt((x2-x)*(x2-x) + (y2-y)*(y2-y));
                    d3 = Math.min(d1,d2);
                    x3 = (double)(x-x1)*(x2-x1)/d12 + (double)(y-y1)*(y2-y1)/d12;
                    y3 = (double)(y-y1)*(x2-x1)/d12 - (double)(x-x1)*(y2-y1)/d12;
                    if(0<x3 && x3<d12)
                        d3 = Math.abs(y3);
                    if (d3 < dist) {
                        dist = d3;
                        rec1 = i;
                        prt1 = j;
                        sel = true;
                    } //End If(d3<dist).
                } //End loop k.
            } // End If contain(x,y).
        } //End loop j.
    } //End loop i.
    break;
case 5:
    for (int i=0;i<rec[qryLayer];i++) {
        if(part[qryLayer][i] == 1) {
            if (polyn[qryLayer][i][0].contains(x,y) == true) {
                rec1 = i;
                prt1 = 0;
                i = rec[qryLayer];
```

```
                    sel = true;
                } //End if contains.
            } //End If part[][] == 1.
            else {
                int inside = 0;
                for (int j=0;j<part[qryLayer][i];j++) {
                    if (polyn[qryLayer][i][j].contains(x,y) == true) {
                        inside++;
                        prt1 = j;
                    } //End if contains.
                } //End loop j.
                if (inside==1) {
                    rec1 = i;
                    i = rec[qryLayer];
                    sel = true;
                } //End if inside == 1.
            } //End else part[][] <> 1.
        } //End loop i.
        break;
default:
        break;
} //End switch.

if (sel==true) {
        switch (type[qryLayer]) {
        case 1:
        case 8:
            g.drawString(Psym[qryLayer],polyn[qryLayer][0][0].xpoints[rec1]
                ,polyn[qryLayer][0][0].ypoints[rec1]);
            break;
```

```
            case 3:
                g.drawPolyline(polyn[qryLayer][rec1][prt1].xpoints
                    ,polyn[qryLayer][rec1][prt1].ypoints
                    ,polyn[qryLayer][rec1][prt1].npoints);
                break;
            case 5:
                g.drawPolygon(polyn[qryLayer][rec1][prt1]);
                break;
            default:
                break;
        }} //End if sel=true and end switch.

        if (rec1 != -1) {
            wqry.updateText(file[qryLayer]+".dbf",String.valueOf(rec1));
            //wqry.searchText("bma.dbf","3","2");
            //wqry.showDatabase("bma.dbf");
            wqry.show();
            wqry.toFront();
        } //End if.
    }

    public void mouseClicked(MouseEvent e) {
        if(iden==true) {
            iden = false;
            query(e.getX(),e.getY());
            this.setCursor(new Cursor(Cursor.DEFAULT_CURSOR));
        }
    }

    public void mouseEntered(MouseEvent e) {
```

```
}

public void mouseExited(MouseEvent e) {
}

public void mousePressed(MouseEvent e) {
    Graphics g = this.getGraphics();
    g.setColor(Color.white);
    startX = endX = e.getX();
    startY = endY = e.getY();
    if (pan==true)
        g.drawLine(startX,startY,endX,endY);
    else if(zoomW==true)
        g.drawRect(Math.min(startX,endX),Math.min(startY,endY)
            ,Math.abs(startX-endX),Math.abs(startY-endY));
}

public void mouseReleased(MouseEvent e) {
    this.setCursor(new Cursor(Cursor.DEFAULT_CURSOR));
    if (pan==true) {
        pan = false;
        int Xdist = endX - startX;
        int Ydist = endY - startY;
        panUpdate(Xdist,Ydist);
    }
    else if(zoomW==true) {
        zoomW = false;
        Math.min(startX,endX);
        Math.min(startY,endY);
        zoomInUpdate(Math.min(startX,endX),Math.min(startY,endY)
```

```
                , Math.max(startX,endX),Math.max(startY,endY));
        }
    }

    public void mouseDragged(MouseEvent e) {
        Graphics g = this.getGraphics();
        g.setColor(Color.white);
        g.setXORMode(getBackground());
        int tempX, tempY;
        tempX = endX;
        tempY = endY;
        endX = e.getX();
        endY = e.getY();
        if (pan==true) {
            g.drawLine(startX,startY,tempX,tempY);
            g.drawLine(startX,startY,endX,endY);
        }
        else if (zoomW==true) {
            g.drawRect(Math.min(startX,tempX),Math.min(startY,tempY)
                ,Math.abs(tempX-startX),Math.abs(tempY-startY));
            g.drawRect(Math.min(startX,endX),Math.min(startY,endY)
                ,Math.abs(endX-startX),Math.abs(endY-startY));
        }
    }

    public void mouseMoved(MouseEvent e) {
    }
}
```

Wqry.java

```
class Wqry extends Frame {
    symantec.itools.awt.MultiList mList;
    int row,cols;
    URL url;
    String httpAddress;
    DataInputStream inStream;
    StringTokenizer st;

    Wqry() {
        // Init Controls
        setLayout(new BorderLayout(0,0));
        addNotify();
        setSize(getInsets().left+getInsets().right+324,
        getInsets().top+getInsets().bottom+369);
        mList = new symantec.itools.awt.MultiList(2);
        mList.setBounds(getInsets().left+0,
        getInsets().top+0, 324, 369);
        add("Center",mList);

        // Set Title of Frame
        setTitle("Data Results");
    }

    void setCGI(String cgiAddress){
        httpAddress = cgiAddress;
        // cgiAddress = "http://161.200.86.68/project/gisquery.dll"
    }

    void updateText(String file,String rec) {
```

```
        row = 0;
        mList.clear();
        String queryString,line;
        setTitle ("Query in " + file + " at record " + rec);
        queryString = httpAddress + "?a=1&b=" + file + "&c=" + rec;

        // Set Item Name
        mList.setColumns(2);
        String[] tempString = new String[2];
        tempString[0] = new String("Fields");
        tempString[1] = new String("Query_Data");
        mList.setHeadings(tempString);

        try {
            url = new URL(queryString);
            BufferedReader buff = new BufferedReader(
                new InputStreamReader(url.openStream()));
            line = buff.readLine();
            st = new StringTokenizer(line,"|");
            while (st.hasMoreTokens()) {
                mList.addTextCell(row,0,st.nextToken());
                mList.addTextCell(row,1,st.nextToken());
                mList.redraw();
                row++;
            }
            buff.close();
        }catch (MalformedURLException ex) {
        }catch (IOException ex) {
        }
    }
```

```
    public boolean handleEvent(Event event) {
        if(event.id == Event.WINDOW_DESTROY) {
            setVisible(false);  //hide the Frame
            return true;
        }
        return super.handleEvent(event);
    }
}
```

ประวัติผู้วิจัย

นายสรรเพชญ ชื้อนิธิไพศาล เกิดวันที่ 15 สิงหาคม พ.ศ.2516 มีภูมิลำเนาอยู่ที่ อำเภอเมือง จังหวัดพิษณุโลก สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตร์บัณฑิต ภาควิชาวิศวกรรมสำรวจ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2537 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตร์มหาบัณฑิต ที่ภาควิชาวิศวกรรมสำรวจ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2538