GIS COMPONENTS AND MAP VISUALIZATION OBJECTS*

LUO Yingwei, WANG Xiaolin and XU Zhuoqun

Dept. of Computer Science and Technology, Peking University, Beijing, P.R.China, 100871

ABSTRACT

In component-based WebGIS, map visualization objects are cornerstones in developing applications. From constructing user interfaces, displaying maps and human-computer interactions to displaying query results and making subject maps, almost all developing work is based on map visualization objects. In this paper, a component-based WebGIS system Geo-Union is introduced, which is built by COM/DCOM technology and can run in Internet/WWW environment. After briefly describing the architecture and the functional partition of Geo-Union, the design and implementation of map visualization objects are described and discussed in details. Map visualization objects in Geo-Union include a map visualization control and three programming objects (map object, display layer object and dynamic layer object). Map visualization control is in charge of displaying map, interacting with users and coordinating other components. The three programming objects are used to manage map data and its display properties. At last, a sample in Visual Basic environment is given to illustrate the use of Geo-Union components and its map visualization objects.

KEY WORDS: Component, WebGIS, Geo-Union, Map Visualization Objects

1. INTRODUCTION

Internet is changing the means of data access and release. WebGIS, which processes geographical spatial data in Internet, is developing rapidly, along with the movement of Internet and Web technique. Just because of the development and application, Internet should become the running platform of further GIS, and Web-based GIS applications are new urgent demands of most GIS users [1].

Component affords a new software construction mode, which is more efficient, agile and powerful than existing technique of object-orientation or traditional modularization. For example, developers could combine and reuse binary modules that independently

* Corresponding author: LUO Yingwei, lyw@pku.edu.cn.

exploited by different individual or groups, which dramatically simplify and expedite the development. Component is independent of language and hardware, and can run on Web. Moreover, this technique provides much more flexibility for application development [2][3].

Right now, there are too many WebGIS products in market, mainly including: Internet Map Server (IMS) and MapObject from ESRI, the former is used to release map on Web, and the last is a component library for further development of GIS applications [4]; MapInfo ProServer and MapX from MapInfo, their capabilities are similar to ESRI's IMS and MapObject [5]; MapGuide from Autodesk [6]; GeoMedia Web Map from Intergraph [7], GeoSurf from GeoStar [8] and so on. Those WebGIS products are different from their own design and implementation, but their basic thought is the same. For the implementation techniques, all of them adopt CGI/Serverlet technique, Plug-in technique, CORBA/DCOM technique, Java Applet technique and so on, or integrate several above techniques. For the implementation mode, there are two models. The first is thin client model. At server side, spatial data is mainly in vector format and most functions are completed there; but at client side, results are displayed in image. In thin client model, each operation of users must interact with server, so there are too many conversations and data transmissions between client and server. The second is fat client model. Spatial data is transferred from server to client in vector format, and most functions are complete in client. In fat client model, computing capability in client is uncertain and there exist many network security leaks, so most WebGIS products only implement some basic functions, such as elementary map visualization and simple spatial query [9].

Assigning functions in reason and improving performance are two key issues for making WebGIS more practicable. In fact, some WebGIS products, especially component-based WebGIS products, have already achieved a high practicability. But for a market reason, their design and implementation technologies are not open. So at first, we analyze the modeling technique of component-based WebGIS, construct a practicable, multi-level WebGIS system Geo-Union, ant explore its architecture, composition and functional partition of components. In component-based WebGIS system, application development becomes to a procedure of assembling WebGIS components according to application requirements. Map visualization objects are cornerstones in developing WebGIS applications. From constructing user interfaces, displaying maps and human-computer interactions, to displaying query results and making subject maps, almost all developing work is based on or around map visualization objects. So in section 3, the design and implementation of map visualization objects are described and discussed in details. At last, a sample developed in Visual Basic is given to illustrate the use of Geo-Union and its map visualization objects.

2. COMPONENT-BASED WebGIS - Geo-Union

Geo-Union is a GIS platform developed by spatial information lab in dept. of computer science and technology, Peking University. Geo-Union has a multi-level Client/Server architecture, which is implemented by principle of ORDB and component techniques. Geo-Union provides an object-oriented, extensible GIS component library for further GIS application developers. Geo-Union can be used in both stand-alone environment and network environment.

2.1 The Architecture of Geo-Union

Component model is a primary approach to deepen the functions of WebGIS. To make the system more clear, Geo-Union can be divided into four layers: Geo-Union application layer, Geo-Union component layer, Geo-Union service layer and Geo-Union storage layer, where service layer has different units to provide both client services and server services. Figure1 shows the architecture [10][11]. Hierarchical spatial component object model can distribute GIS functions in network reasonably and make the system reusable, as well as provide efficiency approach for further development and integration with other systems.

(1) Storage layer is the ground of Geo-Union. Storage layer is responsible for storage and management of both spatial data and non-spatial data based on ORDB. The main problems solved at this layer are how to represent and store spatial data, and how to maintain relationships among spatial data.

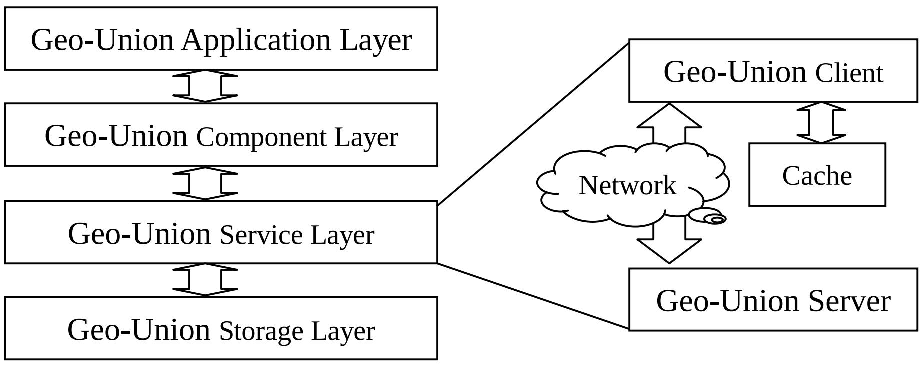


Figure 1 the Architecture of Geo-Union

(2) Service layer is in charge of spatial data access and process, which can be divided into another two parts: Geo-Union client provides data access and process services to component layer, and Geo-Union server provides data access and process services to Geo-Union client through interacting with storage layer. Geo-Union server can manage different spatial data resources, and also can reply to different spatial data requests from different clients. Geo-Union client and Geo-Union server are two independent parts but

have close relationship with each other. Geo-Union server provides the services of data access, spatial index, basic spatial query, transaction process, data share and so on. Geo-Union client provides different GIS tools and further development functions to component layer based on the services from Geo-Union server. Cache is an important unit of Geo-Union client, which is imported to reduce network load and improve response speed of the system. Using Geo-Union client, we can develop a simulation server, which can reduce network load and improve response speed through its cache [11].

(3) Component layer provides a rich set of services (components) to develop domain-oriented GIS application systems for further developers. Component layer provides interface of GIS functions to users, but the implementation details are completed in service layer. Component layer exists as a component library, and servers as a bridge between users and service layer. Component layer provides function-explicit and reusable interface components for users according to the functions of service layer.

(4) The work in application layer is to exploit application systems for different special domains by assembling and integrating Geo-Union components. These application systems can be running both in desktop and network environment.

2.2 The Functional Partition of Geo-Union Component

Component-based WebGIS makes it possible that application developers, data promulgators and spatial database engine vendors can construct domain-oriented GIS application systems together. Moreover, domain-oriented GIS components can be built conveniently according to specified models in one domain, and putted into component library at any time, thus used by other users. All of these will make the system possess high reusability.

Most of functions are implemented in service layer. In order to get high efficiency of reusability and high flexibility of assembly, it is important to give a clear functional partition and design a right architecture for service layer.

Because the function of server side is just to provide spatial data access, so Geo-Union server serves as an application, not any components. Geo-Union server can manage different spatial data resources, and also can reply different spatial data requests from different clients.

But Geo-Union client supports to develop domain-oriented applications. In order to provide flexible developing mode, around **map visualization objects**, we designed another six types of objects: spatial data access objects, map edit objects, spatial analysis objects, mouse tool objects, utility objects and AppTool. Figure 2 shows the architecture of service layer, the roles of different objects and the relationships among them [11][12].

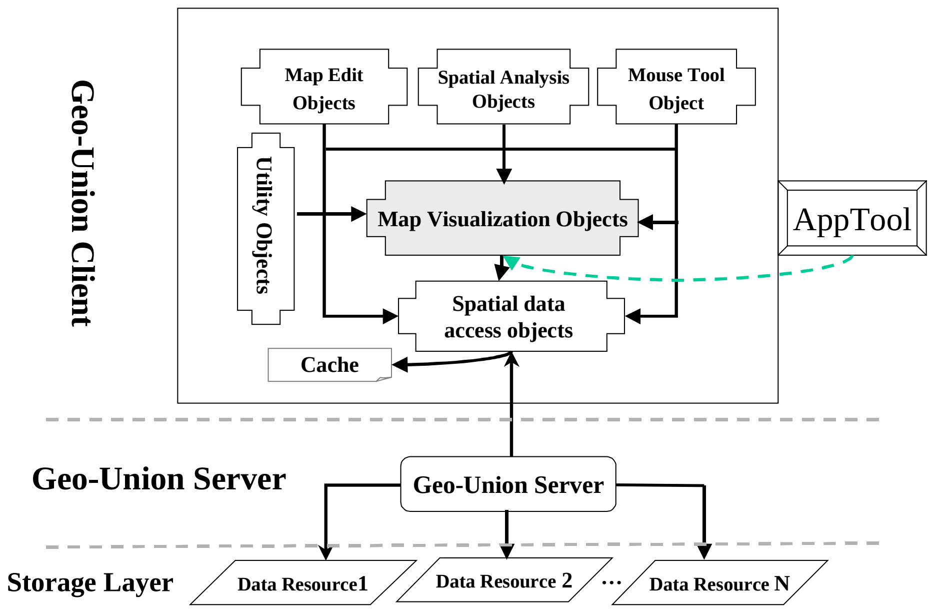


Figure 2 Architecture and Functional Partition of Service Layer

Spatial data access objects connect with Geo-Union server and access spatial data stored in spatial database. GxConnection is the core object of spatial data access objects. Through GxConnection, we can perform spatial data manipulating operations such as opening, adding, deleting, changing and querying corresponding data in spatial database. Also, a special object GxCache in spatial data access objects provides the spatial cache mechanism.

Map edit objects encapsulate mouse actions to edit spatial entity, such as inputting a point (GxInputPoint), inputting a line (GxInputLine) and inputting a polygon (GxInputPolygon).

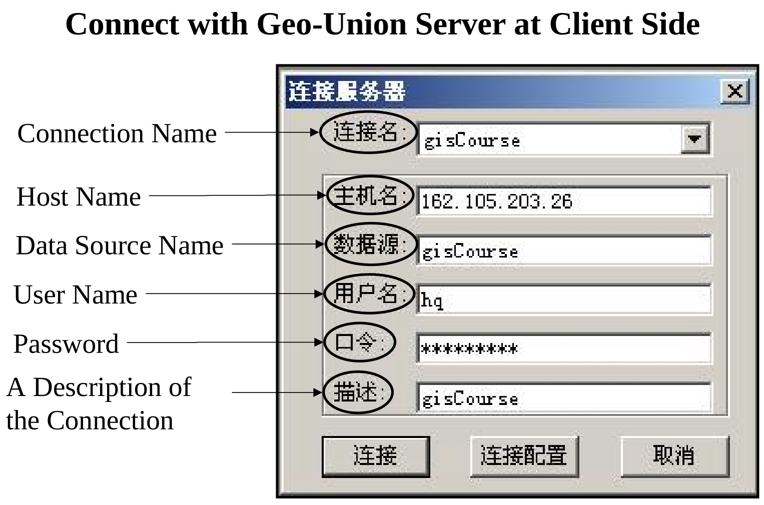
Spatial analysis objects provide spatial analysis functions such as overlay analysis (GxOverlay), buffer analysis (GxBuffer) and network analysis (GxNetwork).

Mouse tool objects are similar to map edit objects. They also encapsulate and manage mouse actions to form some basic GIS operations, such as zooming in map (GxZoomIn), zooming out map (GxZoomOut), roaming in map (GxPalm) and picking entity from map (GxPick).

Utility objects provide an object factory (GxFactory) and some data structure objects such as array (GxArry), set (GxSet) and enumeration (GxEnumeration). Utility objects are frequently used but may not be easily defined in some development environments (Visual Basic, VBScript and so on). For example, in VBScript, some programmable objects cannot be declared directly, but they can be generated easily by GxFactory. Also, Utility objects contain a GxError to provide a mechanism to deal with errors.

AppTool is an application-oriented component that integrates more GIS operations and user's interfaces using above objects. For example, *AppTool.Connect ()*, a method of

AppTool, provides a dialog for user or further developer at client side to indicate remote host, spatial data source, login user name and password when connecting with Geo-Union server (As shown in figure 3). This method integrates GxConnection in data access objects.



连接名:	gisCourse
主机名:	162.105.203.26
数据源:	gisCourse
用户名:	hq
口令:	*********
描述:	gisCourse

Figure 3 An Example of AppTool - AppTool.Connect( )

Map visualization objects are used to display map, interact with users and coordinate other objects to work. From figure 2, we can conclude that map visualization objects is the kernel of the system: spatial data access objects, map edit objects, spatial analysis objects, mouse tool objects and utility objects, all of them need map visualization objects to embody themselves. Any GIS applications need map visualization objects to integrate all other objects.

3. MAP VISUALIZATION OBJECTS IN Geo-Union

Map visualization objects are directly used to organize spatial data (layer and map) and display them. With map visualization objects, we can set the map to be displayed and the display attributes of all layers of the map, including display scale and scope of the map, and the visibility, selectivity and editability of each layer. Meanwhile, map visualization objects can be combined with other objects to provide various additional visualization functions, for example, map zooming, map roaming, information querying, entity picking, shortest path displaying, spatial analysis result displaying, entity label displaying, etc. Furthermore, there include some other functions such as dynamic layer displaying (GPS tracker) and map outputting. Map visualization objects contain a map visualization control (GxMapView) and three programmable objects: map object (GxMap), display layer object (GxMapLayer) and dynamic layer object (GxTrackingLayer). The class hierarchy of map visualization objects is illustrated in figure 4 [11].

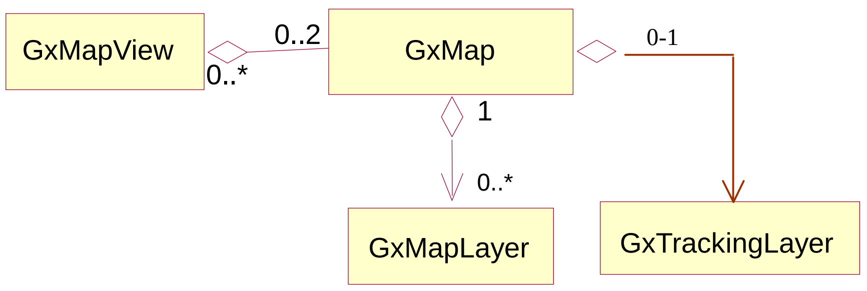


Figure 4 The Class Hierarchy Map Visualization Objects

Map (GxMap) is displayed in control window (GxMapView), and two maps can be displayed in GxMapView. Map (GxMap) consists of several layers (GxMapLayer). Especially, Map (GxMap) can contain a dynamic layer (GxTrackingLayer), for example, to adapt to a GPS tracker.

Generally, it is necessary to determine the function and define the interface (including properties and methods) for each component. In the following, we will give the design and implementation of map visualization objects in details.

3.1 Map Visualization Control - GxMapView

Map visualization is one of the core functions in WebGIS. GxMapView is a map visualization control, and map is displayed in control window. Besides controlling map visualization, GxMapView provides plentiful means to construct human-computer interactions. The interface of GxMapView can be defined as following:

(1) Property

ActiveMap and *BackgroundMap*: foreground map and background map to be displayed in control window. Background map will not be processed and is just used as a visual reference.

MapScale: current display scale of *ActiveMap* in control window.

Tools: a tool manager that manages mouse tool objects and map edit objects to control mouse actions in control window and implement some basic GIS functions in GxMapView (Also see section 2.2 about the functional partition of GIS components).

(2) Event

Further developers can redefine some necessary interacting actions through response to event. Five types of events are defined in GxMapView, they are mouse events

(*OnLButtonDown*, *OnLButtonUp*, etc), keyboard events (*OnKeyDown*, *OnKeyUp*), time event (*OnTimer*, which is generated by *SetTimer* method and killed by *KillTimer* method), command event (*OnCommand*, which is generated by *SendCommand* method or *PostCommand* method) and draw event (*OnDraw*, generated when the control window draws something. Through *OnDraw* event, users can obtain current size of control window, current display scale, extent and display layers of *ActiveMap*, so as to renovate relevant hint information for end-users in time).

(3) Method

Map management method: *GetProjection* – get information about current spatial reference system used to display *ActiveMap*; *GetExtent* - get extent (of *ActiveMap*) currently displayed in control window; *SetExtent* - set extent (of *ActiveMap*) to be displayed in control window; *GetFullExtent* - get full extent of *ActiveMap*; *SetFullExtent* - set full extent of *ActiveMap*; *SetCenter* - set a coordinate of *ActiveMap* to be displayed at the center of control window; *Zoom* – zoom the map by the center of control window; *Redraw* - redraw the map in control window.

Coordinate conversion method (*ProjectionToView*, *ViewToProjection*): coordinate or distance conversion between reference systems of control window and *ActiveMap*.

Event or command method: *SetTimer* and *KillTimer* - generate or kill a time event; *SendCommand* - send a command event and return after the command is finished; *PostCommand* - send a command event and return immediately.

3.2 Map Object - GxMap

Map object (GxMap) can be displayed in GxMapView. GxMap is a container of a series of display layer objects (GxMapLayer), as well as controls their visualization properties. Beside adding and removing a GxMapLayer object, GxMap also provides the functions of controlling the display scope and displaying sequence of all GxMapLayer objects in it. The interface of GxMap can be defined as following:

(1) Property

The properties of GxMap mainly are map name (*MapName*), map description (*MapDescription*) and number of GxMapLayer objects in it (*LayerCount*).

(2) Method, mainly includes map management method and layer management method.

Map management method: *SetDefaultExtent* - set default display extent of the map; *GetDefaultExtent* - get default display extent of the map; *SetFullExtent* - set full display extent of the map; *GetFullExtent*- get full display extent of the map; *Copy* - copy the map from another map; *Clone* – clone a copy object the map.

Layer management method: *GetMapLayer* - get indicated GxMapLayer object from the map; *AddMapLayer* - add a GxMapLayer object into the map; *InsertMapLayer* - insert a GxMapLayer object into the map; *RemoveMapLayer* - remove a GxMapLayer object from the map; *RemoveAllLayers* – remove all GxMapLayer objects from the map; *MoveLayer* – adjust the display sequence of GxMapLayer objects in the map.

3.3 Display Layer Object - GxMapLayer

Display layer object (GxMapLayer) is an abstract concept, which corresponds with a physical layer or a series of layer views. In different display scale, layer views can be used to replace physical layers to enhance layer display speed. GxMapLayer provides a visual layer for users, which contains display properties, legend configuration and label information of a physical layer. The interface of GxMapLayer can be defined as following:

(1) Property

Most properties of GxMapLayer are display-relevant configurations of a physical layer, including: *Layer* - the physical layer it contains; *ViewLayers* – the layer views list it contains; *MaxVisibleScale* - the maximum visible scale of the layer or layer views; *MinVisibleScale* - the minimum visible scale of the layer or layer views; *Visble* - whether the layer is visible; *Pickable* - whether entity in the layer is pickable; *ActiveAnnoed* - whether the layer is displayed with automatic annotation; *Legendable* - whether the layer is displayed using legend.

(2) Method

Method in GxMapLayer is mainly used to set or get a legend configuration that describes how a physical layer is displayed using legend (*SetDisplaySetting* and *GetDisplaySetting*).

3.4 Dynamic Layer Object - GxTrackingLayer

Dynamic layer object (GxTrackingLayer) is used to describe entities that be added into a map dynamically during real operation. For example, it is very common to insert a GPS tracking point, a virtual line or a virtual region into active map in GxMapView. The interface of GxTrackingLayer is same as GxMapLayer, but the data of GxMapLayer is from a physical layer in spatial database, and the data of GxTrackingLayer is uncertain and is added according to users’ or other operations. The data of GxTrackingLayer can be stored into spatial database permanently. Once a GxTrackingLayer is permanently stored, it becomes a GxMapLayer.

4. MAP VISUALIZATION SAMPLE IN Geo-Union

The following is a map visualization and manipulation sample in Microsoft Visual Basic using Geo-Union components and its map visualization objects. Figure 5 is the screen shot of the sample.

```
    //Using GxConnection to connect with Geo-Union server
  Dim Conn As New GxConnection
  Conn.Connect (“162.105.203.26", "gisCourse", "hq", "gis")
    //Get physical layers from spatial Database through GxConnection in client side
  Dim Layer as GxLayer                          //Declare a physical layer object
  Set Layer = Conn.OpenLayer (“Road")           //Get a physical layer whose name is “Road”
  Dim mapLayer As GxMapLayer               //Declare a display layer object
  mapLayer.layer = Layer                   //Associate the display layer object with the physical layer object
  //Set display properties of the display layer object
  mapLayer.MaxVisibleScale = 1
  mapLayer.MinVisibleScale = 1000000
  //Other layers can be acquired using the same way

    //Add a display layer object to a map object
  Dim Map as GxMap                  //Declare a map object
  Map.AddMapLayer mapLayer          //Add the display layer object to the map object
  // Other display layer objects can be added into map object using the same way

//Set the active map and it display properties in map visualization control.
//MapView1 is an instance of GxMapView.
  MapView1.ActiveMap = Map
  MapView1.SetFullExtent Map.GetExtent
  MapView1.SetExtent Map.GetExtent

    //Manage mouse tool objects
  Dim myPick As New GxPick                      //Declare a entity pick object
  Dim myZoomin As New GxZoomIn                  //Declare a map zoom in object
  Dim myZoomout As New GxZoomOut            //Declare a map zoom out object

  Dim myPalm As New GxPalm                  //Declare a map roam object

  //Using tool manager of GxMapView to implement map zoom in function

  If Not MapView1.Tools.IsToolIn (GxZoomIn) Then

       MapView1.Tools.AddTool myZoomin

  End If

  MapView1.Tools.ActiveTool = myZoomin

  //Other interacting functions can be implemented using the same way


    //Manage map edit objects

  Dim myInputLine As New GxInputLine              //Declare a line input object

  Dim myInputPolygon As New GxInputPolygon        //Declare a polygon input object

  // Using tool manager of GxMapView to implement polygon input function

  If Not MapView1.Tools.IsToolIn (GxInputPolygon) Then

       MapView1.Tools.AddTool myInputPolygon

  End If

  MapView1.Tools.ActiveTool = myInputPolygon

  //Other input functions can be implemented using the same way


    //Pick an entity and get its attribute through mouse events of GxMapView and other objects.

  MapView1_OnLButtonUp (ByVal Flags As Long, ByVal x As Long, ByVal y As Long)

       If MapView1.Tools.ActiveTool Is GxPick Then

            Dim Sels As GxArray

            Set Sels = myPick.GetSelections

            Dim Sel As GxSet

            Set Sel = Sels.Element (0)

            Dim Ent1 As GxEntity

            Set Ent1 = Sel.GetEntity(0)             //Get the picked entity

            Dim attr as String
```

```
            Set attr = Ent1.GetField ("name")            //Get value of attribute “name” of the picked entity

        End If

    End Sub



        //Using AppTool to adjust display layer objects and their display properties in map object.

        //AppTool1 is an instance of AppTool.

        AppTool1.LayerControl Conn, Map

        MapView1.Redraw
```

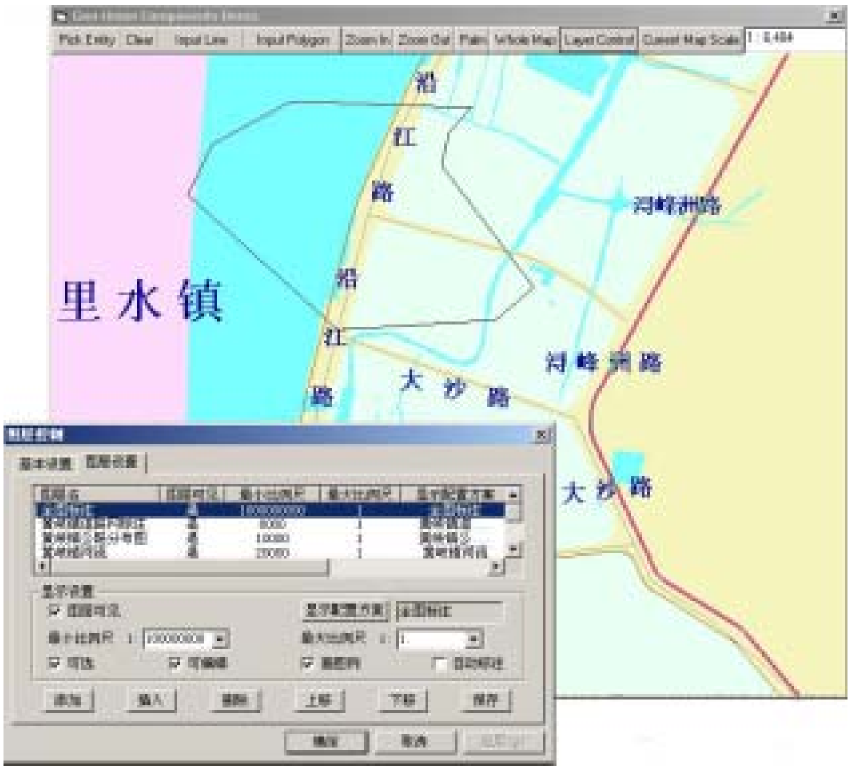


Figure 5 Map Visualization Sample in Geo-Union

5. CONCLUSION

WebGIS is the organic combination of GIS and Internet. It is the extension of Internet in the domain of GIS, which expands the research and application fields of GIS greatly. The development trend of WebGIS is the management of massive distributed spatial data and the construction of distributed component-based WebGIS. Component-based WebGIS can be integrated into other development environments seamlessly, therefore can not only decrease the complexity of applications and speed up development progress, but also descend the cost and increase the maintainability. The architecture of component-based WebGIS is clear, and every component there has definite and centralized function. Component-based system makes it more convenient to realize domain-oriented application systems in a distributed environment.

Geo-Union has finished a preliminary component-based model for distributed WebGIS, and has got into use in many fields with sound effects. In application development, the design and classification of map visualization objects are consistent with people's viewpoint about real map, and bring a great convenience for further development.

Right now, map visualization objects in Geo-Union mainly focus on two-dimensional map visualization. In the future work, we will extend the capability of map visualization objects to three or four-dimensional visualization.

ACKNOWLEDGEMENTS

Supported by the National Natural Science Foundation of China under Grant Nos. 60030005, 60073016, 60203002; the National High Technology Development 863 Program of China under Grant Nos. 2001AA113151, 2002AA135330, 2002AA134030; the National Grand Fundamental Research 973 Program of China under Grant No. 2002CB312000; the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No. 20020001015; the Natural Science Foundation of Beijing of China under Grant No. 4012007.

REFERENCES

[1] Kong Yunfeng *et al* (1998). "Some Issues in System Integration of Web-based Geographic Information Systems" (in Chinese), *Journal of Remote Sensing*, 2(2): 143-148, 1998.

[2] Zhang Li *et al*,(1998). "Geographic Information System in the Internet Age" (in Chinese), *ACTA GEODAETICA et CARTOGRAPHICA SINICA*, 27(1): 9-15, 1998.

[3] Szyperski C (1998). "Component software: beyond object-oriented programming Reading", *MA: Addison-Wesley Press*, 1998.

[4] http://www.esri.com.

[5] http://www.mapinfo.com.

[6] http://www.autodesk.com.

[7] http://www.intergraph.com.

[8] http://www.geostar.com.cn.

[9] http://www.opengis.org.

[10] Dept. of Computer Science and Technology, Peking University (2000). "Operation Guide for Geo-Union Enterprise" (in Chinese), *Technology Material*, http://gis.pku.edu.cn.

[11] Dept. of Computer Science and Technology, Peking University (2000). "Component Guide for Geo-Union Enterprise" (in Chinese), *Technology Material*, http://gis.pku.edu.cn.

[12] Li Muhua (2000). "Study on Component based WebGIS and Its implementation" [Master Dissertation] (in Chinese), Beijing: Peking University, 2000.6.