Client-Server Interaction in GIS Applications Through Web Services

Xuan Shi
Student Assistant of 2004 ESRI International User Conference
West Virginia University

Abstract

Web services have been developed and deployed for GIS applications. Client-server interaction in such applications is limited by the service provider, such as ArcWeb Service. This paper will discuss the needs from service requester which lead to the exploration on the design and manipulation of Web Service interface. It is likely that GIS Web Service promotes the separation of spatial feature and non-spatial feature as well as the data processing modules, considering that Web Service is developer oriented rather than end-user oriented. It will demonstrate that through spatial feature interface development, user-specified real-time dataset can be transferred and integrated in GIS applications through Web Service deployment. Scalable Vector Graphic (SVG) can be an alternative choice in GIS Web Service development and deployment while ArcGIS Server 9.x can offer strong server functions for Web Service development in the near future. Both advantages and disadvantages of these two technologies will be investigated and demonstrated.

Introduction

Web service is a ***developer-oriented*** software component that can be accessed and integrated by application developers through standard application program interfaces (APIs). It can be used by Internet, wireless, and Windows stand-alone GIS applications, and further, provides the foundation for componentizing distributed GIS data and functional modules. However the power of web services is ***not*** yet fully implemented in the GIS domain. The existing GIS Web Services (ESRI 2004, TerraServer 2004) look likely to be a kind of ***end-user oriented*** Internet Mapping Server (IMS) as they are just one-step further to the end users-the application developer simply pass the mapping product to the end user.

The existing GIS Web Services can be summarized into four categories: 1. Mapping services: the user inputs the location (longitude and latitude) and then receives the map; 2. Geocoding and reverse geocoding services; 3. Routing services; 4. Coordinate conversion services - for example, convert coordinates from a longitude/latitude pair to UTM and vice versa. Currently there are no GIS data services and the mapping services cannot integrate developer-specific information. The interaction between the service provider and the service requester is limited.

This paper will first discuss the needs assessment of the development and utilization of GIS Web Services. Next a fundamental classification of GIS Web Services will be explored to set up the present research goal and for the identification of future effort in development. GIS data standards will be of great importance in the development and utilization of GIS Web Services. A generic XML-based GIS data format will be used in this research instead of GML/ArcXML to perform dynamic data and system integration by separating the spatial and non-spatial datasets. This research demonstrates first that the same GIS data service can be integrated with other different data services and/or different geoprocessing service modules to generate different final products (raster maps or vector maps) that can be used in different applications (Internet vs. windows stand-alone applications) through Web Services Description Language (WSDL) interfaces. Secondly it shows that application developers can perform dynamic integration through the Web Services' standard APIs using different commercial or open source products to accomplish their thematic GIS applications by utilizing the same GIS data and functional services. Lastly future research direction and challenges will be discussed.

Needs Assessment

In contrast with the traditional role of end-user as the client, client-server relationship in the Web Service era should be re-defined as the relation between the service provider (server) and service requester (client). Service requester/user is the application developer who will utilize and integrate the Web Services to build thematic applications for the end users. The problem of client-server interaction was raised in utilizing ArcWeb Services for a thematic Web application showed in Figure 1: how can the application developers add their own GIS data into the base map provided by ArcWeb Services?

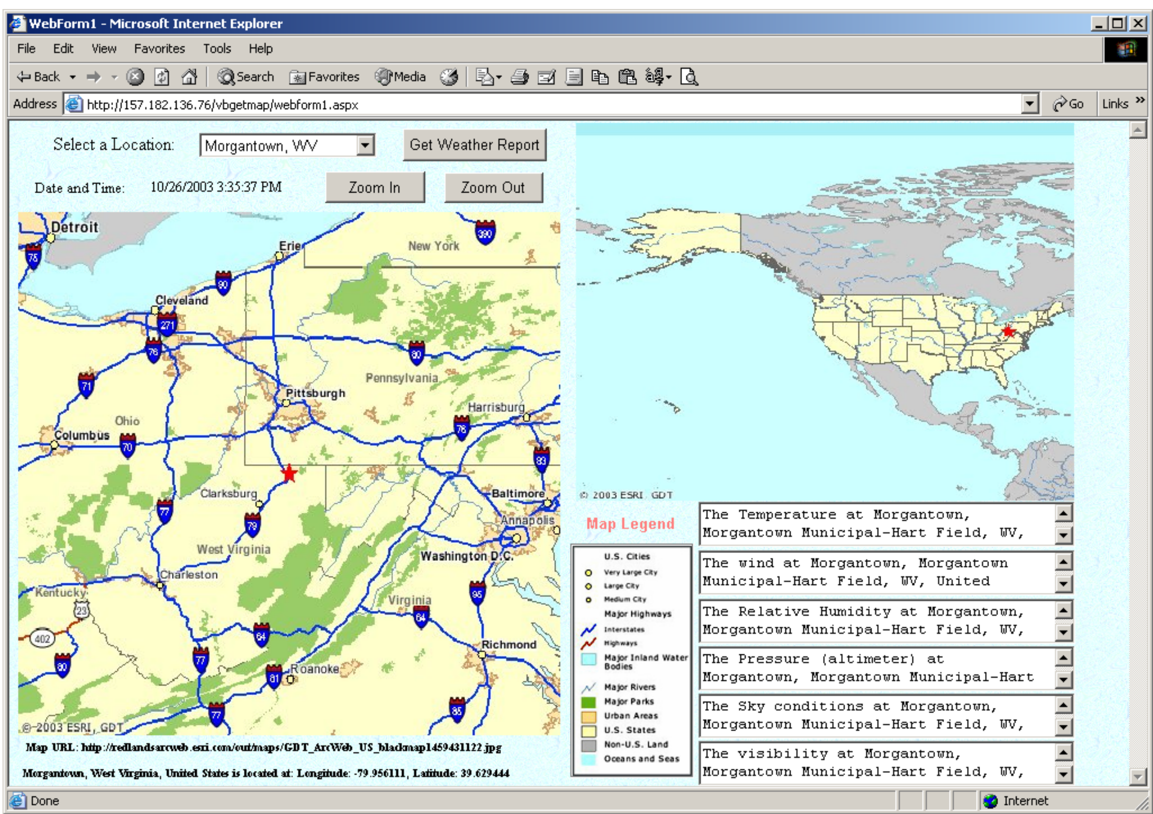

Figure 1. Utilizing ArcWeb Services to develop thematic Web applications

Traditionally, implementing a GIS application starts from needs assessment that should also be an initiation in GIS Web Service development. A needs assessment is a systematic analysis of the required GIS applications, functions, and data that can be a fundamental blueprint to build Web Services. The needs assessment enables the service provider to understand what kinds of GIS Web Services are requested by the application developers so as to build more targeted services efficiently. Since it is hard to make such assessment in the huge group of application developers with widely varying background in different domains, a study of the classification of GIS Web Services will provide an initial direction to the design and implementation issues.

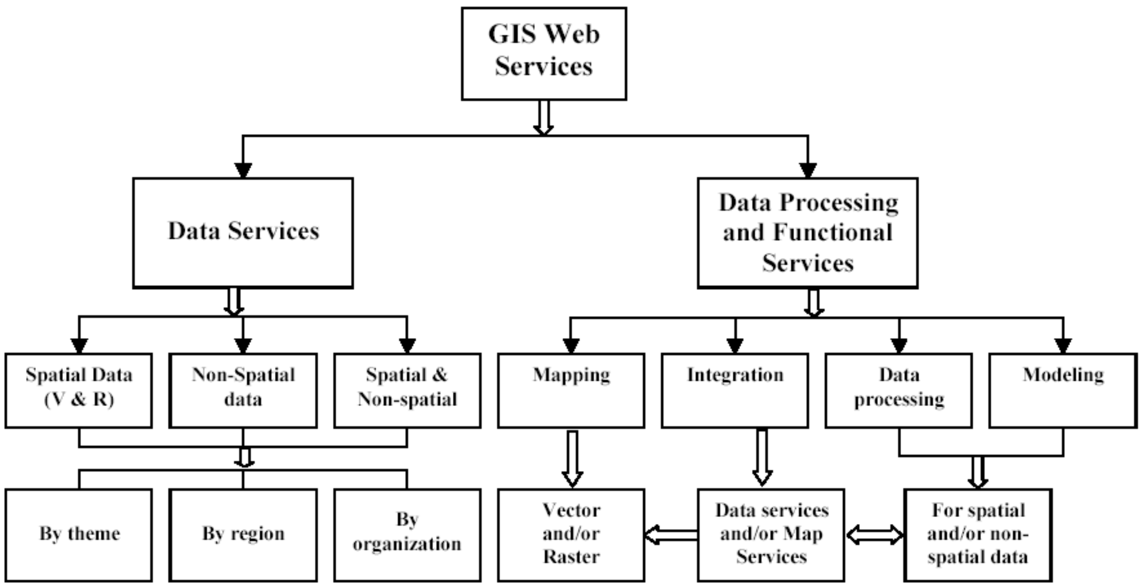

Figure 2. An Initial Classification on GIS Web Services

In Figure 2, GIS Web Services are classified into two categories: GIS data services and geodata processing and functional services. Since GIS data are composed of spatial and non-spatial features, data service providers can offer either one or both kinds of the data to service requesters. Such services can further be classified or organized by vector or raster data types, or by theme, region, or organization. In this way, application developers, i.e. service requesters, can search for their specific dataset to build their applications. Such system architecture will give more chances for data developers to enter the market and will enhance the data accuracy and concurrency.

Web Services for geoprocessing will be much more complex than the available service styles. Mapping service is currently the most commonly available GIS Web Service. Service requesters can obtain a map from the service provider by sending a request to define the map extent. The spatial data are predefined and preset, such as that in ArcWeb Service. However, ***providing mapping services can be more complicated if the GIS data on the requester side can be integrated with the data provided by the service provider – which will be demonstrated in this research***.

An integration service will need the information of the input datasets, the order of the overlaid layers, using or not using a predefined dataset as base map, etc., while the output will result in different formats, either a new data service or a map product in either raster or vector format. OGC's standardized APIs such as "GetMap" cannot match such complicated cases as all those APIs cannot accept user-specified data and can only provide relevant information/map of the predefined/preset data. New standardized interfaces need to be formulated and implemented, such as addPolygon(s), addPolyline(s), addPoint(s), mergePolygon(s), mergePolyline(s), mergePoint(s), etc.

Web Service for GIS data processing and modeling will be a promising domain for development in the near future. Spatial and non-spatial data can be processed separately by third-party developers even without any relation with GIS. As the input and output to and from web services can be the datasets themselves, such services can be easily chained with other service types, such as mapping and integration. The future of GIS Web Services "holds great promise for dedicated spatial service developers and integrators whose ***imagination*** will be the primary limit to the applications that can be built" (Ball, 2002).

Data Format and Standard

GIS data are distinct by their dual structure as a combination of both spatial and non-spatial features. Usually GIS data would be unlikely to "use the same data model or semantic representation of geographic information" (NCGIA 1997, Sondheim, et al., 1999). While XML is the foundation of Web Service technology, the challenge towards the GIS industry and professionals lies in how to convert such vendor-specific GIS data to and from the XML-based, standardized dataset for transmission, geoprocessing, and dynamic integration over the Internet through Web Service technology.

Geography Markup Language (GML) is a non-proprietary XML-based encoding of geospatial data and was originally designed and implemented for the OGC's Web Mapping Testing Bed. However, GML may likely ***not*** be the best GIS data standards for Web Services in GIS application, although GML is XML based that is compatible with Web Services standards and protocols.

Web service is a software component thus GIS data service should be componentized. The goal of Web Service development is not to build a whole software package but rather just a component for dynamic integration by the application developers. GML's ambitious goal to include everything such as spatial feature geometry and properties, topology, temporary data, spatial reference system, measurement units, meta-data, styling, etc. make it difficult for implementation.

Besides the problem of complexity, the size of GML file is too large to be loaded and transmitted effectively. Not all of the GIS data specified in GML are necessary and useful for transmission and integration. For each thematic layer in a map, normally only one non-spatial attribute field can be used for presentation, such as population density. Even in case of generating a thematic map using the method "Unique values, many fields" in

ArcMap, only 3 fields can be used. Given the example of ESRI's US counties shapefile, there are more than 50 fields in the file. If 1-5 fields can be used to generate a thematic map, still only 1/50~5/50 or 2%~10% can be used to generate a thematic map per layer. The size of US counties shapefile is about 1,500 KB while the size of the corresponding GML file is over 16,600 KB.

ArcXML is an XML-based map configuration file for developing and customizing ESRI's ArcIMS project. However, besides its proprietary nature that is not vendor-neutral, ArcXML is not ***coordinate-centered,*** which is the most important part for the spatial data to be utilized in GIS Web Services.

As mentioned in the introduction, existing GIS Web Services can be summarized into four categories (mapping services, geocoding and reverse geocoding services, routing services and coordinate conversion services). All of such services depend on, directly or indirectly, the coordinates of the geographic location. Thus the generic GIS data for Web Services need to be ***coordinate-centered***. In this way, one standardized XML-based GIS data service can be integrated in different kinds of applications through Web Services.

This research proposed an XML-based vector data format to be used in developing GIS data and integration services. The data format can be described as the follows (Figure 3):

Point Feature Format	Line Feature Format	Polygon Feature Format
<?xml version="1.0" encoding="utf-8" standalone="no"?> <GeoFeature> <minX>......</minX> <minY>......</minY> <maxX>......</maxX> <maxY>......</maxY> <**PointFeature**> <FID>......</FID> <NAME>......</NAME> <***coordinates***>......</***coordinates***> </**PointFeature**> </GeoFeature>	<?xml version="1.0" encoding="utf-8" standalone="no"?> <GeoFeature> <minX>......</minX> <minY>......</minY> <maxX>......</maxX> <maxY>......</maxY> <**LineFeature**> <FID>......</FID> <NAME>......</NAME> <***coordinates***>......</***coordinates***> </**LineFeature**> </GeoFeature>	<?xml version="1.0" encoding="utf-8" standalone="no"?> <GeoFeature> <minX>......</minX> <minY>......</minY> <maxX>......</maxX> <maxY>......</maxY> <**PolygonFeature**> <FID>......</FID> <NAME>......</NAME> <***coordinates***>......</***coordinates***> </**PolygonFeature**> </GeoFeature>

Figure 3. Generic vector data format

Metadata information will be especially important to service providers that offer data integration services, however, it's beyond the immediate scope of this discussion. Separation of spatial and non-spatial data is the necessary step for creating such generic data standard for GIS Web Services. The spatial geometric feature will be the fundamental component in the distributed GIS systems, while non-spatial attributes can be integrated and queried dynamically and separately through the unique identifier of the spatial feature only when a certain attribute is needed in deployment (it has been demonstrated in another project). Application developers will have more chances to select and integrate necessary data from different sources with different scales, projections, etc. Results from this particular research project demonstrate that transferring non-spatial attributes in XML-based GIS data is not necessary in the mapping integration services.

GIS Web Service Development

Since ArcGIS Server 9.0 was not available when development began on this project, Scalable Vector Graphic (SVG) was chosen for the Internet mapping applications and to provide the base map. To perform dynamic GIS data and service integration over Internet through Web Services, this project developed GIS Web Services in three categories as shown in Figure 4:

- ***GIS mapping service***: provides a base map by calling getSVGMap() function.
- ***GIS data services***: provides GIS dataset in the above-mentioned three generic formats and the combination of all the features by calling the functions like getPointFeature(), getLineFeature(), getPolygonFeature() and getMultiFeature().
- ***GIS integration services***: by calling the functions like addPoint2SVGMap(), addLine2SVGMap(), addPoly2SVGMap(), addMultiFeat2SVGMap(), or addpntbyURL2SVGMap(), the service provider accepts and integrates the generic dataset sent from the requester into the base map and send back the final product to the service requester.

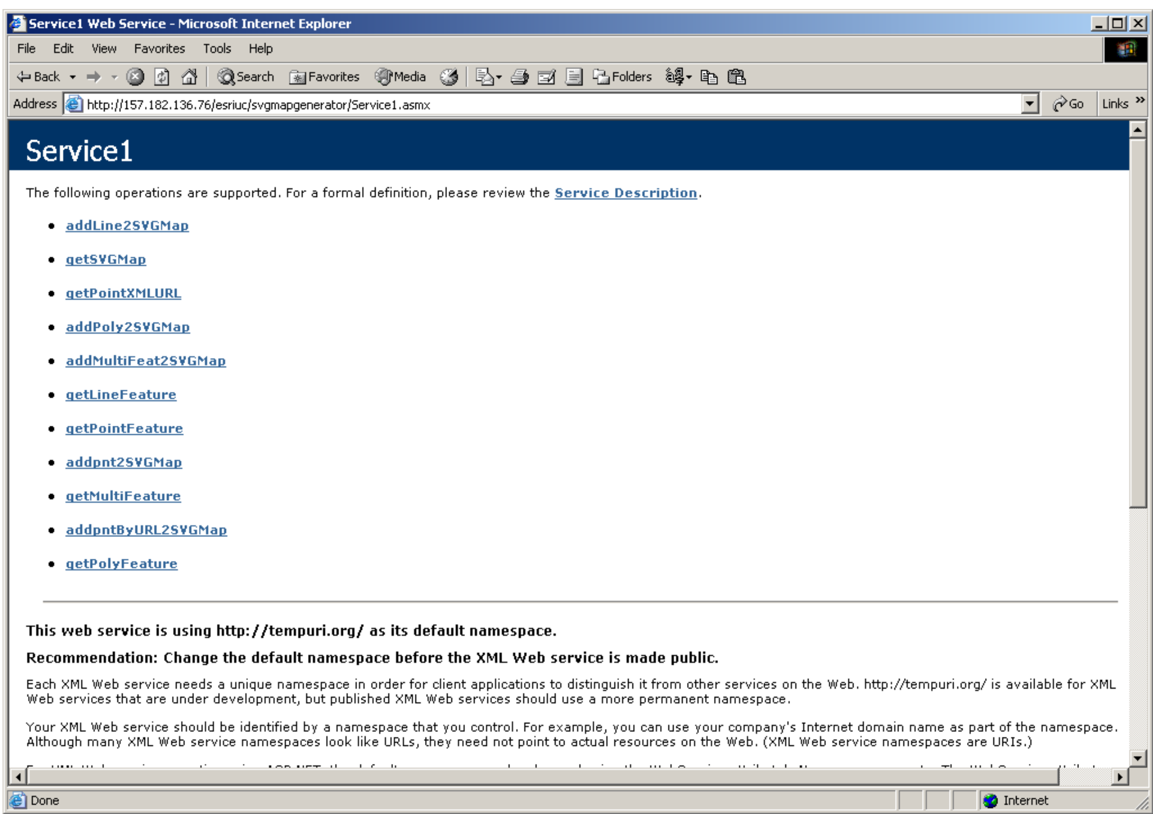

Figure 4. Developed functionalities for GIS Web Services

Microsoft Visual Studio .NET is the platform for developing such GIS Web Services. Initially the mapping service provides a base map of Monongalia County, West Virginia. The point features were extracted from the shapefile of populated places in this county, the line features included the US Routes inside the county, and the polygon feature is the municipal boundary of Morgantown. The coordinate system of the base map and the input data is set up as UTM Zone 17 (NAD 83). At first, all shapefiles of the above-mentioned spatial features were converted into GML file by the ArcGIS OGC Interoperability Add-on. GML files were then converted into the generic data format designed for this project by using the XmlTextWriter and XmlTextReader in .NET development toolkits. Moreover XpathDocument and XpathNavigator were used in generating the GIS Web Services developed for this research.

GIS Web Service Utilization

To perform dynamic integration of distributed GIS through Web Services, this research sets up the goal for the utilization of GIS Web Services as that the same GIS data service can be integrated with other data services and/or different geoprocessing service modules to generate different final products that can be used in different applications (Internet, wireless, or windows stand-alone applications) through WSDL interfaces. Application developers can perform dynamic integration through the Web Services' standard APIs using different commercial or open source products to accomplish their thematic GIS applications by utilizing the same GIS data and functional services.

Two kinds of implementations were developed in this study to utilize the GIS Web Services:

- Figure 5 shows the Internet application using SVG-the open source product-as the map viewer.
- Figure 6 shows the stand-alone Windows form application using ESRI's ArcObjects within Visual Studio .NET environment.

In both applications, the solid blue polygon refers to Monongalia County. The hollow polygon in yellow represents the Morgantown municipal boundary. The red lines are the US Routes (19 and 119) and the red dots are the selected populated places in the county. The user first selects one feature (point, line, polygon) on the form interface. A request will send to the service provider to retrieve the formatted generic data that will be showed in the textbox. The user then clicks button "Add to Map" and the selected feature will be added into the base map. The users can change the data sets by altering the value of the coordinates (i.e. the x, y value of UTM points) to draw their own customized maps dynamically.

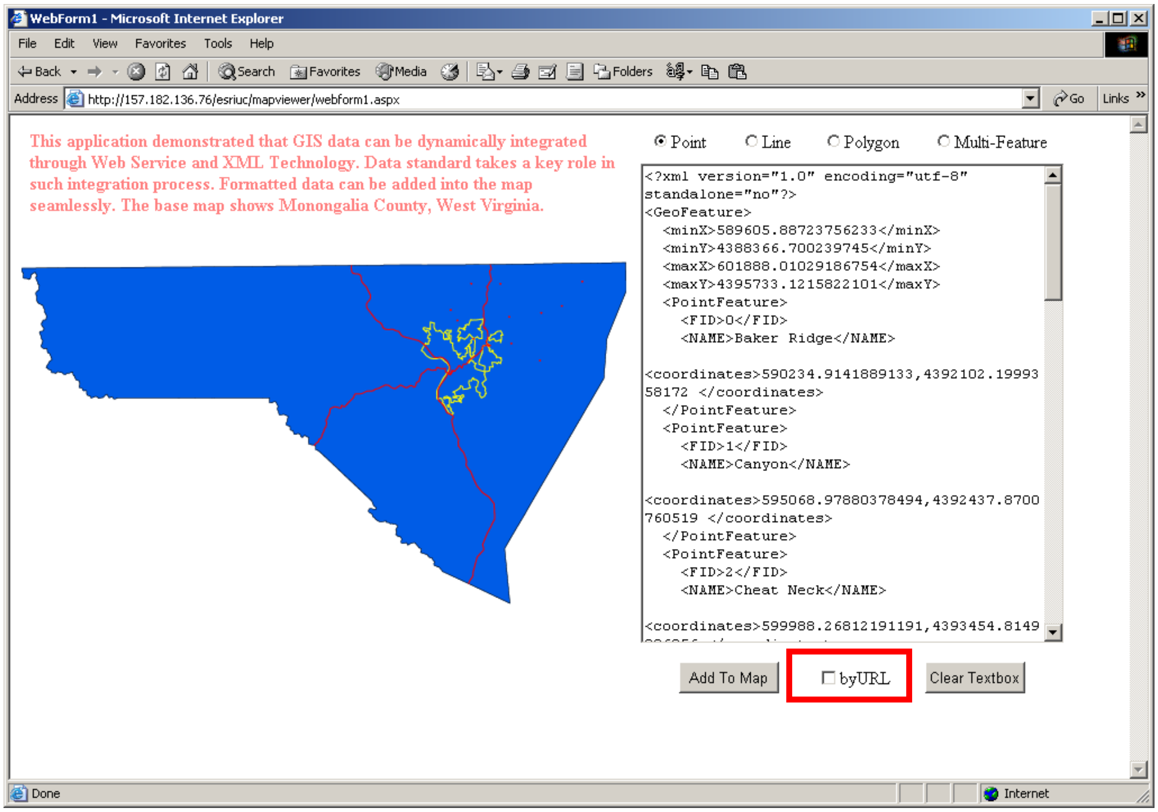

Figure 5. Internet application with SVG viewer for GIS Web Service Utilization

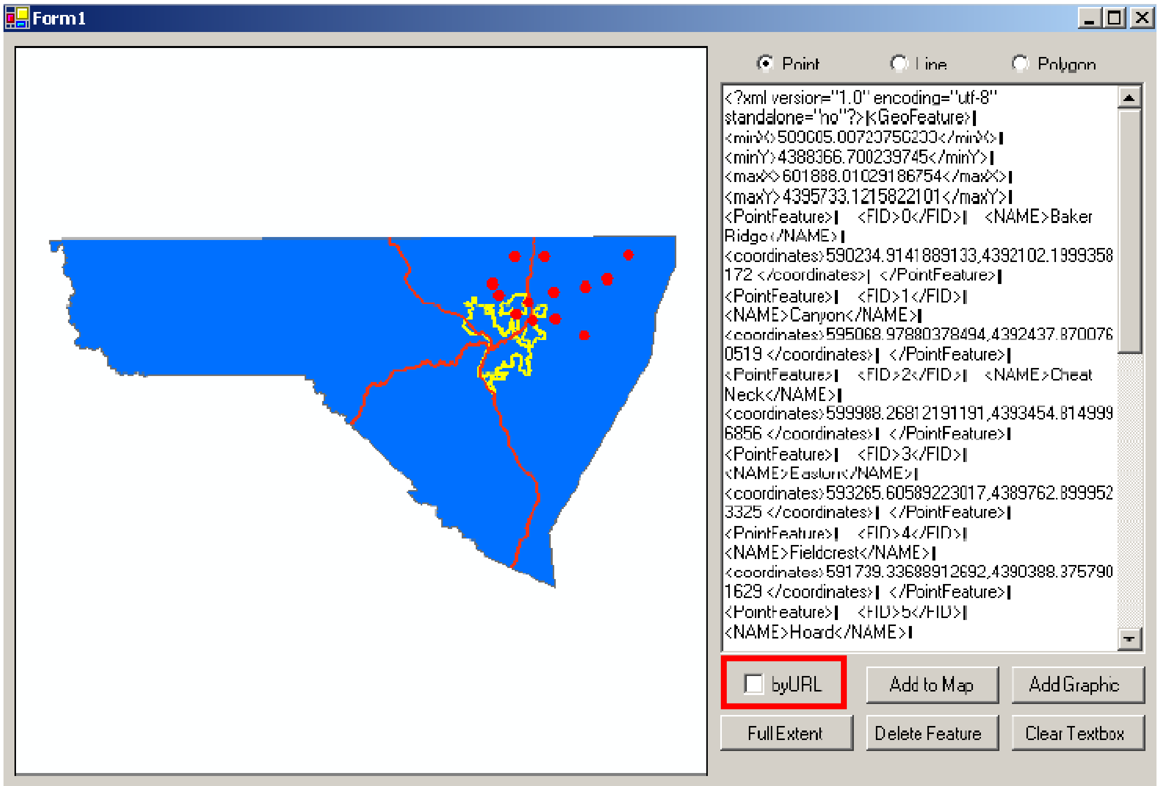

Figure 6. Stand-alone application with ArcObjects Map Viewer for GIS Web Service Utilization

For the Internet application, a request will send to the service provider which will perform the corresponding function to add the feature into the base map. The interface at the end-user side will be refreshed to show the newly generated map on the screen. As for the stand-alone application, firstly, a shapefile was added into the AxMapControl as the base map. Three empty shape files (point, polyline and polygon) were also added into the AxMapControl. Once the XML-based generic data were retrieved from the service provider, the corresponding features were added into the due empty shapefiles. In this process, XpathDocument and XpathNavigator were used to retrieve the coordinates of each point in the input dataset. IWorkspace/IWorkspaceEdit were used to control the edit session. CreateFeature of IFeatureClass was used to create the point, line or polygon features into the shapefile. Different control objects and functions were used to draw different features, such as PutCoords function for IPoint object, the AddPoint function of IPointCollection object for getting IPolyline, the AddGeometry(IRing) function of IGeometryCollection for getting IPolygon.

An AddGraphic function was created in this project to add the point features as graphics on to the map. In this way, no empty shapefiles are needed. Such function has the potential in developing the Server-side applications in the future with the power of ArcGIS Server as the service provider will just generate the new map with the input dataset transferred by the service requester without changing the database.

Developing GIS applications through Web Services will fundamentally change the traditional working flow as showed in Figure 7. It is demonstrated in this research that XML-based standardized GIS data can be integrated and utilized ***directly*** into GIS applications without being ***downloaded*** into the local machine. The XML-based GIS data can be integrated with ESRI's shapefile seamlessly without any ***conversion***.

Future Challenges

Many new problems and questions are raised during this research. Given an example, the interface standards will be more complex in the future, as illustrated in figure 2. Figure 5 and 6 shows that the input of a service request can be either a full XML document or just a URL pointing to that document. However, in programming, both inputs are just simple strings. Similarly, the outcome can be a map product referred by the image URL, an XML-based GIS dataset, a vector base graphic like SVG document or the URL pointing to this document, or an error message. All these outputs are also simple strings in SOAP message exchange. How to differentiate the semantics in data and system exchange will be a big challenge to the GIS and IT community.

Developing algorithms to process XML-based spatial data for spatial analysis will likely be the critical point in GIS Web Service development in the future. Even though the ArcGIS Server is expected to be the preferred GIS Web Service provider, it seems that the conversion between XML-based GIS dataset and the binary-based proprietary data has to be performed two times at data input and output process. The componentization of GIS functions for spatial analyses will be the next target in the research and commercial GIS Web Service development. Eventually, the GIS software will probably be a spatial data interpreter for mapping as all other functions are componentized into modules provided through Web Services which will lead to the changing role of GIS software in the future.

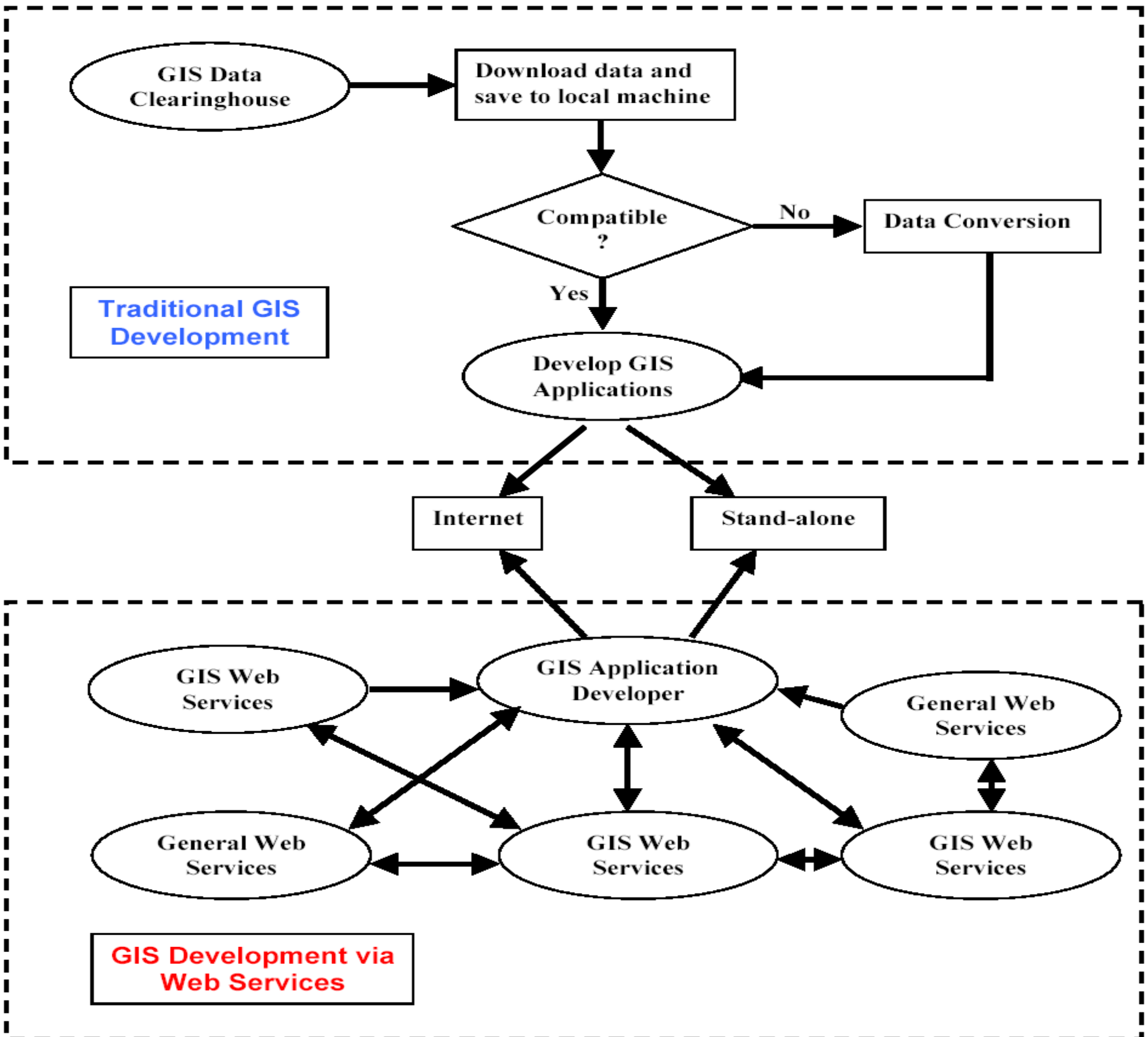

Figure 7. The changing style in developing GIS applications

Reference:

1. Ball, M. 2002. Web Services Have Arrived! http://www.geoplace.com/gw/2002/0208/0208edit.asp (last accessed March 22, 2004)
2. ESRI's ArcWeb Services. 2004. http://arcweb.esri.com/arcwebonline/index.htm
3. Microsoft TerrsServer. 2004. http://terraserver.microsoft.com/webservices.aspx
4. NCGIA. 1997. Interoperability of Geographical Information http://www.ncgia.ucsb.edu/other/ucgis/research_priorities/paper5.html
5. Sondheim, M., Gardels, K. and K. Buehler. 1999. GIS interoperability. In Geographical Information Systems 2nd Edition. Edited by Longley, P. A., Goodchild, M. F., Maguire, D. J. and D. W. Rhind. John Wiley & Sons, Inc.