Publishing Geographic Data for Local Government

Lucian Zavate

Bachelor Degree in Geography, Master Degree in GIS
Lucian Zavate
ESRI Romania
Strada Roma nr.8 sector 1, Bucuresti, Romania
Mobile: 40 742 900405, Fax: 40 21 2339088, E-mail: lzavate@esriro.ro

Abstract

The objective of this paper is to present a tool for GIS-based management systems applied in urban areas. The entire application is based upon ArcIMS's capability to build and deliver maps and data over the Internet. The application takes advantage of the Internet technology that makes it possible to bring GIS data right on your desktop allowing you to display, query and manage those data, build and publish maps without the need of a GIS software installed on your computer.

The study area is a district of Bucharest. The applications is an answer to the demand of a better management system of urban areas, a system that has to be accessible at any time, has to be portable and reliable – providing information at your fingertip.

We are in a continuous interaction with the environment we live in, our actions have an impact over the environment, and the changes in the environment reflect on us. For some of us this environment is represented by the city we live in. But the city itself is alive, it is changing every day, it evolves and all this changes impact on us. The decisions taken by the local government, in our case the City Hall, affect us all, and these decisions have to have a support – geographic data. In order to take the right decisions we need current and accurate data, for example a current and accurate street network is vital for emergency dispatching, transportation routing and other GIS-based operations that local government use when providing services.

Over the years the Internet proved to be a very an extremely popular and efficient mean of sharing information, data and technology. Delivering and publishing geographic data over the Internet emerged as a natural thing. Providing to local government a tool to better organize, use and display geographic data is very important. This paper presents such a tool. Our application is based on ArcIMS's capability to build and deliver maps and data over the Internet. In the following lines the tool's architecture will be presented briefly

THE ARCHITECTURE

This tool is a collection of applications which work together, ArcIMS being the backbone of the tool. For the ArcIMS to work a Web Server and a Servlet Engine need to be installed an properly configured to work together. The Web server receives requests from clients and sends Web pages for distribution. For this application we used Apache 2.0.54. Servlet engines, in our case Tomcat 2.0.48, extend Web servers using a common application programming interface (API) and allow them to process Java code. A servlet is a Java program that runs as part of a Web server and responds to requests made to a special URL. The most common use for a servlet is to extend a Web server by dynamically generating Web content. A servlet can be written so it receives the request, acquires and processes the data as needed by the client and then returns the result to the client. Servlets are analogous to applets, except they are operational on the server side of the architecture.

ArcIMS runs in a distributed environment and consists of both server and client components. It incorporates a processing model that uses many components working in cooperation to run simultaneously and create map data for delivery via the Web. The server side components are: the ArcIMS Spatial Server, ArcIMS Application Server, ArcIMS Application Server Connectors, and ArcIMS Manager. The client side components are: the ArcIMS HTML Viewer and ArcIMS Java Viewer.

ArcIMS Spatial Server

The ArcIMS Spatial Server is the core of ArcIMS, it host the ArcIMS Services, it processes the requests arrived for obtaining maps and/or information about those maps, extracts data, geocodes, but the primary job is to dynamically generate the map images that will be displayed in the map viewer.

There are a number of ArcIMS Service that will present spatial data and metadata to users via a Web Server. ArcIMS has several type of services, each service corresponding to a Virtual Server type and Spatial Server function type:

- Image Service;
- Feature Service;
- ArcMap Image Service – with this service one can serve ArcMap and ArcGIS Publisher documents (.mxd and .pmf files);
- Metadata Service;
- Route Service.

For now the application relies on the Image Service, but in the future a Metadata Service and a Route Service will be also added.

Upon receiving a request the ArcIMS Spatial Server is performing one or more functions.

The image function when called it generates an image based on the data included in the map configuration files created in ArcIMS Author or in ArcMap. For our application the image format is Joint Photographic Experts Group (JPEG), but there are also other possibilities like the Portable Network Graphics format (PNG) or the Graphics Interchange Format (GIF). When a client searches for something the query function is invoked and features that are to match the criteria stated by a user will be returned. The extract function creates shapefiles from the selected features.

There are more functions that the Spatial Server can perform but different ArcIMS Services need to be used. For example the feature function – it streams map features to the client. Practically vector data is streamed to a Java applet in the requesting client and it is only available with the Feature Service. The route and geocode functions depend on the Route Service. The metadata function will not be available unless a Metadata Service is created.

There are two background processes (they run as services on Windows and as daemons on UNIX) that support the ArcIMS Spatial Server: ArcIMS Monitor and ArcIMS Tasker.

ArcIMS Monitor – will track the state of any ArcIMS Services and upon rebooting it will automatically restore those services.

ArcIMS Tasker – removes the output files stored in the output directory of the website. The removal of the files takes place at a specified time interval.

ArcIMS Application Server

The ArcIMS Application Server handles requests and tracks which service is running on which ArcIMS Spatial Server. The ArcIMS Application Server is written as a Java application and runs as a Windows service or as a daemon on UNIX.

ArcIMS Application Server Connectors

The ArcIMS Application Server Connectors provide a communication pipeline between the a Web Server and the ArcIMS Application Server. It also ensures the ArcIMS Application Server always receive requests in ArcXML.

There are five such connectors:

- ArcIMS Servlet Connector – these is the default connector. The Servlet Connector uses the Servlet engine to provide a communication link between the Web server and the ArcIMS Application Server. It accepts and passes ArcXML.
- Java Connector – allows communication between a Java application, Java Servlet, or Java Server Pages (JSP) client and an ArcIMS Server. The Java Connector is comprised of the Java Connector Object Model and the Java Connector Tag Library.
- ActiveX Connector - is COM DLL that can be used in Visual Basic or Active Server Pages (ASP) applications.
- .NET Link – it consists of classes and functions that can be used to build connections from the .NET applications to ArcIMS Application Server through either HTTP or TCP connections. Once the connection is established, ArcXML requests can be sent and ArcIMS responses can be received.
- ColdFusion Connector - The ColdFusion Connector allows the ArcIMS Application Server to communicate with the ColdFusion Server, which then communicates with the Web server. ColdFusion

Connector uses a set of custom ColdFusion tags specific to ArcIMS for generating ArcXML requests and responses.

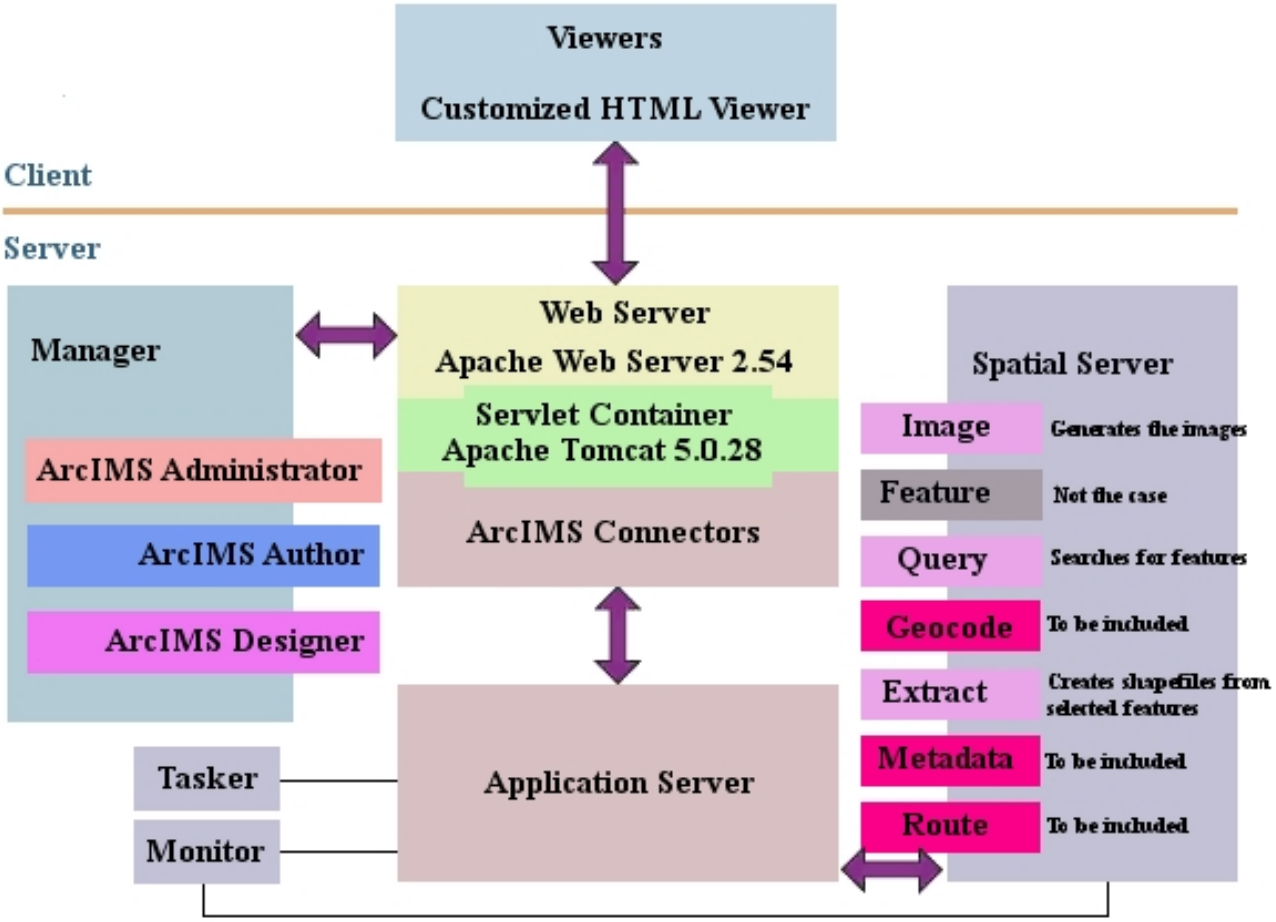

Figure 1. Application Architecture

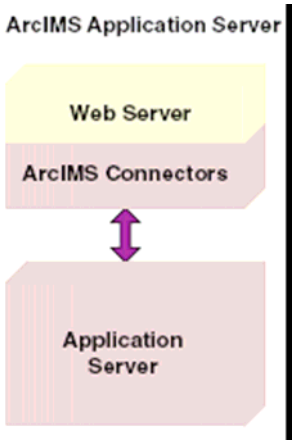

Figure 2. ArcIMS Application Server

ArcIMS Manager

ArcIMS Manager is a Web-based application that supports the three main tasks you perform in ArcIMS – map authoring, Web site design, and site administration. These tasks can also be completed using the three independent ArcIMS applications – ArcIMS Author, ArcIMS Administrator, and ArcIMS Designer.

ArcIMS Viewers

When a user visits the Web site created, he will see a map inside a viewer embedded in that Web site. There are three viewers predefined and included with ArcIMS: HTML Viewer, ArcIMS Java Custom Viewer, and the ArcIMS Java Standard Viewer. One can also build from scratch such viewers.

CASE STUDY

Bucharest is facing challenges typical for a major city with a growing population. A tool was needed to better handle the eventual problems. Such a tool should help the municipality in their daily actions concerning the city well-fare. From the local government's point of view such a tool should allow them to perform the following:

- Full text search and query the data;
- Building customized maps from different layers and print them;
- Make observations directly on the maps;
- Advanced measure tool – display the measure values along the measurement path;
- Aria calculator;
- Route calculator;
- Select features by different shapes (rectangle, line, circle, custom shape) and create buffers;

Issues related to a centralized database, data security and portability were also mentioned

Counting more than 2 million people Bucharest is divided in six administrative districts. One of these districts represents the study area for this paper. All things have a start, in our case we started by gathering and processing data. The result of this process materialized in nine thematic layers (one raster and eight vector) as it follows:

An IKONOS Satellite Image of Bucharest (1 meter resolution), shapefiles for the following elements: the street network, railroads, hydrography, boundary, points of interest, zip codes, functional areas. The application was built upon these layers, using the ArcIMS Author a map configuration file was created. The map configuration file (it has a .axl extension) contains information about the layers (storage paths) and the drawing instructions for each layer in an ArcIMS Service.

Having the map configuration file an ArcIMS Service can now be created in ArcIMS Administrator. The service will be based on this map configuration file. As described earlier in this paper there are a number of ArcIMS Services. For the present application an Image Service was created.

The next step is to build a new client viewer or customize an out-of-the-box viewer included with ArcIMS. Using the ArcIMS Designer a HTML Viewer was created in just a few steps. From this point on all efforts will concentrate on customizing the viewer, fitting it to the application, adding extra functions, and optimizing it using HTML and Java Script.

The viewer is organized in five categories: table of content, toolbar, map frame, overview map, and a text frame.

The table of content, positioned on the left, lists all layers included in the map configuration file and their status – visible, hidden, active or inactive. Using a Java Script nested menu function the layers can be grouped by categories. Each group can be collapsed or extended. The layer list allows the user to activate a layer, make it visible or hidden, thus obtaining different layouts. Each time a layer is turned on or off (visible or hidden) the map frame needs to be updated to display the changes made. There are two way of doing this: by checking the Auto Refresh check box – in which case the map frame is auto-updated when a layer is turned on or off; or uncheck the Auto Refresh check box – in which case the user has to press the Refresh button after he has selected which layer will be displayed or not.

The toolbar is organized in three categories: navigation, query and selection. With the navigation tools the user will be able to zoom in or out, to navigate through the zooms previously applied or to pan the map.

The query tools include: identify, find, measure; tools that can be customized and enhanced. For example the Identify tool allows a user to interactively request information about a feature on an active layer. This tool can be easily customized to allow the user to query and identify all layers not just the active one. The find tool was enhanced and auto-completion functionality was added. The moment a user enters a query value in the textbox and releases the key, the HTML Viewer sends a query request to the ArcIMS Server for any matching value. For example if the "t" key was pressed the ArcIMS Server would receive a query and all values beginning with the letter "t" would be returned. If there are any

matching values they will be displayed in a list and the user can chose the value from the list. The measure tool was also enhanced, the value of the measurement being displayed along the line indicating the measurement path.

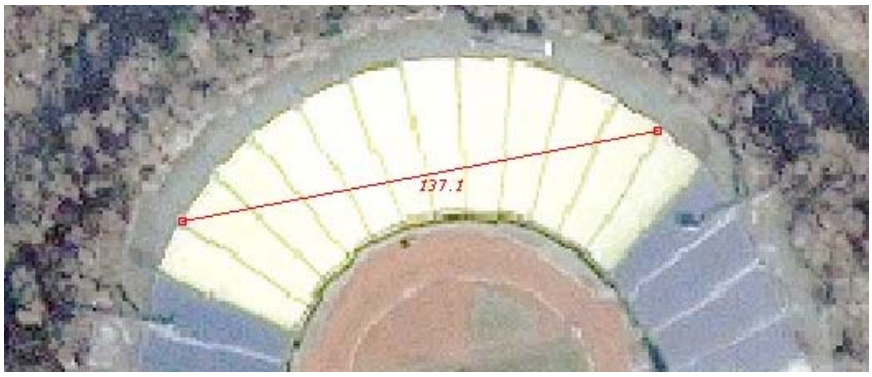

Figure 3. Measure tool

The selection tools include: select by rectangle, select by circle, select by polygon, select by line, deselect and buffer tool. The select by circle tool displays the radius of the circle which makes it a very useful tool.

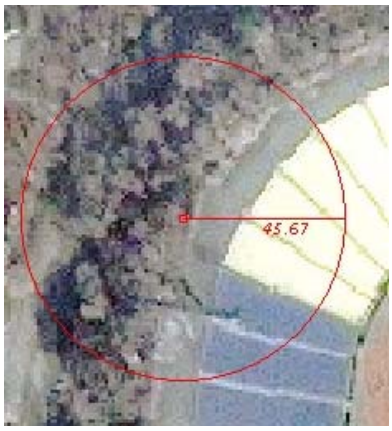

Figure 4. Select by circle tool

There is also a print tool which allows the user to create a ready to print layout with a customized title, a legend, the overview map and the map frame.

This tool is at its beginnings. There is a lot of work ahead and functionalities to be integrated. There is also a TO DO list which we're hoping will be completed in our next version of the application.

Among the advantages of such an application we could mention: data is centralized, managing data is more easier; data is portable, it can be accessed in the field; no GIS software is required on the client side.

To Do

- Extending the database – including buildings, parcels, increasing the number of points of interest etc.;
- Loading all data in ArcSDE;
- Implement the annotation tool;
- Designing and applying a security policy;
- Including a Route Server – for route calculations;
- Improving the client viewer – adding more functionalities and make it more intuitive;
- Extending the application for the entire Bucharest.

References

Some of the terms that were used in this paper need to be clarified:
Geographic Data – is about much more than electronic pictures of maps. The geographic data that describes our world allows for city planning, flood prediction and relief, emergency service routing, environmental assessments, wind pattern monitoring and many other applications. Geographic data is processed with Geographic information system (GIS) software which can, as one aspect of its functioning, produce maps.
A server – In computing, a server is:
A computer software application that carries out some task on behalf of yet another piece of software called a client. In the case of the Web: An example of a server is the Apache web server, and an example of a client is the Internet Explorer web browser or the Mozilla web browser. Other server (and client) software exists for other services such as e-mail, printing, remote login, and even displaying graphical output.
Over the years, the term has been misinterpreted (but in common usage now) to also mean the physical computer on which the server software runs. Software ultimately requires computer hardware to run, and originally server software would be run on a large powerful computer such as a mainframe computer or minicomputer. These have largely been replaced by computers built using a more robust version of the microprocessor technology than is used in personal computers, and the term "server" was adopted to describe microprocessor-based machines designed for this purpose. In a general sense, "server" machines have high-capacity (and sometimes redundant) power supplies, a motherboard built for durability in 24x7 operations, large quantities of ECC RAM, and fast I/O subsystems employing technologies such as SCSI, RAID, and PCI-X or PCI Express. It is important to note, however, that computers referred to as "servers" do not necessarily run any server software, nor is it required that server software only be run on these types of computers.
A web server – can mean one of two things:
A computer that is responsible for accepting HTTP requests from clients, which are known as web browsers, and serving them web pages, which are usually HTML documents.
A computer program that provides the functionality described in the first sense of the term.
A web map server– A web map server (WMS) is a web application which provides portrayal of geographic data which is stored on the server. This data can be stored in a variety of data formats but is served in a limited number of image formats. This provides the useful ability to restrict access to data since only a picture of the data is sent to the user and the high quality vector or raster data remains safe on the server. This is in sharp contrast to a Web Feature Server (WFS) which returns the features of the data as features. On the other hand it is possible to build thin client applications which only have to show georeferenced images instead of rendering the features.
The Java Servlets API allows a software developer to add dynamic content to a web server using the Java platform. The generated content is commonly HTML, but may be other data such as XML. Servlets are the Java counterpart to dynamic web content technologies such as CGI or ASP. It can maintain state across many server transactions. This is done using HTTP Cookies, session variables or URL rewriting. The Servlet API defines the expected interactions of a web container and a servlet. A web container is essentially the component of a web server that interacts with the servlets. The web container is responsible for mapping a URL to a particular servlet and ensuring that the URL requester has the correct access rights. A servlet is an object that receives requests and generates a response based on the request. The API defines HTTP subclasses of the generic servlet requests and responses as well as an HTTP session object that tracks multiple requests and responses between the web server and a client. Servlets may be packaged as a Web application. Moreover, servlets can be generated automatically by Java Server Pages (JSP), or alternately by template engines such as WebMacro. Often servlets are used in conjunction with JSPs in a pattern called "Model 2", which is a flavor of the model-view-controller pattern.
A servlet container – comprises essentially the component of a web server that hosts and interacts with Java servlets. A servlet container controls the servlets that are deployed within the Web Server and is responsible for forwarding the requests and responses for them. It has the functionality of mapping a URL to a particular servlet and of ensuring that the process requesting the URL has the correct access rights.
A markup language – combines text and extra information about the text. The extra information, for example about the text's structure or presentation, is expressed using markup, which is intermingled with the primary text. The best-known markup language in modern use is HTML (Hypertext Markup Language), one of the foundations of the World Wide Web. Historically, markup was (and is) used in the publishing industry in the communication of printed work between authors, editors, and printers.
XML – The Extensible Markup Language (XML) is a W3C-recommended general-purpose markup language for creating special-purpose markup languages, capable of describing many different kinds of data. It is a simplified subset of SGML. Its primary purpose is to facilitate the sharing of data across different systems, particularly systems connected via the Internet. Languages based on XML (for example, Geography Markup Language (GML), RDF/XML, RSS, MathML, Physical Markup Language (PML), XHTML, SVG, MusicXML and cXML) are defined in a formal way, allowing programs to modify and validate documents in these languages without prior knowledge of their form.