

Web-Based GIS Viewer Using MapObjects Java Can Reduce Operating Costs

Cheryl Spencer
Aralola Akinmade

Abstract

As more users get introduced to GIS technology, the line between the casual and advanced user broadens. Organizations with 50 or more employees that actively utilize GIS technology can easily add at least \$100,000 in maintenance and licensing fees to their annual budgets, even though most of the GIS software is under utilized by its users.

This presentation will discuss how Indianapolis/Marion County, Indiana built a web-based GIS data viewer using ESRI MapObjects Java to replicate basic desktop GIS functionality for more than 200 casual users at a large metropolitan government agency. By developing an alternative for its casual users, the agency was able to funnel money saved from licensing and support costs, to the purchase and implementation of cutting-edge ESRI GIS (ArcGIS Server, ArcGIS Engine, etc) for their advanced users.

Introduction

The City of Indianapolis/Marion County has over 500 GIS users within the enterprise. A third of these users are casual users that use ArcView to locate an address or print a simple map. In 2001, the GIS group identified these users through interviews while updating the strategic plan. Criteria for determining casual users was based on the need to edit data, produce high quality cartographic maps, or perform sophisticated analysis. If a user met one or more of these criteria they were not considered a casual user. The City/County currently uses an ArcView extension called DataViewer for casual users. This extension strips out most of the high end analysis tools so that users are not confused by tools they do not use. With the decision not to continue support of ArcView 3.x, due to underutilization, the City/County saw a need to upgrade or replace this application.

Reasons for Choosing a Web Product

The main reason for choosing a web replacement for casual ArcView users was cost. The City/County has spent a significant amount of money on ArcView licenses. As mentioned earlier, many of these people aren't getting the maximum benefit out of ArcView. Deploying an application through the Intranet allows the City/County to reduce license costs and at the same time cover most or all of the needs for casual users.

A second reason for choosing a web product was the ability for users to access the application easily. The users of our application are no longer restricted by needing an ArcView license. Anyone who has Internet Explorer, Intranet access and Windows 2000 or XP has the ability to use the application.

A third reason for choosing a web replacement was the ability to perform easy upgrades. With ArcView, a technician has to upgrade all machines when new versions become available. A web product allows us to install a new version on the server and the next time the user launches the application they will be prompted to perform a silent upgrade before they can continue using the application.

Alternatives

The City of Indianapolis/Marion County considered several different platforms on which to create the Intranet Data Viewer (IDV) on. The table below (Table 1) shows the platforms that were considered and the pros and cons of each one at the time of the decision in 2003.

	ArcGIS Engine 9	ArcGIS Server 9	ArcExplorer Java 4.0.1	ArcIMS Java Viewer 4.0.1
Cost – Licensing	(??) + License (??)	Additional license (??)	Deployment MO Java (?)	(++)
Existing Competence	Beta (-)	Beta (-)	(+)	(++)
Existing Programming Skills	VB,.NET, (++)	VB,.NET, (++)	Java (-)	ASP, JSP, (+)
Long range viability	.NET (++)	.NET (++)	(+)	(+)
Stability	(--)	(--)	(+)	(+)
Delivery Date for Apps	(--)	(--)	(+)	(+)
Existing Code/Sample	(-)	(-)	AEJ Base Code? (+)	Former vendor (+)
Ultimate functionality	(+)	(+)	(-)	(-)
Access Local Data	(++)	(-)	(++)	(+)?
Access Network Data	(+)	(+)	(++)	(++)
Rendering	(++)	(++)	(+)	(+)
Ease of installation	(+)	(++)	(+)	(++)
Ease of upgrades	(++)	(++)	(-?)	(++)
How easy to program functionality	(++)	(++)	(-)	(+)
How easy to program GUI	(++)	(++)	(+/-)	(+/-)
Tech Support	(-)	(-)	(+)	(++)

• Table 1

Initial Plan

From the matrix, the City/County decided to go with the ArcIMS Java Custom Viewer. The initial plan had a three phase development schedule. The idea was to give users a reasonable amount of functionality in the first phase and then add to it. All of the tools to be developed were on the interface, but if a tool wasn't complete a message would pop

up stating that the functionality wasn't currently supported. This gave the users an idea of what tools would be available by the end of the project.

The intent was to keep the look and feel the same for all phases, to ease transition from the old ArcView DataViewer to the new Intranet Data Viewer (IDV). This was not possible in every case.

After development began, problems were encountered. Some of the initial requirements such as multiple layer selection, select by point, and zoom to collective extent of selected features, could not be met. The ability to view an attribute table, like in ArcView, was not available and map printing was limited to a static map and table of contents graphics. At this time, the City/County reviewed their options before continuing with the application.

Second Try

Based on the limitations of the Java Custom Viewer, another solution was needed. Plan 'B' was formed when it was discovered that the ArcIMS Custom Java Viewer applets were built on MapObjects Java. The City/County along with our vendor, Woolpert, Inc., decided the new path to take was to build our own applets using MapObjects Java 2.0. This allowed access to more base objects and gave much more flexibility to create the tools we needed.

Challenges

The biggest challenge of the IDV application was the silent installation. When the user goes to the URL of the IDV for the first time, the user is prompted to install the application. After two clicks from the user, the application silently installs a Sun JRE and custom applets on the user's computer. It also installs a bookmark in the user's Favorites directory for easy access to return to, and a key in the user's registry with a version number. Second and subsequent times the user accesses the IDV, the application checks the version number. If the numbers match, the application is launched. If the numbers don't match, an upgrade has been placed on the server and the user will be prompted to upgrade. The upgrade installation only requires two clicks from the user and the custom applets are upgraded. The root cause of one installation problem was due to the Microsoft/Sun competition involving the Java Virtual Machine (VM). Other problems included file corruption across the network. Getting all pieces and parts of the installation to work together correctly was a big challenge.

The second challenge was the rewriting and reinventing of the wheel. As mentioned earlier, the code was written on top of MapObjects Java. There was no existing code base or object model to start with. The code was written to mimic ESRI's Custom Java Viewer applets but was extended to accommodate other objects we required. There was significant time spent initially to create the base objects.

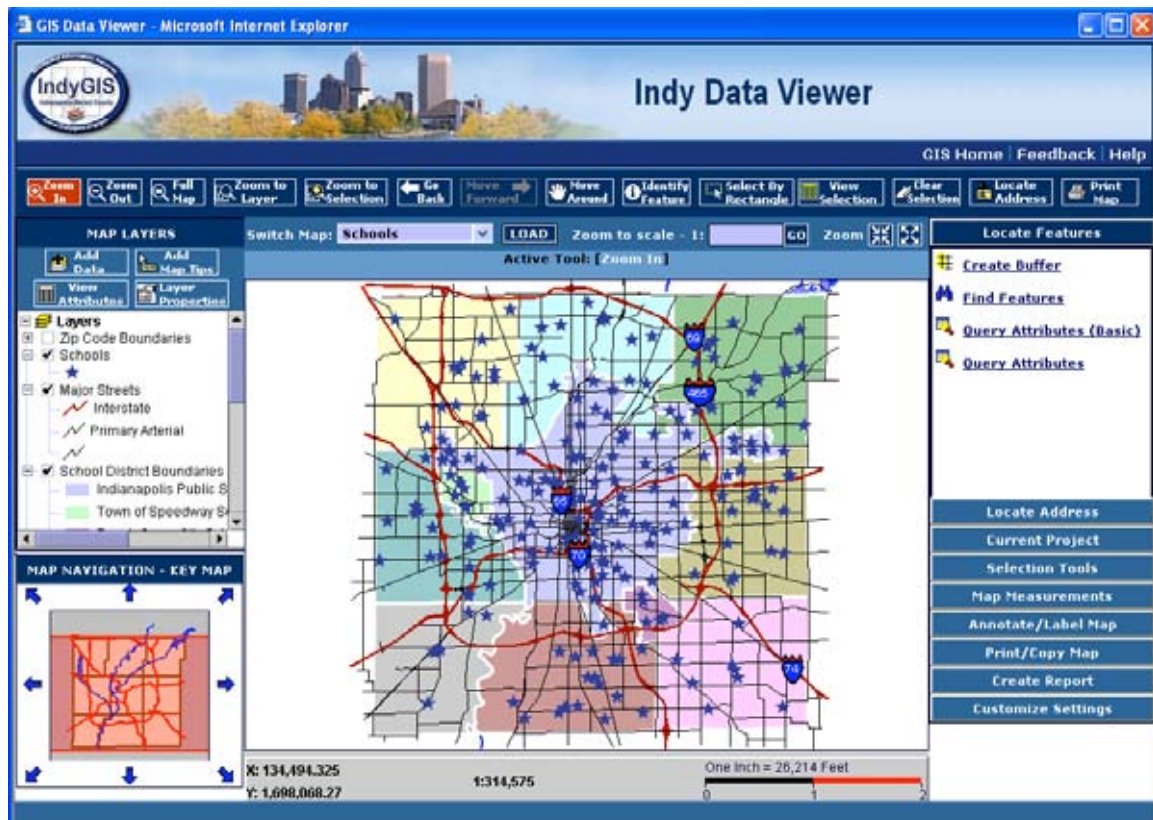
The third challenge was dealing with intrinsic bugs in MapObjects. The scale of the project was very large and covered many different areas allowing for several problems to be discovered during the development process.

Benefits

Despite the challenges, several benefits were reaped from the project. First and foremost, ArcView license costs were reduced. The City/County has saved \$135,000 per year for the past four years, including the current year, for a total license savings of \$540,000 to date. Secondly, desktop installations by a technician are no longer needed. Users can download the application when it is convenient for them without needing a license. All installations are silent, behind the scenes, over the Intranet. Thirdly, all ArcIMS Java Custom Viewer limitations were overcome. The wheel is now our own wheel to fix and extend as we wish.

Interface

Figure 1 shows the user interface of the IDV.



• Figure 1

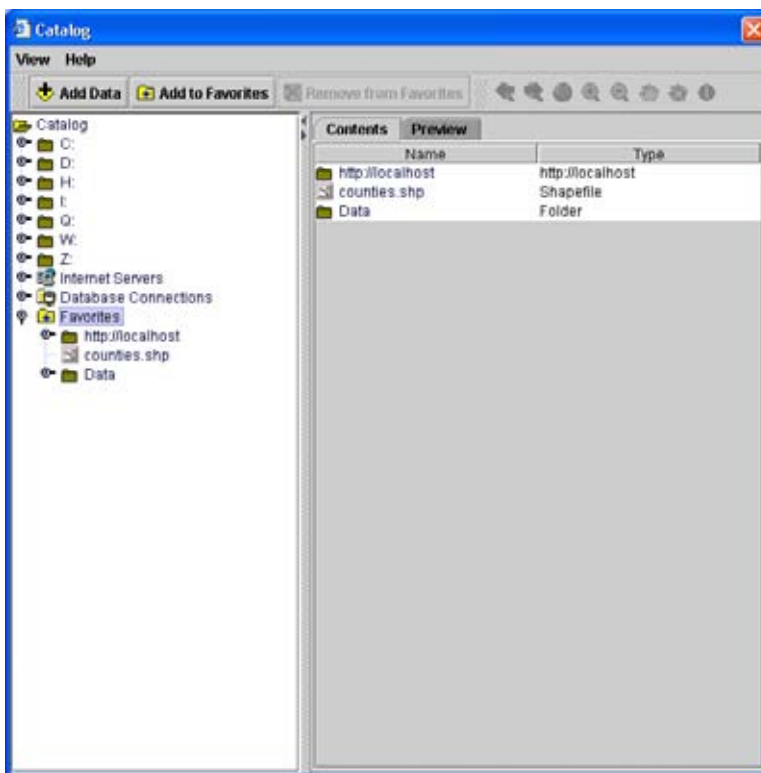
Key IDV Functionality

The IDV provides a plethora of functions. In addition to basic mapping functionality the IDV also has some key features that differentiate it from the ArcView DataViewer application that was used before. Examples of key features are described below.

Adding Data

The 'Add Data' dialog looks similar to an ArcCatalog dialog (Figure 2). Like an ArcCatalog dialog, the IDV dialog gives the user the ability to preview data before adding it to the map.

Through the dialog, users can access any network or local shapefiles as well as SDE and Internet data. The dialog also has a Favorites folder to store favorite connections once they are established.



- Figure 2

Viewing Attributes

Viewing attributes in the IDV is similar to viewing attributes in ArcView. Once the attribute table is open, tools include calculating statistics, summarizing the table, exporting the table, and show/hide columns (Figure3).

FID	LAYERTYPE	AREA	PERIMETER	NE_ID	NE_ID	S
1	POLYGON	1.764697E10	713534.6	2	38	23
2	POLYGON	1.126728E10	525428.0	3	68	23
3	POLYGON	8.187878E9	610382.7	4	75	23
4	POLYGON	1.083091E10	670889.9	5	79	23
5	POLYGON	6.978437E9	718236.4	6	108	23
6	POLYGON	3.909856E9	576850.3	7	140	23
7	POLYGON	1.170278E7	16748.35	8	146	23
8	POLYGON	4.54607E9	414267.8	9	147	23
9	POLYGON	5.515425E9	491504.7	10	175	23
10	POLYGON	5610825.0	12741.32	11	179	23
11	POLYGON	4.786394E9	368090.1	12	180	23
12	POLYGON	8952078.0	16950.58	13	185	23
13	POLYGON	1.915939E9	250071.0	14	193	23
14	POLYGON	2.43013E9	265433.8	15	203	23
15	POLYGON	7597109.0	12431.99	16	206	23
16	POLYGON	2.87781E9	182563.0	17	206	23
17	POLYGON	1.757833E9	220863.0	18	211	50
18	POLYGON	6632841.0	13536.06	19	214	23
19	POLYGON	2.04063E7	19202.21	20	217	23
20	POLYGON	1.865165E9	204802.3	21	223	50
21	POLYGON	3.709243E7	42139.93	22	228	23
22	POLYGON	7318121.0	11936.57	23	232	23
23	POLYGON	7.215103E7	56880.16	24	235	23
24	POLYGON	1.746498E9	177318.4	25	242	50
25	POLYGON	2.656577E7	32769.91	26	243	23

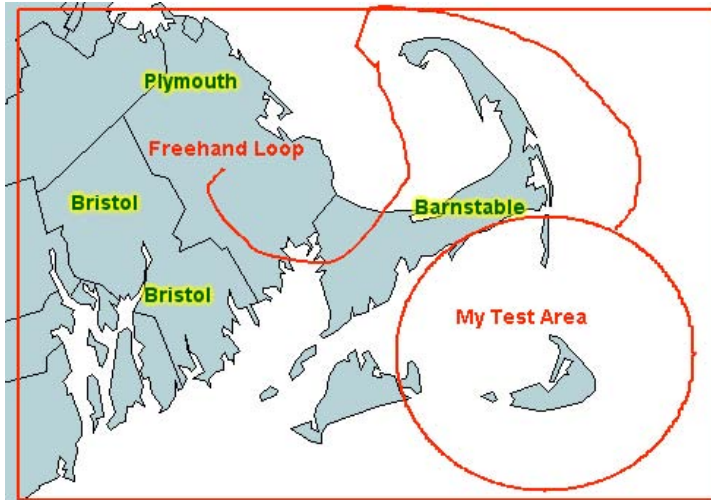
- Figure 3

Single and Batch Geocoding

The IDV allows users to geocode a single address or batch geocode addresses from a file. File formats for batch geocoding include comma separated values (CSV), Microsoft Excel, and dbf. Geocoding is currently done against multiple ArcIMS geocoding services in a fallback fashion. Testing has been done to geocode from an ArcGIS server web service. Preliminary testing has shown the geocoding to be faster with the ArcGIS server web service, but full implementation has not taken place yet.

Annotation and Manual Labeling

Another key function the IDV provides is the ability to manually label a feature or add annotation to the map. The user is prompted to set label or annotation properties before labeling or annotation begins. Figure 4 shows an example of several annotation types.



- Figure 4

Printing and Reporting

Users can print custom maps from the IDV. The tool was based around an ArcView custom extension that the users are familiar with. The tool prompts the user to select paper size, orientation, title, subtitle, and creator name. The application then generates a map for preview. The user can then choose to send the map to a printer or export the map for inclusion in other documents.

The IDV also has a reporting tool. This tool allows users to generate a report based on layer attribute information. The report can run on selected information or all layer information. It also has the option of including a map in the report.

Other Functionality

Behind the scenes an activity logger runs and keeps track of many IDV activities. If the application generates any error messages, the activity logger generates an email for critical errors. The activity logger also keeps track of how many times a certain tool or function is used. This information will be used in future generations of the IDV to determine how frequently a tool is used instead of relying on user feedback.

Right clicking on a layer in the Table of Contents causes a context menu pops up. One of the choices on the context menu is 'View Metadata'. If metadata is available for a layer or image service the application will open a new window displaying the metadata.

Documentation

Help, Online Tutorials, Frequently Asked Questions (FAQ), and a Known Bug List were compiled in time for deployment of the application. Time was spent to develop the documentation so it is easily accessible and easy to use. Help documentation is context

sensitive based on the tool the user has activated in the application. Tutorials are available for users getting started and are also used for training material. FAQ contains 'What's New' information for those used to ArcView, as well as other information we felt might be less intuitive to figure out or where users might get stuck. All documents are available from a menu on the toolbar. The bug list contains a list of ESRI bugs that we don't have workarounds for. The goal was to help lessen frustration and let the user know that we recognize the limitations of the tool.

Conclusion

Despite the challenges encountered during the development of the Intranet Data Viewer application, the overall goal of saving the City of Indianapolis and Marion County money in license fees was achieved. Before the application went live the City/County had already broke even with development costs by not upgrading ArcView licenses for the development duration. From this point on, the savings will add up. The wide range of basic GIS functionality provided by the application suits a large number of casual users without the extra license costs.



Cheryl Spencer
Web Developer/GIS Analyst
Indianapolis/Marion County Information Services Agency
200 E Washington St., Suite 2441
Indianapolis, IN 46204
Phone – (317) 327-5636
Fax – (317) 327-4954
Email - cspencer@indygov.org



WOOLPERT
Aralola B. Akinmade
GIS System Designer
Woolpert, Inc.
2800 Shirlington Rd, Suite 500
Arlington, VA 22206
Phone - 703.820.3840
Fax - 703.820.3894
Email - aralola.akinmade@woolpert.com