



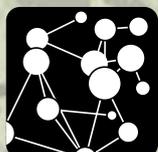
# UCL

## **WORKING PAPERS SERIES**

**Paper 57 - Jul 02**

**Delivering light-weight online  
geographic information  
analysis using ArcIMS**

ISSN 1467-1298



**CASA**

## **Delivering light-weight online geographic information analysis using ArcIMS**

Sanjay Rana  
Centre for Advanced Spatial Analysis  
University College London  
1-19 Torrington Place  
London WC1E 6BT  
UK  
s.rana@ucl.ac.uk, +44-20-76791813 (Tel), +44-20-78132843 (Fax)

### **1. Introduction**

As of July 9, 2002, more than 300 websites, which provide automated mapping and facilities management over the internet, are registered in the ESRI Internet Map Server (IMS) user registry [1]. But it won't be an exaggeration to assume that this is only a tiny fraction of the actual number of IMS sold and used over the world. In fact, realising the potential scope and issues of this new form of geographic information delivery, the International Cartographic Association has formed a Commission dedicated to Maps and the Internet [2]. The IMS software has also kept pace with the growing demand and progress in technology. IMS has evolved from the simple HTML image maps to sophisticated servlet-driven mapping services. During this period, a very significant event happened with the arrival of ArcIMS. With the ArcIMS the use of IMS changed from merely an interactive visualisation and query of the spatial databases to a platform for sharing remote spatial databases. In other words, the ArcIMS made the scale of operation of IMS global.

While the accessibility and usability of the IMS have certainly increased in the past 5-6 years, the functionalities provided by IMS however remained fairly static. A widely felt but unreported criticism of IMS has been that the lack of a broader range of spatial analysis functions (except the usual buffering) in the off-the-shelf IMS installation. Therefore a couple of eyebrows always tend to rise if the word "Internet GIS" is used for IMS. Oddly enough, the ArcIMS 3.x and the older versions carry this limitation as well. Essentially, IMS suffer from the limitation for not being able to allow simultaneous update and dynamic manipulation of the thematic content of the online maps. For example, in the case of ArcIMS while there are provisions for complex scale-dependent rendering, it is not easily possible to manipulate (for example add layers or delete layers) the crucial *axl* file of a mapping service on the fly. The recent launch of ArcIMS 4 promises to bridge this gap by allowing an enhanced integration with the ArcGIS. The aim of this article is to propose a generic framework, which makes the link between an IMS and a standard GIS, to provide geographic analysis in online maps. We will take the example of such a framework developed for ArcIMS.

### **2. Online mapping enabled for geographical analysis (OMEGA)**

Due to the computational complexity in automated mapping, the earliest attempts to perform online spatial analysis did not involve online mapping. So for example, the website developed by Paul Brown at University of Southern California [3] invoked ArcInfo functions from a website using server-side Perl CGI scripts without an online visualisation of the outputs. One many still find some data providers' websites that follow this approach. This loose coupling of the Internet and the spatial analysis functions did not migrate completely into the modern IMS. As mentioned in the introduction, ArcIMS does not support a dynamic update of the content of the online maps without having to shut down the map service. This essentially meant that a spatial analysis, which produced new outputs, could not be done on an online map. Consequently, online mapping is unable to grow as a medium of collaborative research and training.

We propose a simple generic framework, which when constructed allows a coupling between an IMS and a GIS of user-defined complexity. We call the framework *Online Mapping Enabled for Geographical Analysis* or OMEGA. The flowchart for the operation of

the framework in the case of an ArcIMS-ArcInfo combination and its various components is shown in Figure 1. The idea of OMEGA originated from our discussions with Matt Doggett at the Oregon State University who is perhaps the first to establish a clever loose coupling between ArcIMS and ArcInfo with a couple of JavaScripts, Perl scripts and the ArcInfo Inter-application communication (IAC) commands. To see Matt's website write to him at mdoggett@coas.oregonstate.edu.

In a conventional ArcIMS mapping service, an ArcIMS client such as Java Viewer or HTML viewer displays the geographic information (GI) database using the client-side access route (See Figure 1). As mentioned earlier, the client-side access allows only a limited set of spatial analysis on the GI database. The OMEGA framework enables geographical analysis on online mapping in the following way. There are four main components in this OMEGA framework namely the Customised ArcIMS Client, CGI Script, ArcInfo, and GI Database (Figure 1). The design of the OMEGA framework is typical of many other customised IMS. The ArcIMS client is customised to include an interface of the desired geographical analysis operation. This could simply be an html FORM with the relevant html FORM INPUT types. Although not necessary, the html FORM can be loosely or permanently coupled with the content of the online map. On submission, the contents of the html FORM are passed onto a CGI script. The CGI script has three main handlers or subroutines namely the request handler, response handler and the error handler. The request handler parses the html FORM inputs and if an error is found it passes the error to the error handler, to be reported to the user. In the case of a proper input, the request handler constructs one or many AML macros (based on the user inputs from the html FORM) and invokes ArcInfo to process the AML macro(s). Note that in many organisations, ArcInfo is installed on the LAN, away from the web server, therefore, appropriate inter-application communication scripts will be required to access the ArcInfo from the CGI script on the web server. If IAC sounds too much, then an easier option is to install ArcInfo on the same machine as your web server. In this way, calling ArcInfo is much simpler and the only requirement is that the proper environment variables are set up before calling the ArcInfo. See [4] for ways to figure out the environment variables in your ArcInfo installation.

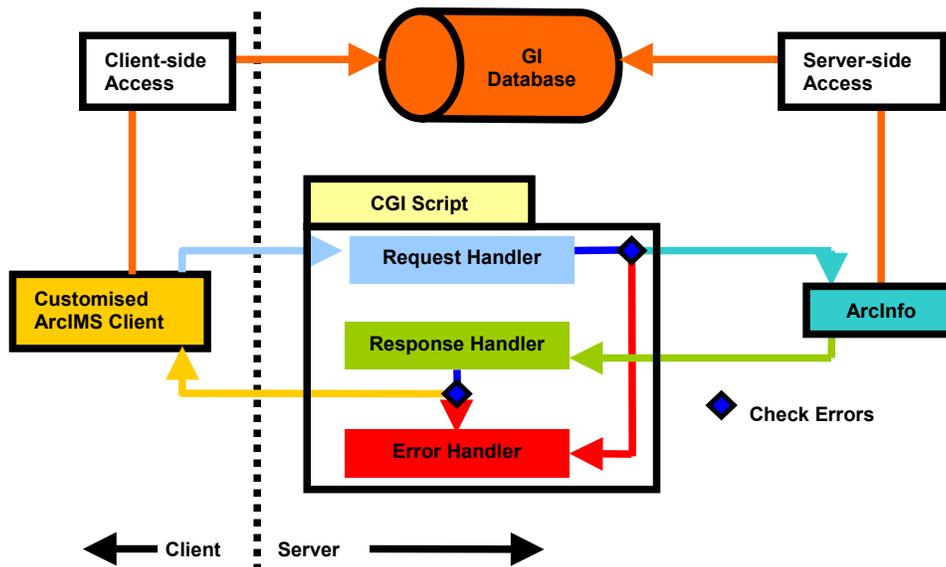


Figure 1. Typical construction of customised Internet Mapping Servers.

As mentioned earlier, ArcIMS does not allow addition or removal of layers in the map content via the ArcIMS client therefore the layer which holds the output of server-side geographical analysis has to be present in the online maps' *axl*. In other words, this means that a layer is kept in the *axl* of the online map solely as a placeholder for the output of the geographical analysis. Once ArcInfo has finished processing the AML macro, the control returns to the CGI script and the response handler in the CGI script makes a check if the ArcInfo processing was successful. In case of an error, the error handler informs the user otherwise the response handler requests the ArcIMS client to refresh so that the output layer from the GI-database is reloaded and accessed by the user. The next section shows an example of the implementation of the OMEGA framework for a simple overlay operation on ArcInfo grids using an HTML Viewer ArcIMS client.

### 3. A simple example

The simple example is a website (<http://vole.casa.ucl.ac.uk/website/towncentre/>), which is a research tool in the Project Sustainable Town Centres [5] at the Centre for Advanced Spatial Analysis, UK. One of the aims of the project is to allow researchers to combine various statistical surfaces (ArcInfo grids) and visualise the concurrence of the various statistics. Figure 2 shows the HTML Viewer client and the html FORM to allow the combination of the grids. The html FORM is invoked by clicking a button on the toolbar on the HTML Viewer. When the user submits the html FORM, a CGI script *mapcalculator.pl* is invoked (See code below). The CGI script first sets the right environment variables (step 1) and then constructs an AML *mapcal.aml* based on the map calculation expression specified by the user (step 2). ArcInfo then executes the AML and the resultant coverage *output* is overwritten with the output of the AML. The response from ArcInfo is parsed by the response handler to

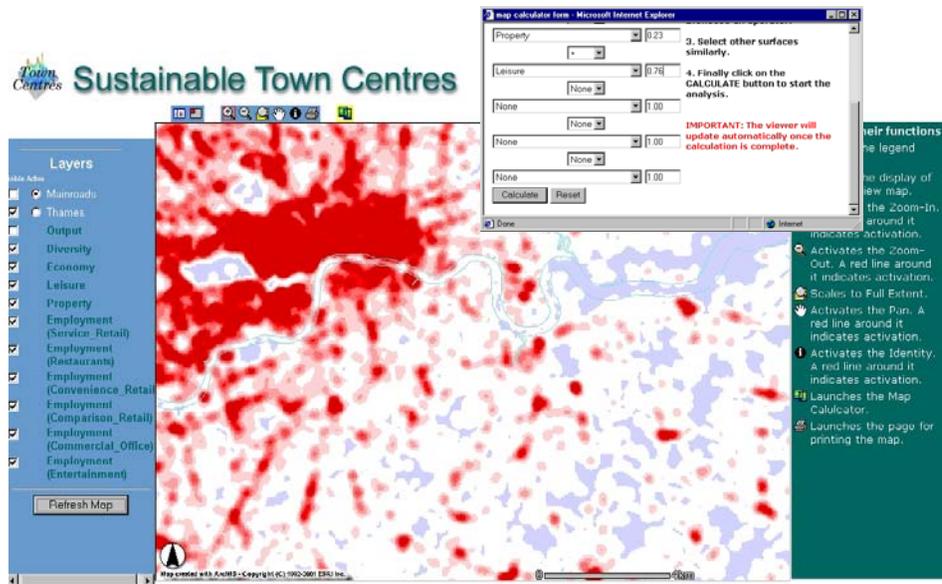


Figure 2 Example of a customised ArcIMS HTML Viewer client and html FORM in a simple OMEGA framework.

check if the map expression was completed successfully. If an error is found then the error handler reports an error otherwise the JavaScript function for refreshing the map (Function `updateLayers()` included in the Table of Contents FRAME of the webpage), is invoked (step 4).

## Listing of mapcalculator.pl

### Step 1 – mapcalculator.pl sets environment variables

```
$ARHOME = "D:/arcgis/arcexe81";
$ENV{'ARHOME'} = $ARHOME;
$ENV{'ARHOME_USER'} = $ARHOME;
$ENV{'ATHOME'} = "$ARHOME/arctools";
$ENV{'HOMEDRIVE'} = "C:";
$ENV{'HOMEPATH'} = "/";
$ENV{'TEMP'} = "C:/TEMP";
```

### Step 2 – mapcalculator.pl makes mapcal.aml

```
$mapcal_aml = 'd:/apache/apache/cgi-bin/mapcal.aml';
open(AML,">$mapcal_aml");
print AML "&severity &error &routine error\n";
print AML "&severity &warning &routine error\n";
print AML "output = $mapexpression\n";
print AML "&return\n";
print AML "&routine error\n";
print AML "&type errorfound\n";
print AML "quit\n";
print AML "quit\n";
print AML "&return &error errorfound\n";
close(AML);
```

### Step 3 – mapcalculator.pl invokes ArcInfo to execute mapcal.aml

```
$retstr = `arc "&r d:/apache/apache/cgi-bin/mapcal.aml``;
```

### Step 4 – mapcalculator.pl refreshes the map

```
print "Content-type:text/html\n\n";
print "<html><head><script language=\"javascript\">";
print "theTOC=opener.parent.TOCFrame\";";
print "</script></head>";
$error = index($retstr, "errorfound");
if ($error != -1)
{
    print "An error has occurred while trying to run the scenario.\n";
}
else
{
    print "<body onload=theTOC.updateLayers();></html>";
}
```

The above simple methodology can be used straightaway in the case of a single online user but a number of steps will have to be taken in the case of multiple users. These steps include the access of the GI Database via ArcSDE to ensure versioning, use of cookies or some other session objects to handle individual AML requests.

## 4. References

- [1] ESRI IMS User Registry, <http://gis.esri.com/ims/index.cfm>, accessed on May 12, 2002.
- [2] International Cartographic Association Commission on Maps and Internet, <http://maps.unomaha.edu/ica/Maps%26Internet/default.html>, accessed on May 12, 200.
- [3] <http://support.esri.com/search/ViewListPost.asp?dbid=19339>, accessed on May 12, 2002
- [4] <http://forums.esri.com/thread.asp?c=64&f=792&t=51016#124925>, accessed on May 12, 2002
- [5] <http://www.casa.ucl.ac.uk/sustain/>, accessed on May 12, 2002.