

Over the last 10 years, GIS (Geographical Information Systems) has become an essential part of a wide range of computerized applications in a variety of fields, including environmental protection, urban and regional planning, business planning, transportation, demographic analysis, military operations, and optimal location planning.

The software market has responded to the increased demand for mapping and geospatial solutions with a diversified set of proprietary GIS packages, such as Arc/Info, ArcView, and ArcIMS (Environmental Systems Research Institute, Inc. [ESRI]); MapInfo (MapInfo Corp.); GeoMedia (Intergraph); IDRISI (Clark Labs); Autodesk GIS (Autodesk); and many others. Simultaneously, the open source community has been actively developing products to meet practically all the basic needs of GIS users. Moreover, GIS users and developers now have the luxury of choosing from a variety of open source components to build viable applications of their own.

GRASS

The centerpiece of this type of open source development is a

robust GIS package. It is widely recognized in the open source GIS community that GRASS (Geographical Resources Analysis Support System) is the largest, most powerful, and most reliable Free Software GIS project. GRASS is a raster/vector GIS and contains over 350 programs and tools that can create, manipulate, and store spatial data.

GRASS is well-documented and provides the opportunity for a person with basic C language programming skills to write and link his/her own modules to the package's internal "front end". The general structure of these modules permits their use in either an interactive or command line version. The ability to study the code organization of existing modules is very helpful and facilitates the creation of new modules.

As with many ongoing projects, the number of options, flags, and parameters has grown quickly, and it has become inconvenient to remain in the command line environment (which was the only GRASS option 10 years ago). The development of an interactive front end was therefore a very useful option. The graphical user interface (GUI) for GRASS is

"tcltkgrass" and is based on Tcl/Tk scripting language libraries. Working with GRASS here in the ITS Social Science, Statistics & Mapping Group, we were able—without significant expertise in Tcl/Tk—to expand the main tcltkgrass menu to include our own subset of modules with specific interactive windows.

POSTGRESQL/POSTGIS

GRASS has a limited internal database that is capable of supporting only a single attribute for each vector object or raster cell category. For projects with more complex data structures, the team of GRASS 5.7 developers has overcome this internal limitation by connecting GRASS to an external database management system such as PostgreSQL (released under the BSD license by The PostgreSQL Global Development Group). This approach parallels ESRI's move to integrate their own data structures with more powerful proprietary databases such as Oracle.

In general, connecting GIS to a relational database requires an additional software layer. For ESRI products like ArcInfo, ArcView, and ArcIMS, the functionality for such a gateway is provided by an

ArcSDE server. ArcSDE is known to work with a variety of different proprietary databases, including Oracle, Informix, IBM DB2, Sybase, and Microsoft SQL Server. In the case of a GRASS to PostgreSQL connection, the “spatially enabled” functionality is provided by the PostGIS package developed by Refractions Research Inc. (PostGIS is released under the GNU General Public License.)

PostGIS allows advanced topological constructs (coverages, surfaces, networks) to be stored, retrieved, and edited in the PostgreSQL object-relational database system. PostGIS works with purely geometric objects, as

well. The names of these objects are: point, line, polygon, multi-point, multiline, multipolygon, and geometrycollections. Our experimentation with PostGIS 0.8.1 under PostgreSQL 7.4.2 in the Red Hat Linux environment was successful, though some announced PostGIS capabilities are still under development. (Note to advanced users: PostGIS 0.8.2 Release has an “Optional Experimental Lightweight Geometry (LWGEOM)” feature that includes four-dimensional geometry.)

GIS ON THE WEB: MAPSERVER

Once a person has developed a GIS application, the next logical

step is to make it Internet-available or, even better, web-interactive. A set of proprietary products can help achieve this goal, but it is also possible to do it using strictly open source tools. The number of websites that have accomplished this goal is growing rapidly and most are running on Linux/Apache platforms. Most systems (as Red Hat distribution) already have an Apache server preinstalled. If not, the Apache installation is quite straightforward.

Based on our experience in the ITS Social Science, Statistics & Mapping Group using MapServer, however, the rest of the work required to get an application

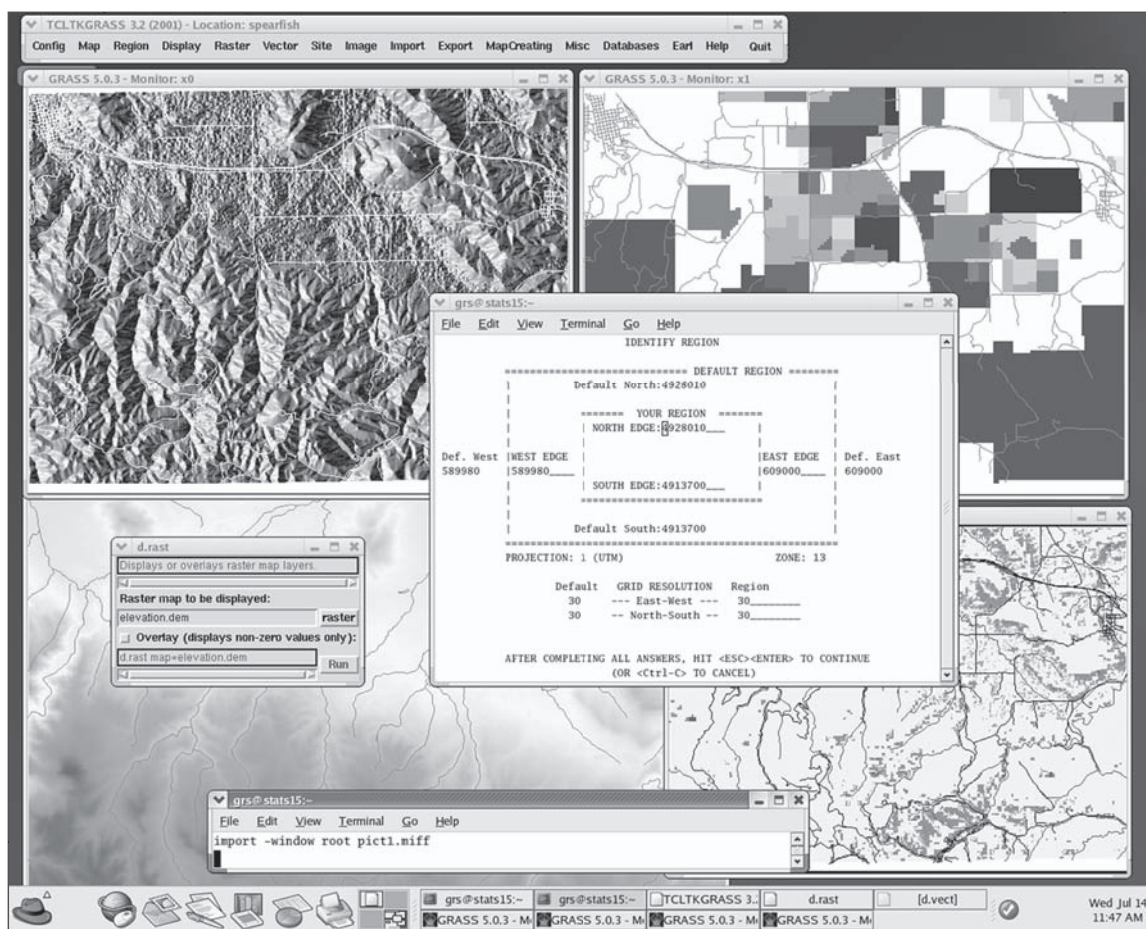


Figure 1. This GRASS GIS session displays different raster and vector layers from the Spearfish Database. The Tcl/Tk GUI is used simultaneously with the command line interface.

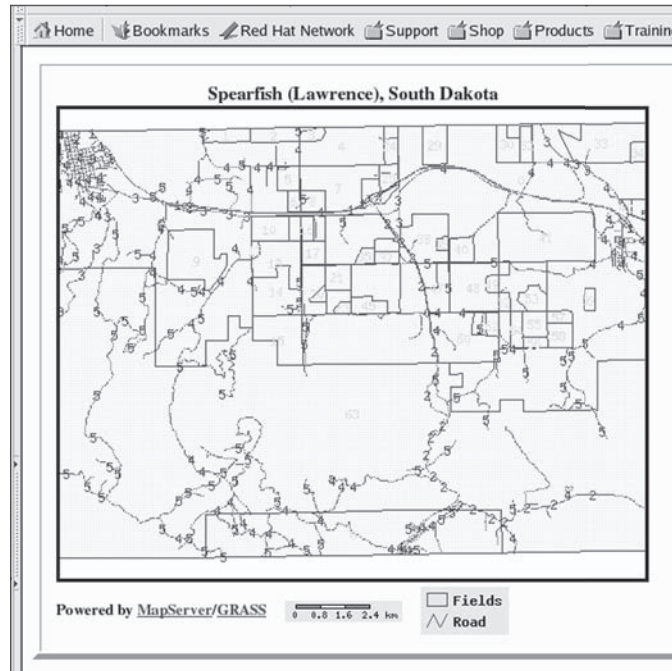


Figure 2. GRASS GIS data can be accessed through the Web using the MapServer package.

online can be a bit more complicated. MapServer is an open source development environment for building spatially enabled Internet applications. It was originally developed by the University of Minnesota (UMN) in cooperation with the TerraSIP project, a NASA-sponsored cooperative effort of UMN and a consortium of land management interests.

Before compiling and installing MapServer, we installed a few supporting packages:

- gd: A graphics library for fast image creation.
- FreeType: A TrueType font engine available from the FreeType Project.
- libTIFF: For TIFF/GeoTIFF support.
- Proj.4: A cartographic projection library from the US Geological Survey (USGS).

- libJPEG: From the International JPEG Group.

The MapServer site, <http://mapserver.gis.umn.edu/>, offers an abundance of information regarding installation, usage, and programming in a MapServer environment. Our particular interest was directed at the PHP/Mapscript module, which provides scripting access to the MapServer C API using a simple yet powerful PHP language. PHP can be configured on Red Hat Linux as a “Server API” in two ways: as an Apache module, or as a CGI. Due to some complicated PHP/Mapscript inner coding problems, this module will not work if PHP is configured as an Apache module... a piece of knowledge which cost us several days of work to learn.

NEW DEVELOPMENTS IN OPEN SOURCE GIS: THUBAN, QGIS, AND GRASS SERVER

Two recently developed open source tools that can serve as geo-

graphic data viewers for GRASS GIS may prove to be helpful. The first is Thuban—an interactive geographic data viewer (Intevation GmbH, GNU General Public Licence). Binding the GDAL library, Thuban will support the GRASS raster file format.

The second development is QGIS. It offers support for common vector and raster formats and already has GRASS GIS among its plug-ins. The process of making QGIS work with GRASS is straightforward. Before starting QGIS, you must export two GRASS shell local environment variables, GISBASE and LD_LIBRARY_PATH. GISBASE provides the full path to the directory where GRASS is installed and LD_LIBRARY_PATH is the path to the GRASS libraries. QGIS is actually based on Qt, a multiplatform C++ GUI toolkit. QGIS works with PostgreSQL DBMS through the PostGIS package once GEOS library has been installed. Fortunately, installing

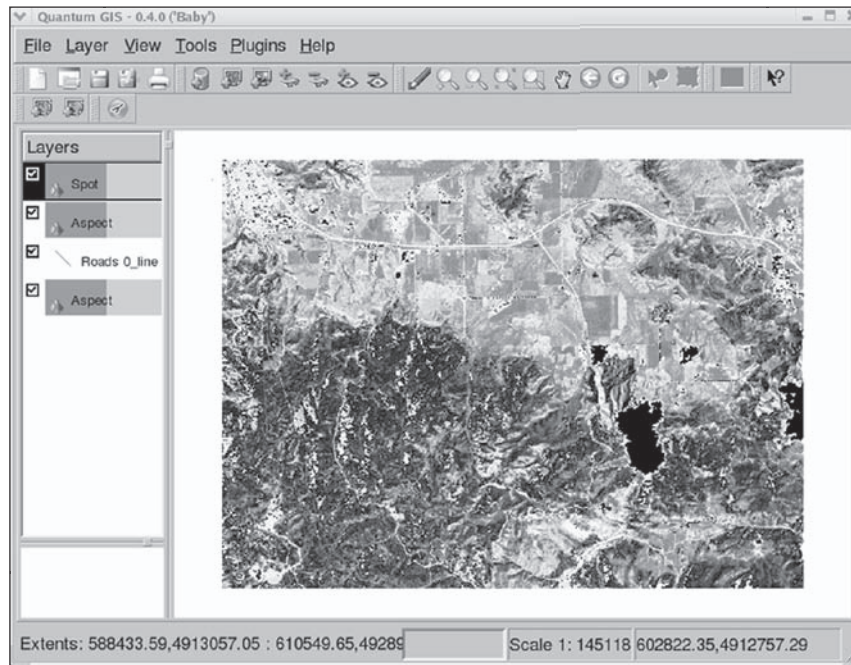


Figure 3. Quantum GIS (QGIS) can display GRASS data using its plugin.

GEOS (Geometry Engine—Open Source) is not complicated. (QGIS is released under the GNU General Public License.)

And finally, in recent news, the “FLOSS/GRASS Users Conference” was held in Bangkok, Thailand on September 12-14, 2004. (FLOSS stands for Free/Libre Open Source Software). At this conference, Radim Blazek and Luca Nardelli

(both from ITC, Trento, Italy) described their current efforts to build a GRASS server on top of GRASS as a client-server system. The challenge they are facing is very serious because the original GRASS design was oriented for stand-alone modules and is not thread safe. If they succeed in their goal, the use of GRASS on the Web will be greatly simplified.

For more information on open source GIS development, I recommend *Open Source GIS: GRASS GIS Approach*, by Markus Neteler and Helena Mitsova, Kluwer Academic Publishers.

Yakov Smotrisky is a Technical System Analyst in the ITS ACS Social Science, Statistics & Mapping Group.

What is Open Source Software?

The term open source refers to any freely-distributed software whose source code is intentionally made available for people to examine and modify. Popular examples include Linux and Mozilla.

The Open Source Initiative (OSI) is a “non-profit corporation dedicated to managing and promoting the Open Source Definition for the good of the community.” As described on their website, “The basic idea behind open source is very simple: When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing.”

For more information about open source, please visit <http://opensource.org/>.