Spatial on a Shoestring

Leveraging Free Open-Source Software

Jonathan W. Lowe

This column covers the role of emerging technologies in the exchange of spatial information.

If you were an extra-terrestrial anthropologist beamed to Earth for reconnaissance, how would you identify our culture's core values? Fortunately, as a preeminent cosmic researcher, you would know that members of all cultures symbolize their most treasured intangible values as real physical objects that they keep close to their bodies at all times.

Most adults from all walks of life wear wristwatches and carry wallets, for instance. Although clocks and cash are so familiar we hardly notice them, an alien might infer that we worship time and money. Not a bad deduction, really. How do you know if something is valuable? It costs a lot of money. How do you know if something is of high quality? It took a long time to create. Corporations live by the phrase "Time is money." For better or worse, we are a society of consumers.

With this observation in mind, an alien anthropologist might be confused to discover the Free Software Foundation (www.fsf.org). What's the explanation for an organization, according to their Web site, "dedicated to promoting computer users' right to use, study, copy, modify, and redistribute computer programs"? Why invest time but not take any money for the resulting products? Even more confusing to aliens — and free software's promoters — is why free software doesn't dominate the marketplace. If it took time to build (that is, high quality), but costs nothing, that ought to make it doubly attractive to consumers, right? The answer is market-dependent; some free software does dominate its market, but much does not.

For a success story, consider the free, open-source Apache HTTP Web server (www.apache.org). According to the Netcraft Web Server Survey (www.netcraft.com/survey/), Apache has been the most popular Web server on the Internet since April 1996. Netcraft's March 2002 survey found that 54 percent of the approximately 37 million Web sites on the Internet are using Apache, thus making it more widely used than all other Web servers combined. Microsoft's proprietary Internet Information Server holds second place with 31 percent of the market.

Get what you pay for? In the spatial industry, however, open-source freeware has a barely measurable market share compared with the proprietary products sold by Autodesk (www.autodesk.com), ESRI (www.esri.com), Intergraph (www.intergraph.com), MapInfo (www.mapinfo.com), and other commercial geospatial vendors. Part of the problem may be that free software sounds too good to be true. Its absence of any price raises the concern that "you get what you pay for."

Unable to find any extra-terrestrial consumer reports that transcended my own narrow cultural biases toward time and money, I decided to test the latest spatial database and Internet map server freeware on my own system, firsthand. In that five-day process, I learned about several strong spatial software alternatives, interviewed their champions about successful applications in the real world, and saw my (cultural?) misconceptions about freeware transformed into respect and admiration.

Antacid for map server heartburn

Relief might be an even better word than respect or admiration. During the past three years, most commercial Internet map server products have become more complex. The justification is that the new products are more scalable, secure, robust, and full-featured than their forebears. Unfortunately, they are also more difficult to install, more expensive, harder to maintain, and slower to serve maps. Considering that the majority of Internet mapping sites serve a small, regional audience generating tens rather than millions of visits per hour, few users really benefit from increased scalability anyway. How many users would instead be delighted with simple zoom, pan, and identify functionality and fast-

Net Results columnist Jonathan W. Lowe is the owner of Local Knowledge Consulting (Berkeley, California), where he designs and implements spatial Web sites. Lowe can be contacted at info@giswebsite.com.

Glossary

CGI: Common gateway interface

DND: Department of Neighborhood Development

SQL: Structured query language

SVG: Scalable vector graphics

XML: Extensible markup language

drawing maps in a pure HTML (non-JavaScript) Web page?

A simple plan. It is exactly this level of clean simplicity that the University of Minnesota's free, open-source MapServer (http://mapserver.gis.umn.edu) promises and delivers. MapServer is one of the most popular of a handful of free Internet map servers, all listed at The FreeGIS Project's (www.freegis.org) Web site, an informative clearinghouse for free spatial software, geodata, and collaborative spatial projects. MapServer's developers are clear about their product's purpose: "MapServer is not a full-featured GIS system, nor does it aspire to be. It does, however, provide enough core functionality to support a wide variety of Web applications."

MapServer users create interactive mapping applications with a time-tested development interface — namely, a text editor for authoring two text files. One plain text file describes which data layers to draw, how to label and symbolize them, their scale dependencies, and so on, using a nested outline resembling XML's structure (but syntactically more basic).

Another contains a static HTML file for the Web page within which the map appears. By substituting placeholders like "[map]" or "[scalebar]" at a few key points within the HTML, the otherwise static page becomes interactive to a user's zoom, pan, and identify requests. MapServer looks for the placeholders when responding to user requests and, on-the-fly, rewrites the contents of the brackets with current references to map and scalebar images it has just generated.

Within an hour of installing the software, I had my own Web site up and running based on the tutorial that accompanies the MapServer download. The development environment couldn't be simpler, and neither could the installation, though the compilation is not a trivial task.

True to the open-source tradition, MapServer relies on numerous other free software to convert vector to raster, draw true-type fonts, or create PNG images. If those programs don't already exist on your system, they must first be downloaded, compiled, and installed. The nice aspect of this process is that the user need only compile the features necessary for the project. The difficult aspect, particularly on Linux and Unix platforms, is that the compilation process enters the realm of the experienced programmer.

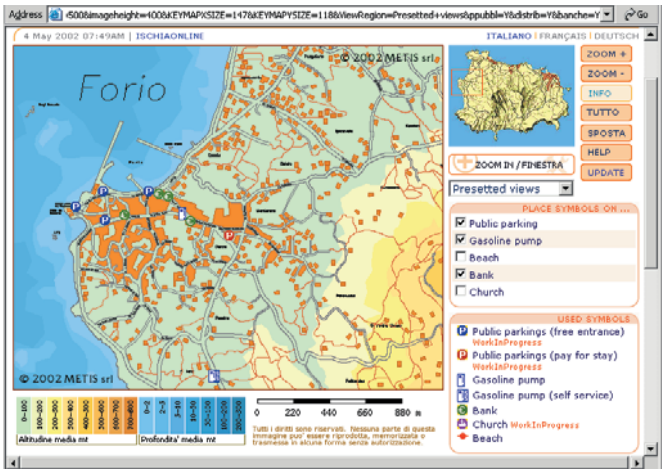

FIGURE 1 The MapServer application for Ischia Island, Italy, uses clean cartography and good layout to make a strong impact.

TIFFing it out. Compiling source code and installing these programs generates what programmers call *libraries* that other programs can refer to when running related processes. For example, MapServer developers didn't have to write their own code to draw TIFF images; another open-source program called libTIFF already exists to do just that task. When a MapServer user includes a TIFF image in his application, MapServer searches for the necessary logic dynamically in the libTIFF library. Keeping track of all the supporting programs and making sure that MapServer can find them on the system demands a modicum of familiarity with file decompression tools, system environment variables, and compilation methodologies. New users who dislike reading technically dense ReadMe files will suffer.

Once through this one-time compilation process, however, all that is required to install MapServer is to put a copy of its executable file (called *mapserv*) in the CGI-BIN directory of your (you guessed it) Apache HTTP Web server. Again, free software developers worth their salt take advantage of good free software. At this point, your system is ready to serve interactive Internet maps from a variety of source data file formats or spatial databases.

Who's using it?

A visit to the MapServer gallery of sample projects supports the claim of "wide variety" not only in application choice, but in extensibility of the basic product. But before covering extensibility, a confession is in order. Thanks to my time-equal-money prejudice against open-source powered applications, I visited sites in MapServer's list expecting to see crude vector graphics and primitive user interfaces. Sometimes, this was the case, but more often the interfaces were so simple and the cartography so beautiful, that I got lost in the data itself and just enjoyed exploring the maps. Any limitations clearly stem from the underlying data or the site designer's creativity, not the MapServer technology.

For instance, the Web mapping application for Ischia Island, Italy (www.ischiamappe.it/), built by METIS (www-users.cs.umn.edu/~karypis/metis/), combines smooth cartography and well-detailed supporting information (see Figure 1). Even though Ischia's map contains only a dozen layers, the clean interface and readable map seem more rich than other sites with more layers.

Not surprisingly, the site combining cartographic sophistication with a high degree of data detail and easy searching tools was Swiss — specifically, the City of Waedenswil, Switzerland, site (http://www.mapserver.ch/waedenswil/index_e.phtml) tidily built by TYDAC, Inc (www.tydac.ch) (see Figure 2). For a flavor of what can be done with engineering data, the simple but speedy Telecoms Line Plant Map built by GIS Solutions

FIGURE 2 The MapServer application for the city of Waedenswil, Switzerland offers a variety of search paths without confusion or clutter.

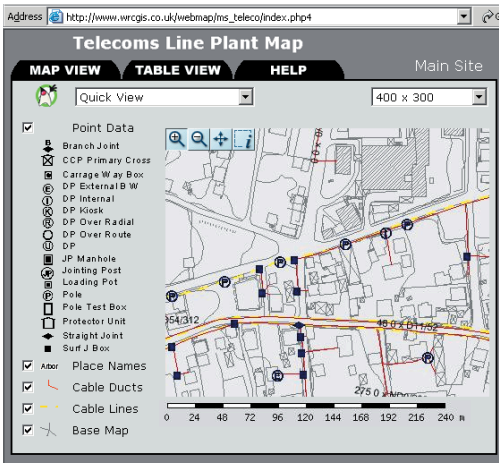

FIGURE 3 Built by GIS Solutions Group, this zippy MapServer application supports engineering symbology and CAD data with an option for HTML or Java interfaces.

Group (www.wrcgis.co.uk/Webmap/ms_teleco/) overlays engineering symbols on what appears to be a CAD basemap to the building footprint level (see Figure 3). This example extends the basic functionality of MapScript with PHP/MapScript and Dynamic HTML.

Other sample sites combine MapServer with JavaScript, Java, or Perl, for instance. Without extension, however, a MapServer site will run on any Web browser that supports Common Gateway Interface (almost all do). MapServer requires no plug-ins, and can be integrated into any existing site with minimal disruption to the existing page structure.

Same side of the coin

Architecturally, MapServer offers its users a very simple set of functions but supports expansion using popular client or server tools, free or commercial. This approach is exactly what David Coggeshall of San Francisco Communications (http://maplab.org) advocates as well. Unlike the MapServer sites, which use the Web server computer to process user requests, Coggeshall's demonstration site takes the opposite approach. The server simply stores the JPEG images and sends a complete set of code to the user's browser in the form of JavaScript and SVG. The receiving Web browser draws the SVG on top of the JPEG image (see Figure 4). The JavaScript turns the Web browser into a mini-GIS viewer. For instance, opening the HTML source of one page reveals JavaScript functions for zooming, panning, and identification of features. Opening the SVG source reveals the names and coordinates for San Francisco's entire street network.

The strategy requires recent versions of Web browsers capable of digesting SVG and interpreting JavaScript (now a default in Internet Explorer and Netscape). This client-side code exposure enables a user willing to learn JavaScript and SVG to copy and modify Coggeshall's application without compiling or installing software. A true open-source champion, he hopes they will do just that.

All you need is love?

Several artists, including John Lennon, have covered Berry Gordy Jr. and Janie Bradford's 1959 song "Money (That's What I Want)." Sooner or later, every open-source programmer will have to face up to the message in Gordy and Bradford's lyrics, particularly these: "Your love gives me a thrill; But your love won't pay my bills; I want money (that's what I want)." Programming is a joy, but the empty stomach growls. How do open-source developers survive?

Both MapServer and San Francisco Communications' development efforts were initially directed by experienced leaders relying on students and/or academic grants, as continues to be the case for San Francisco Communications. MapServer's development, however, has since sprouted several branches, illustrating the unpredictably inclusive, but distributed, process of open-source product evolution. National Aeronautics and Space Administration funding allows University of Minnesota students to write MapServer documentation. DM Solutions (www.dmsolutions.on.ca) has added OpenGIS Web Map Services compliance, format support (Interactive Graphics Design Software, MapInfo, Arc Coverages), and other improvements through contracts with Canadian federal government agencies. A lone programmer, Roderigo Cabral, connected MapServer to the Oracle (www.oracle.com) Spatial format with funding from a Brazilian government agency. MapServer's original author, Steve Lime, contracts for the Minnesota Department of Natural Resources while also actively supporting the core MapServer product. And these are just a sample of the many active participants.

Consulting for dollars. Modular extensions of a core product are sometimes the work of coordinated teams exerting consistent multimonth efforts. Paul Ramsey of Refractions Research, Inc. (www.refractions.net), engages in just this level of serious development on the PostGIS open-source effort. Like ESRI's ArcSDE or Oracle's Spatial extension, PostGIS spatially enables an open-source database called PostgreSQL for use as a backend spatial repository.

Building such complex functionality as topology into a mature database server like PostgreSQL is definitely a team project not for the faint of heart. Ramsey's team relies on services to offset the costs of development. The PostGIS group consists of six to ten local consultants all working to the rhythms of the British Columbia provincial government, Refractions' primary client. Each year, from April to mid-May, business is predictably

slow due to the government's schedule. Ramsey explains, "We generally reserve that time to do professional development: learn new tools, write demos, et cetera."

Ramsey is optimistic about the ongoing success of his development model: "Funding development out of consulting revenues can be a viable open-source business model. Our credibility as experts is backstopped by our development work. Other companies can try and be 'free riders' on our work, but they can never marshal the kind of technical credibility that we have without themselves contributing to the project. So either they contribute, or they compete at a disadvantage. I suppose the key is to not overspend on R&D and chew up all the consulting revenues, nor underspend and lose product visibility."

Optimism about the services model seems to be justified. In the past year the Refractions team's month and a half of free open-source work has generated several paying customers. Clients pay not for software, but for Refractions' consulting services to implement systems built with PostgreSQL, PostGIS, and MapServer for municipal and state data dissemination. Projects Ramsey described as having "fairly large datasets" are the state of Queensland parcel and permitting application, the British Columbia Digital Road Atlas, and the City of Boston properties database.

Like Ramsey, Coggeshall also hopes to turn the open source approach into grant funded service contracts, and San Francisco Communications' MapLab is already providing free assistance to several public safety agencies.

Open-source meets reality

For those browsing on a fast network connection, when visiting Boston's DND Property Inventory Web site (www.cityofboston.gov/dnd/M_Property_Inventory_Intro_Page.asp) be prepared for a blistering 2-second (or faster) map refresh speed (see Figure 5). This response time is all the more startling considering that the server is one of the city's slowest — a 500-Mhz, single-processor, 500-MB machine. The typical commercially served Internet map site pans and zooms in 5 to 7 seconds and relies on more powerful hardware.

Even so, developers have a reputation for being more excited than their users by fast performance. What really attracted Boston to an open-source system? Regina Obe from the Boston's DND explains the department's choice of MapServer with three main reasons:

- It was a breeze to setup — I, with little knowledge of GIS at that time, was able to get it set up and working in a week with very little help except from the mapserver listserv group.
- It had a low foot-print and was surprisingly fast. I could load the city's building footprints, parcel footprints in a shape file (or mapinfo file) and it would run decently on a regular server (1 processor, 500-MB onboard memory), and of course,
- The price was right.

Obe's mention of "help from the mapserver listserv" parallels my own experience when installing a MapServer add-on called msWorkBench, a timesaving graphical user interface for creating the MapServer text files. The compilation worked, but some of the programs wouldn't run. A few hours later, I was exchanging e-mail with someone much better versed in the programs and very helpful — especially considering that his support was entirely voluntary.

As for the DND'S choice of PostgreSQL and PostGIS for their database, Obe says, "We chose PostGIS because we wanted to do more sophisticated things with our data [than SQL Server supported] — (currently showing proximities, internally display the status of projects we are doing, later expenditures by geographic areas, et cetera) and the price of SDE was just out of the question, and sharing the main SDE server with city hall was just unfeasible [for update and testing purposes]."

Furthermore, Obe no longer has to worry about a software budget when scaling her system to include multiple copies of the spatial database or MapServer internally and externally. Departments won't have to fight over who will house the server or who will fund the project. As with most successful projects, there are already future plans for expansion and consolidation. Obe knows her system well, which gives her the confidence to extend her success beyond her department toward the city's greater good.

FIGURE 4 San Francisco Communications' clientside Internet mapping solution relies on JPEGs, JavaScript, and SVG.

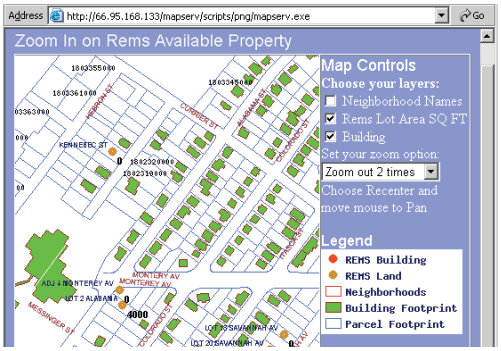

FIGURE 5 Using free, open-source software, Boston's DND site is a fast mapper despite its modest server hardware.

A liberating experience

After a decade of experience with proprietary commercial spatial software, my long-overdue meeting with the open-source community was a welcome one. For the first time in years, instead of struggling to master a software package, I was solving spatial problems with a software package. I felt like a member of a larger community with a strangely personal sense of freedom. Maybe the best things in life really are free!