Spatial Decoupled Architectures

Holding Back the Tiers

Jonathan W. Lowe

Feel like you're wearing more than one hat in your job as a geospatial professional? It's a big club. Follow the trail to multitier architectures — with spatial functions on each tier — for one answer.

As geospatial professionals, we lead double lives. Our technical spatial skills are a profession unto themselves, but we typically apply those spatial skills within another domain — forestry, real estate, utilities, public safety — any market benefiting from spatial knowledge. In forestry, for example, some call themselves "foresters who use GIS," others "spatial experts in forestry." For those who look in the mirror and see a spatial expert, do you also see an IT professional staring back at you? Even though we geospatial professionals ply our trade almost exclusively on computers, many of us know only a bare minimum about databases, application servers, hardware, networking, or object-oriented programming. On business-oriented projects in particular, pure spatial expertise alone is becoming insufficient.

Why? Since the late 1990s, corporate computing has shifted from a two-tier (client–server) to a multitier enterprise application architecture in which it's possible to deploy spatial processes on any or all tiers using the languages and software specific to each. The rise of the Web and consequent need for browser-based application interfaces drove this shift. Vendors with all their business logic buried in rich clients had to rewrite that business logic to another (middle) tier when creating a Web interface. This column explains multitier architectures, shows where spatial functionality has infiltrated each tier, and explains the consequences of this trend on the design of business-oriented enterprise spatial applications and on the résumés of geospatial industry professionals. If you're new to the concept of multitier architectures, prepare to be the most confused by the middle tier, often called the application server. The following analogy and explanation attempts to limit confusion by sticking with only three tiers, though some implementations use more. As in life, we'll start at the bottom and work our way up.

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

Bottoming out. Fresh out of college in 1987, I landed a job in Maine as a logistician at the Hurricane Island Outward Bound School, an organization dedicated to building character through outdoor experiences. A glamorous job it wasn't. An Outward Bound logistician's main responsibility in life is to assemble collections of gear for sailing trips, kayaking trips, dogsledding trips, and so on, at the bidding of trip leaders who announce, for instance, "Hey, logistician! My group of 12 sets sail on Wednesday for our 21-day expedition." Given these parameters — group size, activity, start date, and duration — I would pull exactly the right numbers of sleeping bags, stoves, bags of gorp, and dozens of other food and gear items from the warehouse shelves, and stack them in the trip leader's locker. Though I didn't realize it then, I was the bottom layer of a three-tier architecture.

My gear and food warehouse was equivalent to a database of raw materials, the leader's request was a SQL statement, and I was the transaction manager — a human database search engine. Extending this analogy to a typical enterprise, it's likely that there are numerous data sources — databases, spatial and non-spatial files, and legacy mainframes — all in the bottom tier, with relational databases predominating. What lumps all these sources together is that they store persistent data, the crown jewels of most businesses.

Spatial for DBAs. The news to many spatial experts and DBAs alike is that most leading databases are also standalone spatial databases. IBM's (www.ibm.com) DB2 and Informix databases,

Glossary

DBA: Database administrator

I/O: Input/output

IT: Information technology

GUI: Graphical user interface

LBS: Location-based services

PHP: Preprocessor Hypertext Protocol

SQL: Structured query language

Oracle (www.oracle.com), and PostgreSQL (www.postgresql.org) all store and manipulate not just standard data types — text, numbers, and dates — but also spatial data types. Exclusive of any third party GIS vendor, these databases let users load shapefiles into database tables, then interrogate the tables with spatial SQL.

The advent of spatially extended databases excites DBAs (more so than spatial experts) because without having to learn any new languages or program interfaces, a DBA can now include spatial criteria in her SQL queries. For instance, in a spatial database containing traffic accident and street data, the DBA could return a list of accidents within 300 feet of a selected block using the following SQL:

```
SELECT accidents.id, accidents.type, accidents.date
FROM accidents, streets
WHERE streets.block = “1200 Main Street”
AND DISTANCE(streets.geometry, accidents.geometry) <= 300;
```

To those familiar with SQL, the first two lines should be familiar — we're listing fields in an accidents table by ID, type, and date. The third and fourth lines contain filtering criteria, including a spatial function, DISTANCE, that compares two geometries and returns the distance between them. So, this query isolates the records in the streets table to a single row (where the block value is “1200 Main Street”), and compares that block's geometry (a line) to all the accident points, returning only accidents within 300 feet. The DISTANCE function is just one of many functions, such as INTERSECTS, TOUCHES, WITHIN, and NEAREST, that analyze geometry just as desktop GIS applications do.

What's so special? DBAs are happy, but traditional GIS professionals may be nonplussed; why reinvent spatial processing in databases when it's already part of every GIS vendor's desktop product? One reason is for performance. When a database processes a spatial query, the database engine operates on the data locally or “in place” rather than extracting the data elsewhere (for example, into the client application) for processing. Especially when very large datasets are involved, such local processing can deliver the fastest possible performance by eliminating I/O time between the database and an external application. Also, most desktop mapping tools draw all the datasets graphically, before and after processing. If the user just wants a tabular list, graphic rendering is a waste of processing power and time.

In addition to SQL, DBAs also manage data with stored procedures — a proprietary extension available in all spatial databases allowing users to write their own functions or to have one action trigger others. For instance, a trigger could automatically enforce this rule: “When a county boundary is realigned, also update any corresponding municipal boundaries, even if the county and municipal polygons are in different tables.” In *Patterns of Enterprise Architecture*, however, Martin Fowler points out, “There's usually a fair bit of debate over stored procedures. They're often the fastest way to do things since they run in the same process as your database and thus reduce the laggardly remote calls. However, most stored procedure environments don't give you good structuring mechanisms for your stored procedures, and stored procedures will lock you into a particular database vendor.” On the other hand, Fowler notes Oracle's recent support for running Java applications inside the database's processes is an improvement over proprietary stored procedures.

> DBAs are happy, but traditional GIS professionals may be nonplussed; why reinvent spatial processing in databases when it's already part of every GIS vendor's desktop product? One reason is performance.

Stored procedures are one thorn in a prickly debate over multitier architecture. Storing and quickly accessing data has always been the realm of databases. Managing that data's behavior, also known as the business logic of an organization's workflow, is a different story. For instance, GIS and IT vendors choose different tiers to house the same business logic. Oracle's support for topology in the database conflicts with ESRI's (www.esri.com) published research — *Building a Robust Relational Implementation of Topology* — which explains how nonpersistent topology calculated on the fly returns the fastest benchmarks, particularly for versioned editing.

ESRI's CTO, Scott Morehouse, recognizes the value of SQL for modeling short transactions in tightly-coupled systems, such as ATM withdrawals. However, notes Morehouse, “Business logic for applications such as mobile computing performs better in the middle tier than in databases, which were not designed to handle long or disconnected transactions and loosely-coupled systems.” Buts what makes the middle tier a better or worse choice as the home of business logic?

Manic middleman.

Logistics at Outward Bound had no future, so I became a trip leader, unwittingly climbing from the bottom to the middle tier, and morphing into a human application server. Trip leaders at Outward Bound have to be excellent communicators, synthesizers, and delegators. Their job is to assemble and deliver a useable product -— namely, a multi-day wilderness experience — to their customers: a clueless group of students. Leaders don't just requisition raw materials from logisticians; they also combine those supplies intelligently: rehydrating powdered milk, putting the different fuels in stoves and lanterns, packing the life vests accessibly, and countless other tasks. In short, they use their training (business logic) to organize a collection of gear and food (raw data) so their students (the users) can draw value from it. And when several students need help at the same time, leaders delegate (load balance) to their assistants.

Trip leaders are analogous to application servers because both integrate a variety of data, messages, and rules into a meaningful result. To be clear: application servers are software that run on the middle tier; they are not the tier itself. And logical tiers are not necessarily physically separate, though often databases reside on different machines than application servers do.

Just passing through. Application servers can be confusing because they are neither raw data nor what we recognize from client–server days as an application. In fact, application servers are actually a growing collection of multiple subprograms. Marketers are eager to taut application server features such as Java containers, connection pooling, messaging, scheduling, transaction management, clustering, personalization, JSP, Web services, systems management, security, communication services and others. For commercial application servers — such as BEA's (www.bea.com) WebLogic, IBM's WebSphere, or Oracle's 9*i* Application Server, this range of integrated features sells for between $5,000 and $80,000 depending on the package and list of functions. The leading open-source application server, JBOSS (www.jboss.org), is free.

The idea is that most businesses need a majority of the above-listed capabilities working in concert to operate in today's electronic commerce environment. So whether the business involves online flea markets, getting a date, or running a utility company, there are sure to be issues of security, availability, scalability, and others — all supposedly already on the application server's functionality menu. Businesses that buy application servers hope to save development time by plugging their specific business logic into a preexisting framework.

For an application server to process spatial data, it must pull records from the bottom tier, analyze them, and pass the results to the upper tier. ESRI's ArcSDE is an application server of sorts, enabling even nonspatial databases such as Microsoft's (www.microsoft.com) SQL Server to impersonate true spatial databases. SQL Server can't query for DISTANCE, INTERSECTS, TOUCHES, or any geometry-aware operators, but ArcSDE overcomes this problem by storing geometry in SQL Server, while interpreting and processing that geometry in the middle tier. In this regard, ArcSDE is like a multilingual author filing English-language books in a Japanese library and translating them on demand for Japanese readers who don't speak English. ArcSDE follows a similar strategy for true spatial databases, but delegates a subset of spatial processing to the databases themselves. Other vendors, such as Laser Scan (www.laser-scan.com) and e-Spatial (www.e-spatial.co.nz), bring IT-centric solutions to the enterprise by relying even more heavily on native spatial database functionality, but exclusively within Oracle databases.

In this regard, ArcSDE is like a multilingual author filing English-language books in a Japanese library and translating them on demand for Japanese readers who don't speak English.

"Databases go to college." For an example of a mainstream IT application server in action, consider what happens in the admissions department of mythical Local Community College, good old LCC, during enrollment of a new student, Dion Doe. The physical evidence of Dion's enrollment is a "welcome packet" with a congratulatory note and various brochures about LCC. Different students get different brochures based on the efforts of LCC's application server, whose work begins with a security check: Does the user entering Dion's name have clearance to enroll new students? Another step might involve extra-curricular perks. Check the bottom tier: Is Dion male or female? The mens' and womens' gyms have different locker use policies and therefore different brochures. There are money matters as well: Will Dion be a full- or part-time student? Dion's status influences the application server's query to LCC's financial database (another bottom-tier dweller) when generating an accurate bill for Dion's first semester.

These different requests and responses may be happening simultaneously. Part of the application server's job is balancing that multi-processing load. The requests may also include spatial processing and external systems. For instance, the application server gets Dion's text address (again from the bottom tier) and sends it to an external geocoding Web service — not controlled by LCC — which returns the address's geographic point as a latitude and longitude. The application server then queries an internal LCC spatial database to see within which public transit region (if any) Dion's address falls. Depending on the result of this point-in-polygon spatial function, the correct bus map and schedule becomes part of Dion's welcome packet. Each small step is part of an overall workflow. LCC's developers used components from their application server software package and strung them together to satisfy LCC's business logic regarding enrollment and creation of welcome packets.

Although the transit regions come from a spatial database, LCC's application server might have used spatial database functionality, or maybe GIS vendor software, to accomplish the point-in-polygon analysis. The choice is one piece of a complicated enterprise architecture design that weighs cost, performance, existing staff skills, need for scalability, delivery schedules, and many other details specific to each application and enterprise. Examples from vendors in both spatial-centric and IT-centric camps confirmed that each case requires unique solutions.

GUI on top

The uppermost tier in a multitier system is for presenting information to the user. Variations in presentation mainly involve whether the client displaying all the buttons,widgets, and maps will be a desktop application or a generic Web browser.

Web browsers win the prize for ease of maintenance. Desktop applications win for responsiveness and broad functionality. In client–server architectures, the database was the bottom tier and the application was the upper tier, with no middle tier between them. In three-tier architectures, the application GUI is decoupled from the business logic and (spatial) processing code. This makes it easier to change a GUI without changing the logic below it. Many organizations whose products began as desktop applications now can also support a browser-based GUI thanks to the three-tier model.

So, traditional desktop GIS applications are changing, but not necessarily perceptibly. Typical users get few clues about what's going on beneath the surface of their GUIs. And that's okay. My Outward Bound students didn't much care how the loaves of bread and dry sleeping bags had arrived in their duffel bags; they just wanted to eat and snuggle off to sleep. From the beginning of our industry's history, most spatial users have had similar needs — demanding intuitive, performant user interfaces, not diagrams of the underlying plumbing.

IT and spatial users, however, are both increasingly sophisticated. For instance, some are Web page designers who want to include interactive maps in their Web sites. Others are DBAs who want to add spatial function buttons to their desktop query tools. Should the buttons on Web pages link directly to a bottom-tier database or use a middle-tier application server? Both are options. It's possible to wrap up all three tiers in the GUI itself, as some spatial experts do with the geographic capabilities of Macromedia Flash. HTML extensions written in PHP, Javascript, Java applets, and plug-ins like Flash make otherwise "thin" browser clients useful for specific tasks at a low price. For some projects, this is all that's needed, at least initially. Knowing whether and when a project's scope will creep is part of the designer's responsibility. The appearance of spatial data and functionality at all levels of enterprise architecture makes this job both more difficult and more interesting.

Melting into IT?

The appearance of spatial functionality at all levels of multitier enterprise architectures invites more people to the party. Three-tier architectures are changing the focus of mainstream IT and spatial vendors alike by opening doors to better performance, easier maintenance, and lower costs for applications involving (but not revolving around) spatial processing.

Three-tier architectures are changing the focus of mainstream IT and spatial vendors alike by opening doors to better performance, easier maintenance, and lower costs for applications involving (but not revolving around) spatial processing.

This shift is enlarging the pool of people who can work with spatial data, from GIS-only to IT folks in general. Spatially-enabled multitier architectures empower database administrators to answer SQL queries involving point-in-polygon analysis. Likewise, an application server programmer can now identify the nearest taxicab to the caller's location using pure Java. Given your ZIP code, a Web page designer can identify movie theaters within a 10-minute drive time using Web services and PHP. Traditional GIS vendors are aligning their products to work within IT-centric multitier architectures, but many IT professionals still have never even heard product names such as ESRI'ArcView, Integraph Mapping and Geospatial Solutions' (www.integraph.com/gis) GeoMedia, or Autodesk's (www.autodesk.com) MapGuide. In fact, their spatial applications may not need graphic maps at all.

Understanding multitier architectures and application server software helps make sense of GIS and database vendors new products (and marketing strategies). Regardless of the vendor, however, tools that draw the maps or present spatial results are increasingly decoupled from the software that analyzes the data. And the spatial data itself often belongs to yet another separate persistent tier. Reflecting this trend in their product line, for instance, ESRI has recently announced "ArcGIS Server," a spatial data analysis product with no graphic map at all. Oracle offers not only a spatial database, but also their own middle-tier application server, including a location-aware e-Business Suite in support of applications such as LBS.

Since the late 1990s we've been hearing about GIS melting into IT, accompanied by images of the GIS ice cube slowly disappearing in the mainstream IT glass of water. Are we there yet? Is the ice cube ever going to melt? I think the answer is that much of it already has, but also that some of it never will. Today's traditional desktop geospatial applications now support highly sophisticated editing tools and workflows (and patents!) that are unlikely to be subsumed by mainstream IT anytime soon, if ever. Their designs are the result of many years of vendors' and users' shared experiences. At the same time, database companies, IT-centric spatial integrators, and traditional GIS vendors continue to embed spatial functionality throughout the multitier stack for their business-oriented enterprise customers, not necessarily even labeling the projects as spatial. So go ahead — drink the water and chew the ice cube, too, if so inclined. Both are now possible at a tier near you. ⊕