GE Energy

Smallworld Core Spatial Technology 4 Presenting Spatial Data In Multiple Contexts – Using The Right Reference System For The Task In Hand

GER-4232 (07/04)

Presenting spatial data in multiple contexts – using the right reference system for the task in hand

Abstract

In a conventional Geospatial Information Systems (GIS) architecture, spatial data is stored as a single representation that essentially models the humble map. Each feature is referenced by a single coordinate system using a common unit system. For applications that require simple map-like functions such as quick navigation, simple spatial queries and rudimentary analyses, this is an adequate approach. However, in the enterprise GIS environment this is an inflexible solution. Many assets have more than one representation: perhaps a geographic one that specifies its location and an additional representation that models its intricate internal workings. This kind of asset is difficult to model using a conventional GIS architecture. Often the only solution is a piecemeal one using custom code that links the geographic data to external files or databases, which in turn frequently leads to maintenance problems and data integrity issues.

The advanced Smallworld spatial database from GE Energy has been designed from the beginning to support these often quite different representations of spatial data allowing enterprises the flexibility to model the intricate and complex assets they operate, often improving the efficiency of many business critical processes.

More than one map's worth

Most people are familiar with a map. It shows a pictorial representation of an area of geography. This area is confined by the physical size of the paper and is rendered in the context of a scale that defines the units used and the magnification.

This simple map model is the basis of many conventional GIS databases.

When creating a spatial database for the first time, the administrator needs to define the maximum extent of the data to be mapped, how accurately features will be located and how many different kinds of objects will be stored. This is a very important step in the initial configuration of a spatial database as it will affect the performance and capabilities of applications built on top of it.

There is an important relationship between the maximum geographic extent of a spatial database, the resolution of the data stored within it and the number of different kinds of objects to be stored. Generally speaking, the larger the area that needs to be stored, the lower the resolution of the spatial data has to be. For example, a spatial database that maps all of North America might only require features to be positioned to one centimetre accuracy, whereas a database that maps the city of New York might require spatial accuracy down to one millimetre.

These three factors all need to be considered carefully before populating a spatial database as they have a significant impact on the efficiency of commonly used GIS functionality such rendering, selection, spatial queries and so on.

Many enterprise applications require assets to be modelled in different ways, some of which spatially overlap each other, but not in any meaningful way. For example, drawings often have (0,0) as an origin and share a common coordinate system and units. In this example, multiple drawings would appear drawn on top of each other and be illegible to the user. A single view of the world, therefore, is potentially a significant restriction. It does not support the operations of major enterprises such as power distribution or telecommunications that require different representations of the same asset. For example:

- Field engineers will require a geographic map to help them locate assets for maintenance, whereas control engineers might be more interested in a schematic view of the same spatial data that emphasises connectivity.
- Some assets have a complex set of internal components that affect the operation of the network that they are connected to. Engineers often find it useful to see inside these assets to examine their configuration.
- Quite often many assets will have associated drawings or photographs that provide useful supplementary information that will aid an engineer.

These examples illustrate some of the problems with having a single view of spatial data:

- Geographic data is typically spread over great distances whereas associated drawings are laid out using paper-sized dimensions. This often leads to some sort of compromise between the best databases units to use and performance.
- The performance of a spatial index is also affected by the amount of data that has to be sifted through to obtain the correct result. Clearly, placing all kinds of disparate data together rather than segregating it into logical groups based on use results in a lot of unnecessary spatial data being processed. For example, there should be no need to consider the internals of an asset if all that is required is its geographic location.

For all these reasons using a single representation to store many fundamentally different types of spatial data is a poor strategy and one that is clearly not best suited for enterprise GIS applications.

A universal approach

The Smallworld platform's advanced spatial technology allows radically different kinds of spatial data to be stored in a single database, simplifying its administration and use. It has integrated support for the demarcation of spatial data that optimises performance and storage.

At the highest level, the Smallworld architecture categorises spatial data into what are called universes. Each universe contains one or more worlds. A world defines the maximum extent of the spatial data to be stored and the base units of each coordinate. Each world can be used to store different kinds of spatial data. A single world, for example, might store a single schematic diagram. Segregating spatial data this way improves performance of many spatial functions by allowing irrelevant data to be ignored.

When defining universes and worlds, it is important to think carefully about the spatial data to be stored. This is because each piece of spatial data is stored in the database using a spatial tag. These spatial tags are then organised into an index which provides much of the core functionality of a spatial database (map rendering, spatial queries and so on). It is vital, therefore, to optimise this spatial index for the spatial data being stored.

A bit part

When a Smallworld spatial database is first created, the administrator has to initially decide how many universes are required. For each universe the number of worlds has to be defined along with what is called its priority bits. The priority bits influence the control of layers of spatial data (allowing layers to be quickly turned off or on depending on the application). These three values (universe number, world number and priority bits) represent the first part of the spatial tag. The remaining bits are used for a pure spatial component that allows objects in close proximity to each other to be efficiently stored to and accessed from disc.

An example database might have eight universes: one for map data, another for drawings and a third for schematics. The remaining five universes are held in reserve to meet future requirements (this is a common and prudent practice).

The map universe will probably have a single world, but it will likely represent a large volume of expansive spatial data with many layers. In this case, most of the bits would be allocated to the priority and spatial components. This optimises the world toward efficiently being able to query large amounts of spatial data and also allows for many layers to be turned on or off.

A schematics universe, on the hand, would be configured differently. There is likely to be a large number of schematics to store requiring an equally large number of worlds. However, schematics often contain much less spatial data than the main map and have fewer layers. Consequently, more bits of the spatial tag would be assigned to the world number than the priority bits or the spatial component. This configuration would allow the storage of millions of schematic drawings.

Each world also has its own coordinate system that can be converted on-the-fly to a common application-wide coordinate system facilitating global analysis and display.

This flexibility permits a wide range of different types of spatial data to be efficiently stored and accessed. Enabling the Smallworld database to be tuned in this way provides customers with highly scalable, enterprise-wide solutions.

More than just spatial data

The powerful Smallworld database technology developed by GE, is complemented by an integrated and flexible application programming interface (API) that enables application architects to leverage this technology in a consistent way. This novel approach is based around the concept of what is called a spatial context.

A spatial context is a way for assets to publish, via a well defined API, a set of spatial representations of that asset. For example, an electricity substation might have a geographic spatial context (its location in the real world) and an internal spatial context (which shows the various discrete components it is made up of). Each spatial context definition includes a name and returns a set of geometries (points, lines, areas and so on) from a particular world. Spatial contexts provide a loosely coupled link between an asset and its spatial data. This flexible approach means that spatial contexts can be defined on assets that have no spatial data themselves but might, via a join, have a relationship with another asset that does.

A reusable spatial context viewer is provided that can be easily embedded into an application to provide much of the viewing functionality (panning, zooming and so on) with little effort by the programmer. Core support is also provided for displaying multiple viewers allowing multiple representations of the same spatial data to be easily compared (for example a geographic section of an electricity network and its schematic equivalent).
Navigation between spatial contexts is another important requirement. In-built functionality

allows users to select an asset in one representation and locate the same asset in another representation, quickly revealing important relationships.

A connected world

As well as being able to efficiently store, query and view data from multiple worlds, many enterprise GIS applications also require analysis of data that is distributed over more than one world. One example might be that of a substation that contains switches. Here a trace that was initiated in the geographic world will pass through the substation and into its internal world traversing the network of components it contains. The state of these components will determine whether the trace re-emerges into the geographic world (for example, an open switch might block its progress).

This kind of powerful analytical functionality is provided by a special spatial object called a hypernode. Each hypernode acts as a bridge linking spatial data in one world to spatial data in another.

Conclusion

Over fifteen years ago the architects of the Smallworld platform recognised that not all spatial data is the same. Designing the core of its spatial database from the beginning to model this view of the world allowed the storage, access and analysis of this spatial data to be highly optimised, increasing scalability and performance.

This flexible approach enables the advanced Smallworld spatial database technology to model the world's most complex and intricate networks.