GE Energy

Smallworld Core Spatial Technology™ 4

Spatial data is more than maps – tracking the topology of complex network models

GER-4230 (10/04)

Spatial data is more than maps – tracking the topology of complex network models

Abstract

Spatial data is much more than a pictorial representation of an enterprise's infrastructure. Most conventional Geospatial Information Systems (GIS) architectures allow spatial data to be stored and then retrieved at a later date for display or printing. More advanced architectures support spatially-based queries and analysis, but even these often fail to fully exploit the real value of spatial data: its complex interrelationships. These relationships can sometimes be inferred by close visual inspection of map features, but more often than not an invisible underworld exists that complements the geographic world by modeling the complex logical network that supports many of the day-to-day business processes that are critical to today's enterprise. Many conventional GIS architectures do not support this kind of modeling (or do so in a cumbersome way), making them unsuitable for managing the kind of complex network operations demanded by enterprise GIS.

The advanced Smallworld spatial database technology designed by GE Energy not only provides class leading spatial storage and analytical functionality, but also was designed from the beginning to support these kinds of complex network models allowing businesses to fully exploit their investment in their spatial data.

Spatial data is more than map deep

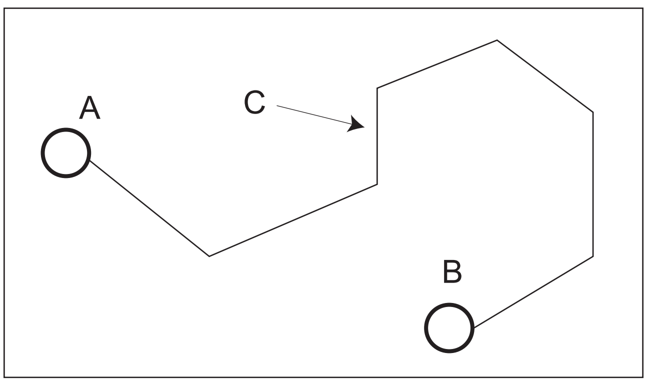
Being able to produce a map of spatial data is perhaps the most basic function of a GIS. Replacing paper maps with digital equivalents that are easier to store, distribute and maintain allowed GIS to deliver real business benefits. Storing spatial data in a digital form also enabled exciting new analytical and query functions. For example, businesses could now quickly locate assets, accurately produce inventories based on a geographic region, plan growth by spatially analyzing demographic data and so on. It would be easy to assume, therefore, that the key to understanding gis in the enterprise would be to understand the map. The map presents itself as an obvious way to represent spatial data: it is a familiar object that has been in use for hundreds of years. However, when the Smallworld platform architects looked at the map, they realized that it was not the focal point of spatial data that conventional thinking suggested. Rather, it was really a thin decorative veneer obscuring the real underlying value of spatial data: the network.

For utilities as diverse as water and telecommunications the network is really the heart of the business. The network is a complex set of connected assets many of which can affect the operation of the network by changing their state (for example, a valve can be closed or opened). This complexity is often compounded by the fact that many networks are in fact constructed from many virtual sub-networks in an effort to provide redundancy (for example, if a customer experiences an outage, the network will often reconfigure itself by supplying power via an alternative circuit). None of this underlying value is obvious from looking at a conventional map-based model of the world.

This revelation meant that many conventional GIS architectures had made a fundamental mistake: they had designed their GIS architectures to model the map, instead of modeling the network.

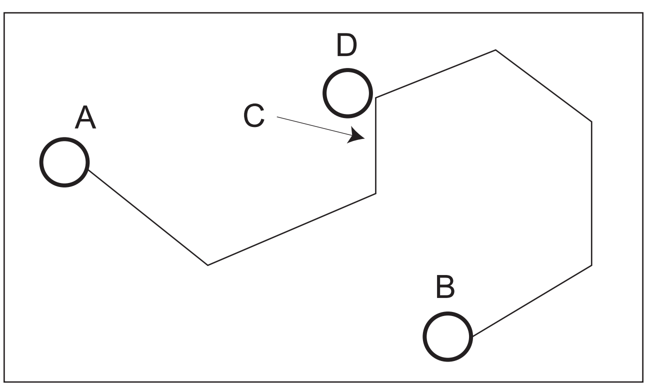
To address this fundamental problem, many GIS vendors attempted to network enable their map-based architectures by supplementing the geographical model with simple connectivity data based on conventional database techniques such as joins and lookup tables. This is an adequate architecture for very simple network applications, but this piecemeal approach becomes cumbersome to write applications against and is inefficient to store and manage. This approach effectively puts the implementation load on to the shoulders of the developer of a network application, rather than providing a reusable common library of functionality provided by the core architecture. This often leads to many simple network models and custom code tailored to a particular application that rapidly becomes expensive to maintain.

There are also numerous subtleties that be-come apparent when modeling networks that are difficult to support with a superficial connectivity representation. A trivial example is an electricity utility that might have two switches; A and B, that are connected with a cable, C. When modeling this simple network, the user would probably position A and B on the screen, then draw the appropriate route representing the cable connecting A and B. Internally at this point some network relationships would be implicitly created (A connects to B, A connects to C, B connects to C and so on).

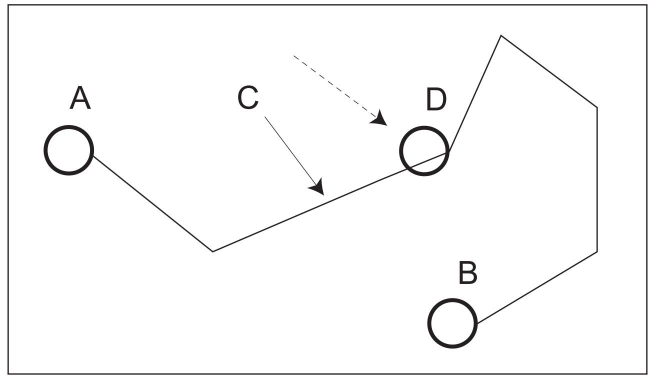


Suppose another asset, D, is now added that lies on the cable.

As before, the new asset is positioned and new connectivity information built up. Consider now that D was incorrectly positioned. Ideally, it would be helpful



in this circumstance if the connectivity model could drive the geographic model allowing the route of the cable to be automatically modified to keep D visually, as well as logically, connected.



This straightforward example reveals two additional limitations of the simple connectivity model. First, the simple network model is often too simple to support many of the important network business process demanded by enterprise gis applications. Second, the granularity of the simple network model is too coarse to accurately model the geometric representation of the network with which it has a very close association. The consequence of this is that in the example the cable would have to be manually edited so that it passes through D or some complex application code would be required to help automate the process.

The importance of being able to efficiently model a network extends well beyond its storage and management. Being able to accurately model the network means that it can also be accurately analyzed and that the results are worthwhile. Electricity utilities, for example, need to be able to perform complex calculations on sections of network to ensure peak loads will be within operational and safety margins. Water utilities faced with a burst pipe will quickly want to work out the extent of the outage and identify a set of valves to allow its isolation (not to mention to notify those customers affected). The cost of routine inspections can often be reduced by optimizing the order in which assets are visited to reduce travel times. Even trivial network questions such as 'what is connected to what?' frequently form the basis of many critical business processes.

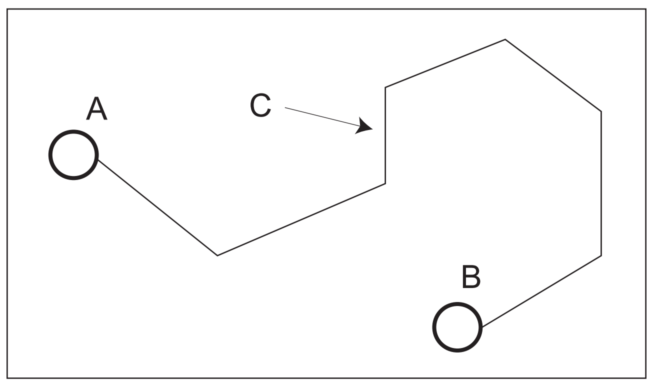
These kinds of requirements not only demand an efficient model, but also a set of powerful network analysis tools that are able to quickly yield answers. Perhaps the most important of these is an efficient network trace engine that is capable of quickly traversing the network model revealing important business relationships as it progresses. This is common network functionality, not a bespoke add on.

Many utilities have strict rules about how various assets in their network can interact with each other. These rules are often complex and are often derived from strict operating or safety standards (for example, a simplistic rule might prevent electricity cables from connecting to water mains causing a short circuit). These rules not only govern logical connections, but also control how the geographic representations of these assets interact with each other (for example, if two routes cross should they be allowed to intersect or simply overlap each other when inputted). These kinds of rules are often global to a network—a fact that makes building this kind of functionality using bespoke code in individual applications even less appealing.

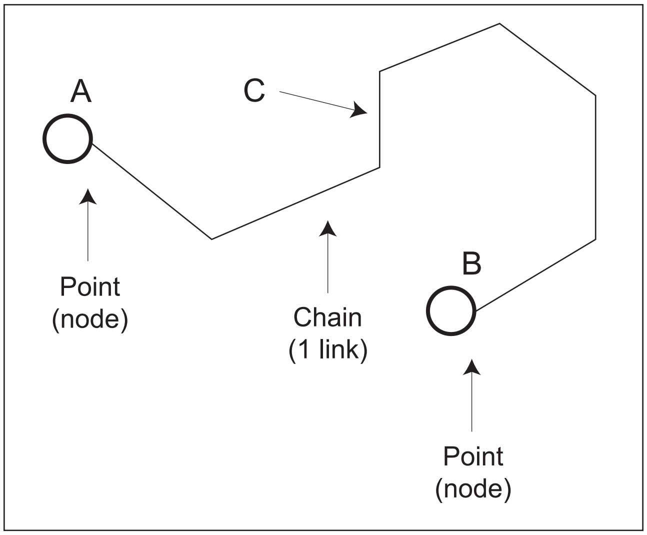
The missing link

Support for complex network models is built directly into the core of the Smallworld advanced spatial database (not as afterthought). This design focus provides an integrated, flexible and very efficient way to manage even the most complex of the world's networks.

Deep within the Smallworld architecture a set of simple spatial and topological structures are used in tandem to represent complex networks. This highly granular approach allows intricate networks to be constructed out of just three basic network entities: nodes, links and polygons. Nodes represent connection points (for example, a switch), links represent the connection between nodes (for example, a cable) and polygons are collections of links and nodes that make up an area (for example, a service region). These objects not only store information about connectivity, but are also linked both to the geometry used to represent them in the geographic world and, indirectly, to the object holding the asset's attribute information. This last point is an important one as it allows essential business rules that affect the logical topology of the network to be easily implemented. For example, a network topology defined purely in terms of its physical connections would have to be re-wired to reflect changes in state (for example, closing a switch). In the Smallworld architecture this behavior can be implemented on the owning object (for example, a switch) either by the inclusion of additional attributes (for example, an open flag) or supplemental code that can implement more complex business rules. Using this approach, the logical topology of the network can change on-the-fly responding quickly to the often dynamic operational needs of today's enterprise networks.

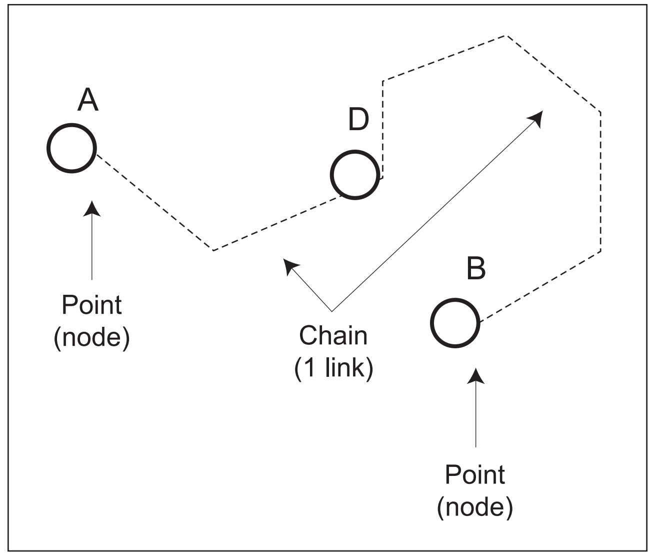


Logical collections of network objects can be grouped together into what is called a manifold. This is a group of assets that would normally be expected to interact with each other on a day-to-day basis. An electricity manifold, for example, might contain switches, cables and transformers. On the other hand, a water manifold might contain valves, pipes and hydrants. This important concept means that objects in different manifolds cannot interact with each other (an electricity cable can never connect to a water pipe, for example). Also, each manifold defines a set of rules that dictate how objects within a manifold interact with each other. These rules not only declare the condition that fires the rule, but also the action that is a consequence. Many types of rules are provided to control the interaction of these network objects, but in the interests of brevity, an idea of the capabilities can be realized by returning to the contrived example of the previous section.



In the Smallworld view of the world the two switches A and B are represented by what are called points. Each point is linked to a single node that stores its geographic location. The cable C is represented by what is called a chain. In this example, it consists of a single link. This link holds the route of the cable and also represents its connectivity.

The manifold for this network would contain at least switches and cables. A rule would be defined within this manifold that automatically connects the switches to the cable (when they overlap during input). This is called a connect-split link rule in Smallworld parlance. Adding a third switch, D, reveals the benefit of this approach. The chain representing the cable originally had only one link. Adding the additional switch causes the system to automatically split this link into two (one between A and D another between D and B) using the connect-split link rule.



This more granular approach makes managing geometries easier as there is now a close relationship between the connectivity model and its geographic representation (it is now quite easy to implement code to move D since the ends of the links it is connected to are readily available). Another benefit is that features can be more efficiently stored by sharing existing links. For example, two polygons that have a common face can use a single link to represent that face (as opposed to two) minimizing storage requirements and reducing the amount of data that needs to be processed during analysis.

This model also enables more accurate network analysis (for example, a route passing though A and terminating at D is completely represented as links, as opposed to the less granular approach which only models the geography as A to B). This might be a subtle point, but it is nevertheless an important one. For example, utilities often want to know as accurately as possible the length of the pipes required to connect as subset of assets for planning purposes. In other cases the length of an asset might determine its operational efficiency or safety.

Integrated on top of this flexible architecture are two important technologies designed to manage this network model data and provide the foundation of much of the analysis functionality required to exploit its true value.

The topology engine manages much of functionality provided by a manifold. All network enabled spatial data is first processed using the topology engine before it is stored in the database. An example of this is in the case where new network assets are first inputted into the system. Sophisticated editors allow vertices of chains to snap to existing network features as they are entered or edited improving data quality. Advanced core algorithms then process the new data automatically building up connectivity relationships and processing individual geometries (merging some, breaking up others) to maintain an optimized model of the network. Quality assurance tools are also provided for quickly validating connectivity using easy to interpret visual cues. All these features are typical of an integrated approach to network modeling that extends from providing low level structures and functionality right though to the user interface.

The analysis of the network is provided by a highly advanced and optimized trace engine. This functionality provides the backbone of numerous network analysis tools such as shortest path calculations or drive time analysis. This trace functionality is not constrained by the physical topology of the network: as links and nodes are traversed their owning objects may be queried on-the-fly allowing alternative paths through the physical topology to be chosen depending on the values of the object's attributes or a pre-defined business rule. For example, a two way street that becomes a one way street can be reflected in the network topology simply by changing the value of an attribute on the street object (the next time the trace engine visits this street, it will only be able to enter it from one direction). It also forms the basis of many extremely complex business functions such as electricity load analysis. The trace engine also provides sophisticated rendering capabilities allowing network traversals to be overlaid on background maps to emphasis relationships or display routes. Traversing a network can also yield a useful set of assets visited, supporting important regulatory reporting processes.

Both these key technologies form an integrated part of the Smallworld advanced networking solution and represent a consistent, reusable application programming interface (API) to support today's advanced network applications. By adopting this approach the Smallworld architecture has shifted the burden of modeling complex networks off the shoulders of the application developer and on to the back of an advanced network-focused architecture designed from the outset to fulfill this role.

The true role of maps

Much of this paper rightly concentrates on the importance of the network over the map, but it would be doing maps a disservice if it also ignored their true role: setting the stage for the network. If GIS were a play, then the network would be portrayed by the actors whereas the background scenery and sets would be provided by maps. This is an important role for maps. Background data such as streets, building outlines, landmarks and natural features such as lakes and rivers all provide an important means of putting network data in context. This allows the user to be quickly familiarized with local geography (for example, most persons still navigate using street names).

Obviously, pictorial information such as this does not need to be network enabled (it would be an unnecessary overhead in terms of both processing and storage). Consequently, the Smallworld spatial technology is able to cater to both of these different roles: providing powerful network functionality when it is required, and a leaner, less processor-intensive model when it is not. This approach gives enterprises the flexibility to decide whether their business processes require network intelligence or whether a simpler representation is more appropriate that focuses on compactness and performance.

Conclusion

Over fifteen years ago the architects of the Smallworld platform were the first to recognize the importance of the network in enterprise GIS. This foresight avoided the assumption that the map was the best way to model spatial data. By focusing on the network, the Smallworld platform has pioneered an integrated, advanced and flexible architecture designed to efficiently manage the world's networks.