The Generation of Schematic Diagrams from Geographic Representations of Networks

Paul Yates[1] & Mark Humphreys[2]

[1]Department of Geomatics
University of Melbourne, Parkville,
Victoria, 3052, Australia
Phone: +61 3 9344-9696 Fax: +61 3 9347-2916
Email: pmy@sli.unimelb.edu.au

[2]Smallworld Systems (Australia) Pty Ltd
G.P.O. Box 842 J, Melbourne,
Victoria, 3001, Australia.
Phone: +61 3 9614-7444 Fax: +61 3 9614-7555
Email: mark.humphreys@smallworld.com.au

*Presented at the 10th Colloquium of the Spatial Information Research Centre,
University of Otago, New Zealand, 16-19 November, 1998*

Abstract

Often, the geographic view of a network hides important connectivity information found in the network. In these cases a schematic diagram of the network data provides a representation in which connectivity relationships are highlighted. In this paper we explore issues relating to the development of information systems that not only support schematic representations of network data, but also allow for the computer-assisted layout of the schematic. A simple design that consist of two main components is investigated. The components in the design include (1) database management system software and (2) network layout software that implements common graph drawing algorithms.

A prototype implementation of the design is discussed. The implementation makes use of the Smallworld Spatial Resource Planning product to provide the underlying database management facilities. A Java application implements a simple graph drawing algorithm. Network data is passed from Smallworld SRP to the Java application. A schematic version of the network is developed using the Java application and then sent back to the Smallworld product.

Keywords and phrases: network data, schematic diagrams, graph drawing algorithms, Java, Smallworld.

1.0 Introduction

An important application of Geographic Information Systems involves the geographic representation of networks. Electrical, telecommunications, water and roads are all examples of networks for which a geographic represented maybe maintained using geographic information system software. Two related varieties of information are contained in a network; information related to the precise spatial location of network elements and information related to the topological connectivity between elements in the network. As the spatial location of network elements is essentially fixed, there is little scope for variety in how to represent this data in an information system. This is not the case when developing representations of the connectivity relationships in a network. To effectively display this information decisions must be made about exactly how to depict the data. The solution depends on the nature of the data, its use, quality and, to some degree, personal preference.

1.1 Some Motivation

The schematic representation of network data can be used as a tool for data capture, network design or simply as mechanism for network description (Bond, R. 1996). In many cases the correct spatial representation of linear data is obviously desirable, but in many practical implementations, the user of the data also has to call upon a view of the data which is untied from the constraints of geographical accuracy.

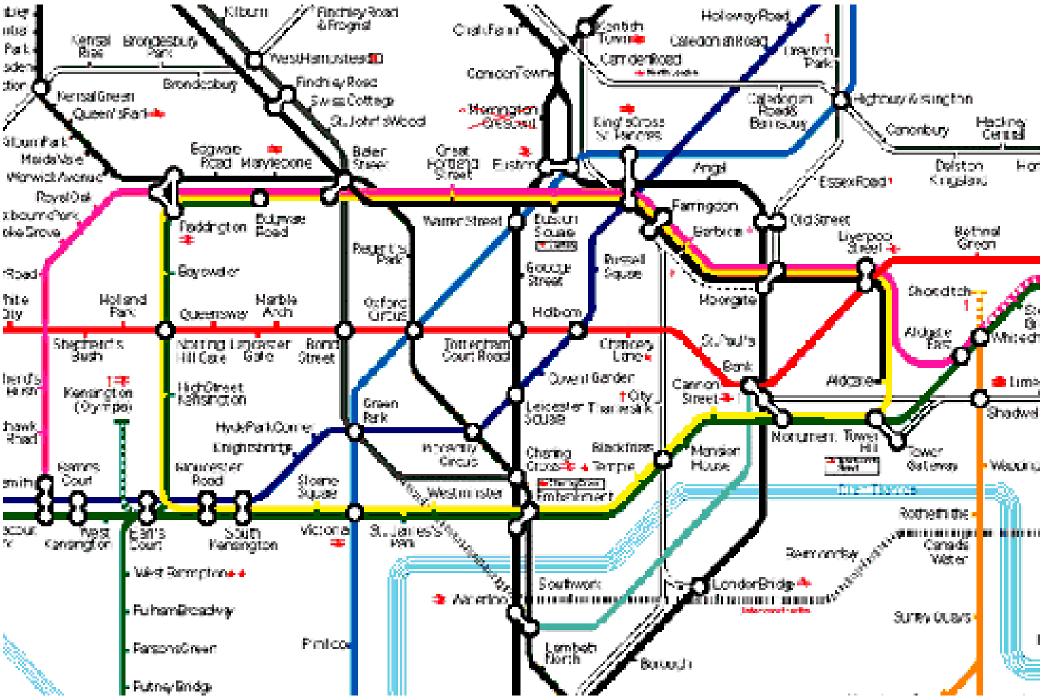

Figure 1. The London Underground Schematic: an example of a geoschematic. The geographic relationships between nodes (stations) in the network have, generally, been maintained. The edges (lines) in the network have undergone significant transformation (e.g. straightening etc.).

Consider an electrical engineer in the field attending to cable equipment. In order to carry out common tasks, they require many views of essentially the same data. A true geographic view is essential so that the engineer can accurately place the cable and can dig holes in the ground with confidence. However the benefit of schematic representations becomes apparent when describing the cable in detail. Many cables can share the same horizontal or vertical plane. This can confuse the interpretation of network information. Once an engineer has gained physical access to a particular cable, they are less interested in geographic accuracy and more interested in connectivity (*i.e.* which customers are connected to the cable I am about to work on). Such issues occur throughout the use of network data; as electricity current flows through complete circuits, water flows through sewer systems and traffic follows roads and intersections. At some point, the user of network data becomes less interested in the exact position of a cable/pipe/road and more interested in its relationship to the rest of a network. In explaining what a true network looks like we are forced into describing the network using a schematic representation (Bond, R. 1996). Nothing illustrates this point better than the famous (and often used) example of the London Underground map (a subset of which is found in Figure 1). Such cases form the motivation for the development of schematic representations of networks.

1.2 Types of Schematic Diagrams

There are a variety of forms of schematic representations that are available to a network modeller. The geographic view of the network is often used as the starting point for generating schematic data. Geographic accuracy is essential for computing the precise relationship between objects, *e.g.*, the relationship between a road and a water pipe or computing the precise length of a road, pipe or other linear feature. Schematic representations compliment this information by providing alternative views of the data, rather than directly replacing them. The schematics associated with networks are often primarily concerned with representing the topological connectivity relationship between the linear features represented. Often, it is also the case that some form

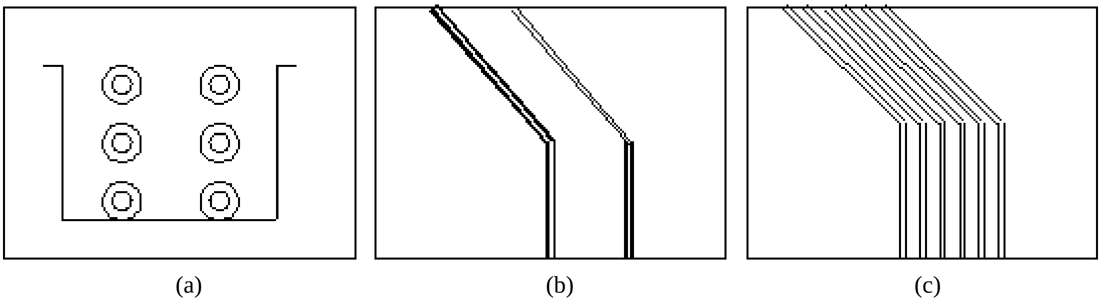

Figure 2. (a) the cross section view of a trench; (b) the geographic view of the same trench; (c) the schematic view of the pipes in the trench.

of "geographic" relationship is represented in the schematic diagram. For example, the relative positions (*e.g.* north of, south of, *etc.*) of electrical substations found in a electricity network may be approximately maintained. If this is the case then these representations are often called *geo-schematic*.

Geo-schematic views of data retain some of their spatial accuracy when displaying the network, but allow the freedom to alter some of the network to allow data to be more informatively displayed. Geo-schematic representations are used commonly by people mapping utility networks. Nodes within a network retain, or are close to, their original geographic location while the linear features within the network are exploded, pulled and otherwise transformed as required.

A second example of a schematic diagram with a different purpose appears in figure 2. In this example there are three different representations of a trench that contains several cables. First there is the cross-sectional schematic view showing how the cables appear in the trench. Second, there is a geographic (plan) view of the trench. And third, there is the schematic view of the trench showing all the cables that can be found in the trench.

Another form of representation is purely schematic. For these representations the geographic location of a node is not used when positioning a node in the schematic representation. Nodes are place with regard to connectivity only and complete manipulation of their schematic location is allowed so as to achieve the most effective result. What defines an effective result is usually subjective. There often are some desirable characteristics of a schematic diagram that can be used when determining the effectiveness of a particular diagram. These include the need for:

- straight edges between nodes in the schematic;
- a grid based layout of the network where each node appears at an intersection point of the grid;
- edges to be either horizontal or vertical;
- the diagram to cover the minimum area;
- the diagram to have the least number of edge crossings;
- *etc.*

Many of these nice features of a schematic diagram are in conflict; many are difficult to attain in either a manually or automatically generated schematic diagram.

2.0 The Requirements for Schematics

The full advantage a schematic representation is realised when it is used in conjunction with its geographic network. The geographic network provides answers relating to length, distance between points and similar spatial queries. However, visualising connectivity relationships is sometimes difficult using a geographic representation. In this case a properly built schematic representation provides the benefits of displaying connectivity in an uncluttered environment. Using both representations simultaneously is a significant step forward for understanding both the spatial and topological relationships found in a given network. For a system to provide proper support for schematic diagrams it should provide the ability to associate (at least) both a schematic representation and a geographic representation of an entity maintained in the system.

In properly providing two spatial representations of a network in a spatial database, the software managing the database must have the capacity to manage alternative representations of a given network element. As each spatial representation of an entity is usually different, a spatial *universe* or *world* should be available for each spatial representation of the entity. For example, for a substation in an electrical application there is the geographic representation found the geographic world that has geographic world coordinates, and there is the schematic representation in the schematic world with schematic world coordinates.

Given that we have a system capable of maintaining multiple representations of a given object in multiple worlds, the question arises of how these representations can be maintained efficiently. Clearly, it is not possible to automatically generate a geographic representation of a network from its schematic representation. However, as seen in this paper, it is possible to design a system in which it is possible to effectively, efficiently and semi-automatically generate a schematic representation of a network from the geographic representation.

Deciding to take this approach to schematic generation raises a whole set of new issues related to the common maintenance of geographic and schematic representations. Once a schematic is created (either manually or utilising various algorithms and modelling tools) there is the question of how to reflect changes between the two network displays. As the system now manages multiple representations of the same data, it also has to manage updates, deletions and changes accordingly. Such management can be either data driven or user driven. Some examples of how to implement these updates include:

- periodic update where "static" schematics are re-generated at given time intervals;
- automatic real-time updates, so that changes to data in the geographic view are immediately reflected in the schematic;
- initial schematic generation, leaving the responsibility of maintenance with the user community (some suitable editing tools may be provided); or
- simply regenerate the schematic each time it is required, so ensuring that the representation shown is the most up to date view of the geographic data.

These various techniques of how to manage schematics are due, in part, to the degree of emphasis placed on either generation or manipulation of the schematics.

Some practical examples of these concepts can be seen in any organisation migrating from a paper based representation of a network data to digital format. In the United Kingdom, Southern Electric Plc. hold their geographic data using the Smallworld Spatial Resource Planning product. Their geographic database includes raster backdrops, a vector based cadastre and electrical network information.

Figure 3 shows the geographic view of part of the network. Two schematic views are manually produced using this geographic. These are geo-

Figure 3. A geographic view part of a LV network, supply points on an LV network, HV network and a substation (Headington Road SS).

schematic in nature but differ in that one deals with purely Low Voltage (LV) network and one in HV (High Voltage). Because the users of the data are only interested in either the LV or HV side of the network the geographic data it can also be separated according to its attributes. This mirrors the way the paper based maps where organised but also allows Southern Electric to tailor the maintenance and upkeep of these schematics. LV networks are susceptible to more change than HV and so require more emphasis on the use of schematic manipulation tools.

Within the LV schematics system, Southern Electric implemented a number of manipulation tools. Once the network is divorced from its geographic ties, geometry can be moved and edited with a more “artistic” influence. These manipulation tools provided this freedom. These tools allowed engineers with years of network design experience to tailor certain sections of network according to their own preferences and network design ability. Such cannot be replicated in automatic generation algorithms, but, there is still an important role to be played by such network layout algorithms.

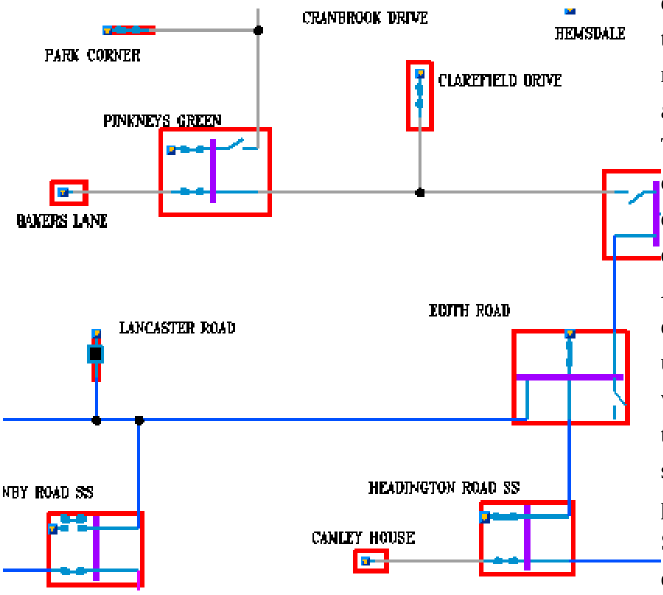

Figure 4. A schematic representation of a high voltage network.

A schematic representation of a section of the HV network in the Southern Electric database is shown in figure 4. The red rectangles represent substations (nodes in the schematic network). The blue lines running between substations are called load sections. They represent continuous non-branching sections of the high voltage network. The black dots at the intersections of the load sections are called HV joints and are also nodes in the schematic representation. Of note in this diagram is the representation of the internal networks of the substations. These are schematic diagrams in their own right; representing the internal elements found in a given substation.

3.0 The Systems Design

Essentially, the design of the system described in this paper consists of two significant components. The first important component is the underlying the database management system. The second component is the software that produces schematic diagrams (or layouts) for networks.

As identified earlier in the paper, beyond the standard requirements for a geographic information system, the needs for a system that manages schematic diagrams includes the requirement for supporting multiple spatial representations of a given element. For any given entity that appears in a schematic diagram we need an attribute of the entity that is associated with the geographic representation as well as an attribut that is associated with the schematic representation. The requirement for multiple representations of a given entity means that the spatial database management system should provide direct support for multiple representations. Another important requirement is for the database management system to be *open*. By using an open database management system we can make use of pre-existing software for the e network layout component of the system. As described in the section on the prototype implementation, we chose the Smallworld Spatial Resource Planning engine as it is both open and enables multiple spatial representations of entities.

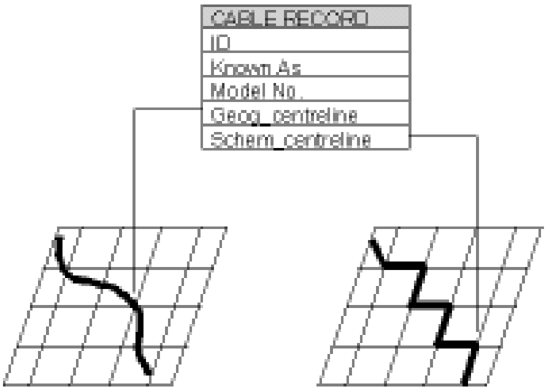

Figure 5. An example database entity (a cable).

The second component of the design consists of the mechanism for the semi-automatic generation of the schematic version of a network. Much prior research can be drawn upon for the implementation of a system that attempts to meet these requirements. The automatic generation of schematics has much in common with requirements associated with the drawing of graphs. A graph consists of a set of nodes and a set of edges. An edge is a pair of elements from the set of nodes. This type of graph is equivalent to the concept of a network. Solving the problem associated with generating a schematic becomes equivalent to solving the problem associated with drawing a graph. There are many algorithms for drawing graphs (Di Battista et.al. 1994). Some of these algorithms attempt to satisfy some of the requirements associated with schematic drawings.

3.1 The Database Structure

At the logical level (as we have stated already) we have chosen to model the geographic location and the schematic location as separate attributes of an entity to be modelled. Figure 5 shows an example of a cable element. There are a number of attributes associated with the cable entity; the important ones are geog_centreline and schem_centreline. Each of these attributes is a linear object. There is however a significant difference, the values for each of these attributes exist in different worlds or universes. One exists in the geographic universe with coordinates in a geographic coordinate system. The other exists in a schematic world with coordinates in a schematic coordinate system.

As we shall see in the section on the prototype implementation, we used the Smallworld Spatial Resource Planning engine for management of the data. One of the main reasons for choosing this software is its ability to model separate universes for spatial elements. In Smallworld, the number of available spatial universes is defined at the time of database creation. At creation time, the coordinate units and the maximum size of each universe (i.e. the largest coordinate which can be stored) is defined. Geometry such as lines, points and polygons are then created in a particular universe. The universe associated with a spatial attribute of an entity can be specified in the definition of the data model. For example, lines stored in the geog_centreline are specified in the data model to be created in the geographic universe and lines stored in the schem_centreline are specified to be created in a cable schematic universe. The benefits of this structure are realised when this approach is coupled with the object oriented nature of the Smallworld development environment. The object oriented nature of Smallworld allows database records and geometry to be treated as instances of “objects”, therefore a road, pipe or cable can all be interpreted as separate objects. Continuing this process, an object can be modelled to have more than one geometry associated with it, so facilitating multiple geometric representations of the same entity.

3.2 Algorithms for Drawing Graphs

The graph drawing (or network layout) component of the system the semi-automatic generation of the schematic version of a geographic network. It can be considered as an independent component separate from the other components of the system. Formally, a *graph G* is a pair *(V,E)* where *V* is a set of *nodes* (sometimes called *vertices*) and *E* is a set of pairs *<u,v>*, called *edges*, such that *u* and *v* are from the set *V*. A *2 dimensional embedding* of a graph in an x-y plane provides a x and y coordinate for each node of the graph. A *2 dimensional graph drawing algorithm* is simply any algorithm that provides an 2 dimensional embedding of a graph in a plane.

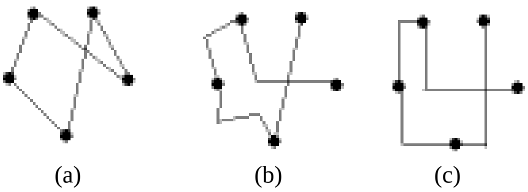

Figure 6. Some example graphs: (a) a straight line drawing; (b) a polyline drawing; (c) an orthogonal drawing

There are many different types of graphs, such as trees, directed graphs, directed acyclic graphs and so on. For each of these cases there exists specialist algorithms for drawing the specific type of graph. In general, however, for many of applications, the networks do not conform to these special cases of graphs. Hence, usually a general graph drawing algorithm will need to be provided. There are many 2 dimensional graph drawing algorithms (see Di Battista *et.al* 1994) that will embed a general graph. They come in a variety of forms (see figure 6), such as:

- straight line drawing algorithms where all edges are straight lines;
- polyline drawing algorithms where the edges generated are allowed to bend;
- orthogonal drawing algorithms where the edges are only either horizontal or vertical.

It is important that any graph drawing algorithm can plugged in and used as the graph layout component for generating schematic diagrams. This allows for a specific algorithm to be developed for a specific application and used as needed.

There are many graph drawing systems available (Scott 1995). There is no reason why the network layout component could not be an existing system. The only requirement is that the system is capable of interoperating with the database component of the system for schematics generation.

4.0 A Prototype Implementation

A simple prototype implementation of the software system following the design outlined in the previous sections has been developed. This implementation consists of three main components:

- a graphical user interface built using the Java language;
- an extension to a standard electricity data model implemented using the Smallworld Spatial Resource Planning Engine; and
- the use of a graph drawing algorithm (written in Java).

In the prototype implementation the Java interface is invoked from within Smallworld. The Java application then establishes a connection back to the Smallworld engine through the Transport Independent Client/Server (TICS) interface. The TICS interface allows connection to the Smallworld engine through a host and service name. Messages/data are passed between the Java application and Smallworld using the established TICS connection. There are two basic messages that are passed between the Smallworld and the Java graph drawing application. These are simply to *get a graph* and to *put a graph*. Before invoking the *getGraph* method on the Smallworld engine a subset of a network is selected using the standard network following mechanism found in Smallworld. The *getGraph* method places the currently selected network on the TICS connection for the Java application to read as the starting point for the schematic network generation.

The layout of the network is performed totally within the Java application. The graph drawing algorithm we have used is based upon one found in the standard Java development kit sample library (Sun JDK1.1.6 1998). The applet uses a heuristic algorithm that measures the cost of a given configuration and attempts to reduce the cost of the configuration by moving nodes contained in the graph. We have taken the standard GraphLayout applet and extended it so it now communicates with the Smallworld SRP engine to retrieve and return network data. A number of additional buttons have added. Using the standard applet enables for the centre node of a graph to be identified. This node then becomes fixed at the origin of the coordinate system. An important feature we have added to the standard Java applet is the ability to fix multiple nodes at their locations; thereby extending the concept of fixing the centre point at the origin. This enables the implementation of the geo-schematic requirement for the electricity networks. A number of

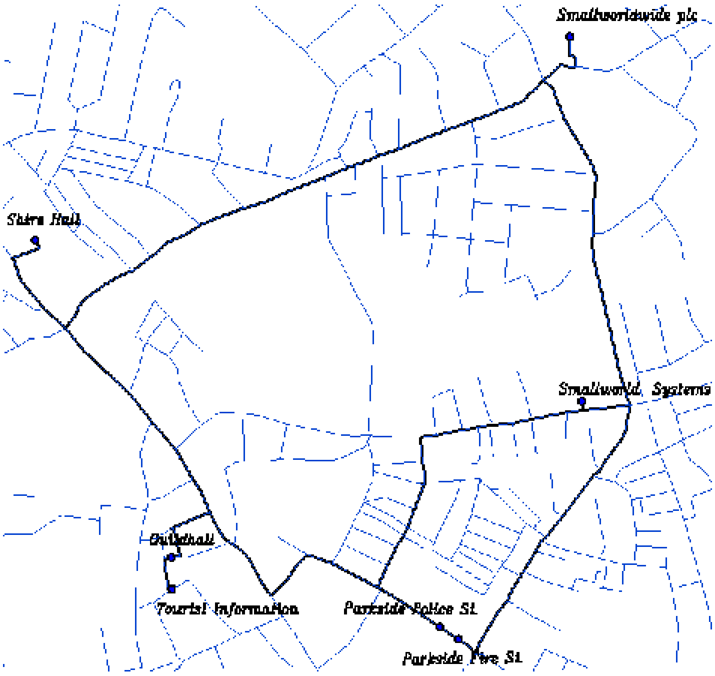

Figure 7. A geographic view of some places of interest on a road network (in Cambridge, UK), from a sample database.

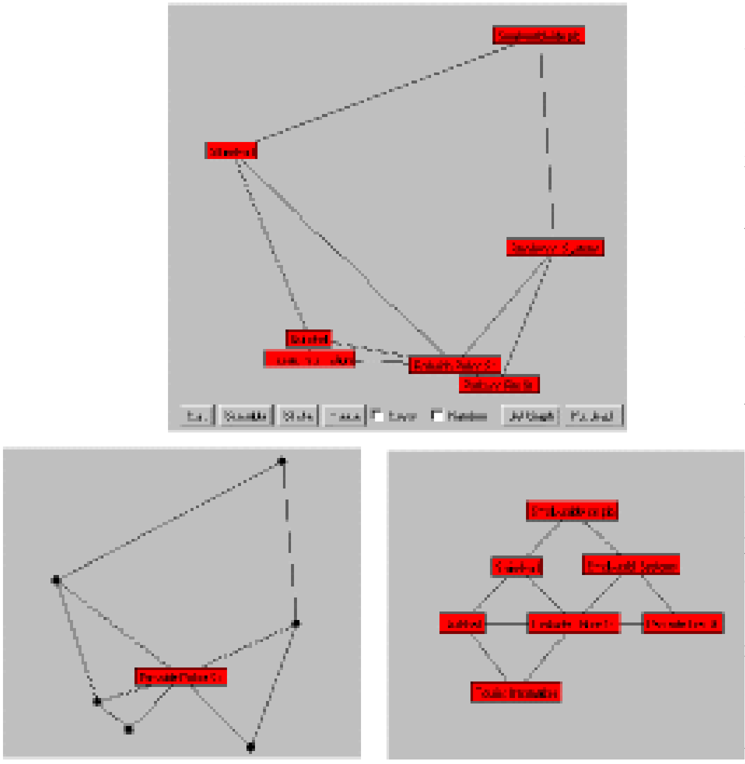

Figure 8. Three views of the graph layout applet in action. First, the initial geographic positions are retrieved from the database. Second, nodes are released from their original positions. In this case the Parkside Police Station is still fixed. Third, a final layout of the sample network.

important nodes (*e.g.* substations) can be identified as being fixed in their geographic location. This means that the geographic relations between these nodes are maintained during the drawing process. All other nodes are then "relaxed" and transformed so as to reduce the energy contained in the graph.

The schematic network layout is generated interactively using the applet. Nodes can be frozen, unfrozen, dragged, placed and moved using either the mouse or by allowing them to be relaxed using the graph drawing algorithm. At some point the user of the system will (or should) be satisfied by a particular graph layout to be used for the generation of the schematic diagram. At this point the locations for each node in the schematic are sent back to the Smallworld SRP system. These locations are used as input to a schematic diagram generation method. This method will collect up all the required input (*e.g.* internals *etc.*) and build a schematic diagram as found in figure 4.

A sub-network of offices from the Smallworld sample database for Cambridge in the UK appears in Figure 7. The same network was passed into the graph layout Java application. Figure 8 provides three generations of the layout of this network. The first is the original geographic view. The second generation is a snapshot of the layout for the network as it appears as the network layout algorithm in applied. The third generation is derived from a final "settled" layout of the network. It has been hand edited so as make sure that the labels do not overlap.

It is important to note that although the algorithm used in the prototype implementation is a heuristic algorithm any algorithm could have been used. In fact the use of Java in the prototype is not mandatory for the development of this system. It was used essentially because it was already available, it enabled the generation of a form of geo-schematic and the solution is platform independent.

5.0 Conclusion

In this paper we have investigated issues relating to the generation of schematic diagrams of geographic networks. Some important issues in the development of systems that support the generation of schematic diagrams include:

- underlying database support for multiple representation of a given entity;
- the ability to provide semi-automatic generation and update of the locational data associated with objects represented in a schematic world;
- the ability to define certain constraints on the layout of a generated schematic diagram (*e.g.* as found in geo-schematics).

These important issues were addressed in the design of the system described in this paper. The design of the system consists of two important components, the database management system and the schematic layout/graph drawing algorithm. In addition to the support for multiple representations, the database management system must provide for multiple worlds for each of the representations. For example, for this application, there must a geographic world in which geographic elements are stored, a schematic world in which schematic elements are kept, and in some cases a world associated with the internals of a network element (*e.g.* the internals of a substation).

To provide a solution to the network layout component of the system we have drawn upon research into graph drawing algorithms. Many algorithms have been developed for graph drawing. We have shown through the development of a prototype system that the use of one of these graph drawing algorithms has proven to be valuable for solving the problem of efficiently generating schematic diagrams for electricity, transportation, telecommunication and road networks.

Acknowledgements

Some of this work has been undertaken with the support of both Smallworld Systems (Australia) Pty Ltd and the Department of Geomatics, University of Melbourne. In particular, the authors wish to thank Neil Smith and Jim Green of Smallworld Systems who have been influential in outlining the requirements for schematic diagrams.

References

Bond, R (1996) A View of Alternative Approaches to Schematics in Telecoms/Cable T.V. *Smallworld International Confernce Proceedings*, 1996.

Di Battista G., Tamassia T., Eades P. and Tollis I.C., (1994) Algorithms for Drawing Graphs: an Annotated Bibliography, Accessed 29 July 1998, <ftp://wilma.cs.brown.edu/pub/papers/compgeo/gdbiblio.ps.Z>, June 1994.

Long P. (1998) Schematics, *Development Design Specification V1.1(draft)*, Smallworld Systems.

Newell, S. (1995) The Representation of Network Information in a GIS. *Smallworld International Conference Proceedings*, 1995.

Scott, A. (1995) A Survey of Graph Drawing Algorithms, Technical Report 95-1, University of Newcastle, Australia.

Sun JDK1.1.6 (1998) The Java Development Kit Software. Accessed 27 April 1998. <http://www.javasoft.com/>