Design of a Knowledge-Based Geographic Information System

A Thesis Submitted by

P. Jayaprakash

in partial fulfilment of the requirements for
the Degree of Doctor of Philosophy
under the Faculty of Technology

Department of Computer Science
School of Computer Science Studies
Cochin University of Science and Technology
Kochi, Kerala State 682 022
India

March 1999

Certificate

This is to certify that the thesis entitled **Design of a Knowledge-Based Geographic Information System,** is a *bona fide* record of the research work carried out by **Sri. P. Jayaprakash,** under my supervision and guidance in the Department of Computer Science, School of Computer Science Studies, Cochin University of Science and Technology, and that no part thereof has been presented for the award of any other degree.

Kochi
15 March 1999

Dr. A. K. Menon
Thesis Supervisor
Department of Computer Science
School of Computer Science Studies
Cochin University of Science & Technology

Declaration

I, **P. Jayaprakash,** hereby declare that the work presented in this thesis, entitled **Design of a Knowledge-Based Geographic Information System,** is based on the original work done by me under the supervision of Dr. A. K. Menon (Former Professor and Head, Department of Computer Science, Cochin University of Science and Technology), in the Department of Computer Science, School of Computer Science Studies, Cochin University of Science and Technology, and that no part thereof has been presented for the award of any other degree.

P. Jayaprakash

Kochi
15 March 1999

Table of Contents

Acknowledgements

First and foremost, I would like to express my profound gratitude to my research guide, Dr. A. K. Menon, former Professor and Head, Department of Computer Science, Cochin University of Science and Technology, for his invaluable help during the course of this research. This thesis is the outcome of education, able guidance, encouragement, and inspiration that I received from him.

My sincere thanks are due to Dr. K. Paulose Jacob, Professor and Head, Department of Computer Science, School of Computer Science Studies, Cochin University of Science and Technology, for his valuable help in providing me with all the facilities in the Department of Computer Science for the smooth progress of my research.

I am greatly indebted to my wife, Sheela, and my daughters, Divya and Deepthi, who put up with lots of difficulties and inconveniences throughout my research. But for their unremitting co-operation, my research would not have reached this stage.

I am particularly indebted to Dr. S. Sasikumaran, former Reader of this Department, for his suggestions and help during the course of this work.

I take this opportunity to express my sincere thanks to all other members of the faculty, technical staff, and the secretarial staff of the Department of Computer Science, for all the help and co-operation extended to me.

I am indebted to the Cochin University of Science and Technology for granting me the Junior Research Fellowship during the course of my research.

P. Jayaprakash

Table of Figures

Chapter 1

The Nature and Characteristics of GIS

GEOGRAPHIC Information System (GIS) is a broad term that is applicable to a system that uses computer facilities for handling data, referenced in the spatial domain, with the help of appropriate data sets, for carrying out spatial analysis, and for presenting the results in the most convenient mode of display. In a GIS environment, maps are used to visualise spatial data to reveal and understand relations between them. Maps are no longer just the final products they used to be, when they functioned as a medium for both storing and presenting spatial data. The introduction of digital data has brought about a split between the two functions of storage and presentation [Krack 1996].

In modern GIS, the map is replaced by a database accessed through a software system; the software simply reproduces graphic products that look like traditional maps, but with added visual representations. At this level, the technology only provides a replacement for the map, often at a much higher cost. The next level of technology requires more sophisticated software that can reorganise the raw data to discover new relationships. It is at this level that the real advantages of a GIS become apparent.

1. 1. DEFINING A GEOGRAPHIC INFORMATION SYSTEM

THERE are many definitions for the term *geographic information system,* each developed from a different perspective or disciplinary origin. See Appendix A. Some focus on the map connection, some stress the database or the software, and still others emphasise applications such as decision support. One of the most general definitions, developed by consensus among 30 specialists, is as follows:

> Geographic information system is a system of hardware, software, data, people, organisations, and institutional arrangements for collecting, storing, analysing, and disseminating information about areas of the earth. (Cited in [Chrisman 1997] page 7)

Unimpressive, as it may sound initially, this definition reveals all the crucial characteristics of a GIS, as the terms, system, data, information, organisations, *et cetra*, are expanded to their intended meanings. It also gives an idea of the sweeping range and scope for the applications of GIS.

In the digital perspective, a GIS could be considered to provide users not only with an array of tools for managing and linking attribute and spatial data, but also with advanced modelling functions, tools for designing and planning, and advanced imaging capabilities. While many of these capabilities also exist in other types of systems such as visualisation and virtual reality systems, modern GIS are unique because of their emphasis on providing users with a representation of objects in a cartographically-accurate spatial system and on supporting analysis and decision making [Mennecke 1997].

A GIS, therefore, must fulfil the following functions:

> Provide tools for the creation of digital representation of the spatial phenomena, also known as data acquisition and encoding.
> Handle and secure these encodings efficiently, by providing tools for editing, updating, managing and storing; for reorganisation or conversion of data from one form to another, and for verifying and validating those data.
> Facilities for information browsing, querying, summarising, and simulation, enabling the user to gain additional insight into theoretical or applied problems.

Assist the task of spatial reasoning by providing for efficient retrieval of data for complex queries.
Create human compatible output in varied forms of printed tables, plotted maps, pictures, sound, scientific graph, or other suitable multimedia modes.

1. 2. THE STRUCTURE OF A GEOGRAPHIC INFORMATION SYSTEM

IN the light of the above criteria, a GIS could be considered as a software that has several subsystems and which communicates with the various hardware devices in its environment. It includes a database (DBMS) subsystem for storing and managing data, and a graphics subsystem for displaying the relevant spatial data. These two subsystems are controlled by the computer operating system. They communicate, through the graphics workstations, with the users by means of a user interface and command language interpreters.

The devices employed in a typical GIS are high capacity disks for storing data in various formats (text, graphic, and image), digitising tablets, and scanners to enter graphic data, high resolution printers/plotters for presenting the results. Connecting the GIS computer to a computer network, will allow for the exchange of data with other users (Figure 1.1)

The focus of GIS design is slowly turning towards incorporating them with multimedia data types. Pictures, sound, animated sequences, and unstructured texts may form an integral part of geographically located entities in information systems. According to Kemp, multimedia's technical issues, such as storage, retrieval, display, and transmission, impose considerable additional demands on the GIS. Video/audio data are several orders of magnitude greater than even bitmaps or rasters, and they often require special storage devices. Storing and managing such data make GIS data

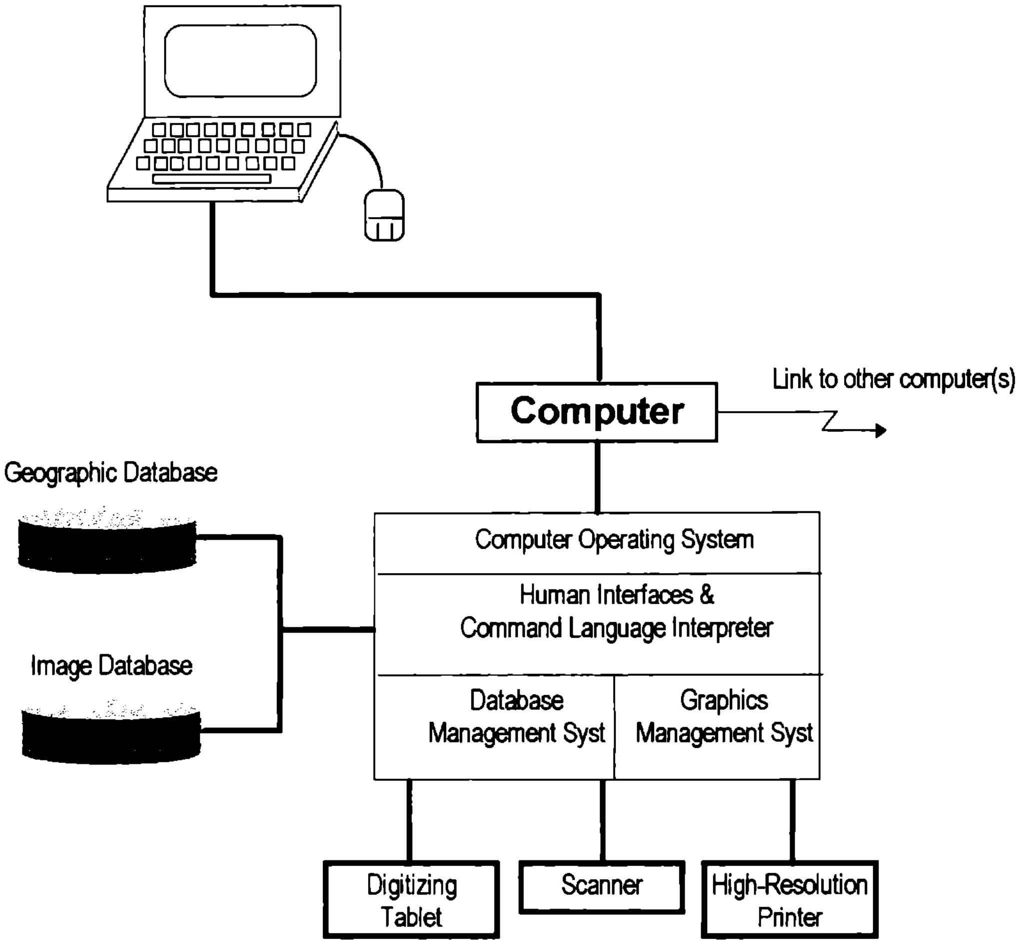


Figure 1.1. Components of a Typical GIS

management, already designed to handle large volumes of spatial data, even more complex [Kemp 1995]. Kemp cites other characteristics of multimedia data types that have negative implications in integrating them with GISs, based on the current state of multimedia technology.

As noted in 1.1, a GIS ideally must include the tools to keep track of data, bring information into the database, produce output, provide security, and allow for a range of spatially as well as non-spatially-oriented processing.

Data acquisition involves the direct capture of data, either through remotely sensed images, or close at hand, by still video cameras used on a field trip. This could also be accomplished through digitisation of accurate maps, tables of data from socio-economic surveys, *et cetra*, all of which have involved some kind of prior processing. Direct digital encoding via satellite is followed by an interpretation step.

Tools for **output** may be many, depending in part on the kind of information in the system. Analytically-oriented systems will most likely provide capabilities for the creation of physical documents in the form of maps, graphs, presentation graphics, tables, or narrative text. Query-oriented systems will focus more on the preparation of virtual output in the form of quickly produced maps or graphs, the contents of relevant segment of the database, or derived statistics, all made available immediately in response to a user request.

The **processing functions** are the tools for performing the spatial analysis. Again, these depend on many factors. A census data may require aggregation and classification. A query-oriented system is more likely to emphasise functions of search and retrieval.

1. 3. CHARACTERISTICS OF SPATIAL DATA

MAPS have been used as media for revealing details of shapes of entities as well as position and variations in attributes. While maps can clearly depict objects, simple or complex, unitary or compound, the cartographical techniques employed for effectiveness and the high information content are usually a challenge to the imagination of the map designer A **map model**, nevertheless, remains an alternative form of looking at reality so much so, that the designer of a GIS should be aware of the cartographical conventions of generalisation, feature displacement, and symbolisation. The map model has limitations and may distort the perception of space. It is based on a two-dimensional view of the world, and is only one influence on conceptualising spatial objects and their variations.

The computer-based organisation of digitised data into tables or other formats, is quite different from the map model. **Data model,** the term used for a comprehensive set of conceptual tools for organising data with a semantic basis, addresses the computing issues of minimising the storage requirements and enhancing performance.

Spatial data are different from the usual data stored in databases, as they involve an infinite number of points in space, instead of a fixed number of entities like "parts-id", or number of people in a city They are present in both intensional and extensional[1] forms of representation and hence there will be varying degrees of precision in the representions [Laurini & Thompson 1994].

Spatial entities can be viewed from several perspectives regarding dimensionality and precision. For some applications of GIS, the reduction in dimensionality associated with space-filling curves is important. The broad extension of this concept of precision of representation forces us to recognise that for many phenomena, boundary lines are artefacts, and that a field-oriented view can be more appropriate than an entity-orientation. In short, spatial data are more complex than

[1] In the extensional mode of representing a road (or a river), the end points of a series of segments are stored. A point between the end points of an aribitrary segment will not exist in the database. In the case of a non-spatial entity, like an amount or a number, data will always be extensional as every digit of an invoice amnount will be important.
When the road is stored in intensional mode, the end points as well as the equation of an arc (or other methods for constructing the arc) that could approximate the road are stored. Therefore any point between the end points of the road can be generated.

non-spatial data. They are special with regard to continuity in space, anisotropicity, sphericity, multi-dimensionality, duality (attribute and spatial) of access and intensionality [Laurini & Thompson 1994].

1. 4. THE EVOLUTION OF GEOGRAPHIC INFORMATION SYSTEMS

GEOGRAPHIC Information Processing have their roots in the history of thematic cartography Modern GISs evolved from the combinations of increased capabilities of computers, better analytical techniques, and a renewed interest in environmental/ social responsibilities. The earliest systems such as the Canadian Geographic Information Systems (CGIS) [Tomlinson 1976], Storage and Retrieval of Water Quality Control (STORET) [Green 1964], Management Information Assembly and Display System (MIADS) [Amidon 1964], were all consequences of the above factors and application requirements.

The GIS-user community is a diverse group drawn from a wide spectrum of disciplines. Foresters, geographers, city administrators, public utilities, academic researchers—all exploring the same territory, but from different directions and with different goals. As soon as the cartographers succeeded in the production of fairly accurate base maps, attempts were initiated for making thematic mapping wherein, different attributes of a given geographic region are appropriately defined and displayed. The forerunner to a geographic information system came into being when maps portraying magnetic variations with isolines, and wind directions with arrows were produced [Robinson 1982].

The transportation modelling study, commissioned in 1955 by the Detroit Metropolitan Area, might well have been the first GIS application. By early 1960's academicians were beginning to undertake research along these lines. If transportation studies were the first automated geographic analysis systems, with statistical capabilities, resource inventories gave widespread applications to the emerging field. Governments in Canada, and the United States began to see the immensity of opportunities that could be realised by the electronic storage and retrieval of land-based data. In 1962, Roger Tomlinson of the Canada Land Inventory, developed the Canadian Geographic Information System (CGIS) [Tomlinson 1976]. CGIS was designed for more than just one specific application. Designed as a polygon-based system, it had many short-comings in terms of real-time graphic editing, thereby preventing it from being an interactive system. Besides possessing storage and retrieval capabilities, CGIS could reclassify attributes, change scales, merge and create new polygons, and create lists and reports.

At the same time, the United States Forest Service was developing its own land Management Information Assembly and Display Systems (MIADS). This was a much more advanced system, one that could not only store and retrieve attribute of a given cell; but also perform simple overlay functions and mathematical calculations and prepare simulations over time. The outputs were generated on a line printer, but arranged topologically to produce cartographic images [Amidon 1964]. MIADS, therefore, could be considered as the first full-service GIS in the natural resource environment.

Along with the development of environment-based GIS models of the 1960's, there were similar attempts on large-scale, geo-based urban models. One such model was developed in Australia in 1970. TOPAZ (Techniques for the Optimum Placement Activities in Zones) is a general planning technique that has been applied at the regional, urban, and facility planning levels [Brotchic *et al.* 1980].

One of the early systems that has had a great impact in the applications-sector of GIS was the Dual Independent Map Encoded (DIME) file system of the United States Census Bureau. The DIME files represented a dual encoded map of streets and intersections along with geographic codes and addresses. Each street segment was coded twice, once as an edge to one block (right), and again as an edge to the other block (left). A block is a unique area defined by a boundary of street segments, and uniquely numbered. Geographic coordinates were specified for each intersection with each node and block. The computer could then construct two independent networks and then match them to ensure that all areas are accounted for and that the network is complete [Totschek *et al.* 1969]. Although technically the DIME file is only a data file, it has probably been used in more applications than any other system.

1. 5. THE IMPORTANCE OF GIS

THERE has been a profound change in the processing of geographic information since the emergence of information technology Even though a number of new journals and new technical societies in this field are being spawned, and several universities in the U. S. are instituting academic programmes in GIS at undergraduate and graduate levels, the importance of this technology is not well-understood, and it remains a grossly under-utilised technology

In April 1994, President Clinton of the United States issued an Executive Order under the title "Coordinating Geographic Data Acquisition and Access: The National Spatial Data Infrastructure," in which he justified the order by noting that

> "Geographic information is critical to promote economic development, improve our stewardship of natural resources, and protect environment. Modern technology now permits improved acquisition, distribution, and utilization of geographic (or geospatial) data and mapping. The National Performance Review has recommended that the executive branch develop, in cooperation with State, local, and tribal governments, and the private sector, a coordinated National Spatial Data Infrastructure to support public and private sector applications of geospatial data in such areas as transportation, community development, agriculture, emergency response, environmental management, and information technology "
> (Cited in [Guptill 1994])

The above order puts the importance of GIS in the current perspective.

Urban planners and cadastral agencies need detailed information about the distribution of land and resources in the area under their jurisdiction. Civil engineers need to plan the layout of roads and canals and to estimate construction costs, including those of cutting away of hill sides and filling in valleys. When a highway is planned to pass through a heavily populated state such as Kerala, its alignment at every zone could become controversial. This often results in a re-alignment, entailing cost over-runs and waste of manpower If a properly designed GIS is available for the entire state, the initial alignment can be drafted accurately and presented for public discussion. Any revisions resulting from these discussions can be made within hours, and the final alignment can be published within a matter of weeks, instead of many years. The police department need to know the spatial distribution of various kinds of crimes and accidents; medical institutions are interested in the spatial distribution of sickness and disease; and the commercial interests are eager to know the the distribution of consumer preference.

The enormous infrastructure of water supply lines, electricity distribution, telephone circuits, sewerage lines – all need to be recorded and manipulated in map-form, if the civic authorities are to react timely and efficiently to the breakdown in any of these public utilities. A robust GIS for a given town can display at different layers, the pertinent information of the pipeline layout, location of the valves, alternate supply route; type and route of power supply lines, interlinkings, location of transformers, location and types of circuit breakers, *et cetra.* This will enable the utility authorities to despatch the right technicians to the right place with the right tools, instead of perambulating across the town from one corner to another in search of the fault as is the current style of fault repairing.

A GIS is a tool box as well as a database. As a tool box, it can allow planners to perform spatial analysis using its geoprocessing and cartographic modelling functions such as data retrieval, map overlay, and connectivity [Berry 1987]. Map overlay is probably the most useful tool for planners who have a long tradition of using map overlays in the analyses of land suitability As a database, spatial and textual data can be linked by a geo-relational model for data query and retrieval. Planners can also extract data from the GIS database, use them as input for other modelling and analysis programme, or merge them with data amassed from spatially conducted surveys, for making planning decisions.

The inventory, analysis, and mapping capabilities of GIS have wide applications in urban and regional planning, ranging from data retrieval and site selection to project monitoring and implementation, development control, mapping and zoning analysis [Anthony 1991]. A GIS is most useful at the analysis, forecasting, and plan-formulation stages. Through the use of map overlays, problem areas such as urban redevelopment districts and environmentally vulnerable areas can be identified. At the plan formulation stage, land suitability maps can be generated using GIS for the preparation of master plans and land-use plans.

Remote sensing provides one of the latest and cheapest source of data of the terrestrial surface (*vide* page 5). Images recorded by sensors onboard geo-synchronous satellites are being beamed at regular intervals at favourable cost per unit area. Such images can be processed to produce basic data sets, combined with other types of spatially referenced data, and integrate with a GIS to generate value-added products. Various spatial information pertaining to urban areas can be extracted from digital remotely sensed images using new analytical tools such as pixel-based spatial re-classification procedure. The latter technique also allows surface and volume descriptions of buildings. Further, the object-based stereo digitising system allows heights and volumes of buildings to be aerially determined. The output of this can be imported into a GIS system [Barnsley *et al.* 1993]. In the area of soil conservation, approaches along the following lines are bound to be fruitful:

> A hydrological model can be linked to a GIS with special techniques that allow on-line data handling with time series grids, and the linked up combo can be used for Catchment run off forecasting and evaluation of the spatial distribution of run off production.
> Evaluation of spatial distribution of evapotranspiration and soil moisture content on a catchment scale.
> Distributed prediction of flow in different river reaches for flood or pollution control.

GISs can be designed, incorporating capabilities for the analysis of hazards. Hazards from surface instabilities such as land slides and subsidence can be modelled and embedded in a GIS [Wadge *et al.* 1993]. In fact, many countries are moving towards making evaluation of the visual impacts of any new development proposals, a statutory requirement, as part of the environmental assessment

It is the developing nations like India and the heavily-populated regions like the state of Kerala that should derive the maximum utility from tools like GIS. The uncontrolled and indiscriminate growth of every settlement—*panchayat*, municipal, or corporation areas alike—is turning the towns and cities fast into huge uninhabitable "slums." It is ironic that GIS remains an unknown entity in these over-crowded parts of the world, where it is needed the most

It would be fool-hardiness to think of GIS as mere means of coding, storing, and retrieving data about aspects of the earth's surface. In a real sense, the data in a GIS should be considered to represent a model of the real world domain, because these data can be accessed and transformed interactively, and they can serve as a test bed for studying the environmental process or for simulating the possible long-term effects of planning decisions. Such studies can preclude the planners' decisions turning into death-traps for future generations.

1. 6. MAJOR APPLICATION AREAS OF GIS

APART from the applications of GIS in the solution of environmental and social problems, as discussed in Section 1.5, there are many other areas where GIS can be put to productive use. These are categorised and discussed in the following Sub-sections.

1. 6. 1. Automatic Mapping

One of the first applications of GIS was that of capturing spatial data to automatically generate maps [Coppock & Rhind 1991]. Software designed to support automated mapping (AM) represents a powerful tool for business applications because it enables managers to generate spatial data in-house. It did not prove itself to be very popular because data capture was problematic for its users. There were other problems due to positional accuracy, attribute accuracy, completeness (are all the relevant objects included in the map?) and interpretive issues (is this a tree or a bush?), *et cetra* [Goodchild 1992]. The training of personnel was another constraint that dissuaded organisations from whole-heartedly accepting the idea of in-house map-making.

Although error propagation and training continue as problems, potential opportunities still exist for business to make use of AM tools. Remote sensing and global positioning systems (GPS) allow more accurate map production by removing paper map as a data source [Goodchild 1992]. In the United States, Electric companies lead the rest of the business in the use of AM, having begun early in the 1970s to use commercial GIS technology, for managing facilities data like the transmission line routing, locations of switching yards, locations of transformers, *et cetra,* customer records, order processing, and network analysis [Mennecke 1997].

Petroleum companies are some of the largest users of AM operations in the world. They have adopted GIS and digital mapping for supporting their operational and exploratory activities like managing well locations, lease information, and seismic information.

As with many technologies, AM applications have been caught in the Internet. InfoNow, a company based in Aurora, Colorado in the United States, has an on-line service called FindNow which enables a subscribing company to provide customers with information and maps showing the location of facilities, service centres, or retail locations. Visa Plus uses this service to develop an ATM (automatic teller machine) Location Service that allows Visa customers to locate the three nearest ATM machines to a specified street address or intersection.

1. 6. 2. Facilities Management (FM)

GIS fulfills "an ever-increasing demand for information pertaining to the location, condition, and performance of the utilities infrastructure" [Rector 1993]. The demand for information is not confined to utilities. Many organisations in other sectors must manage and control facilities such as manufacturing plants, distribution centres, retail outlets, and other departments of the organisations.

The key functions used in FM are the spatial visualisation, and database management functions. Most FM applications use historical or transaction (real-time) data for managing or monitoring facilities [Schek & Wolf 1993]. They rely heavily on the imaging capabilities of GIS. The AM functions of GIS (defined on Page 14) are often combined with FM functions to provide organisations with a system for generating, managing, and utilising maps and other spatial data, in the management of the organisation's physical plant [Mennecke 1997].

1. 6. 3. Transportation and Logistics

Transportation systems use tools and algorithms such as transportation network models and material flow models that came from disciplines such as operations research and production management. Specific GIS tools that fit into this category include vehicle routing and navigation system, intelligent vehicle highway systems, dispatch systems, production control systems, and inventory systems [White 1991]. Each of these technologies represent useful applications that managers can use to develop tactics to reduce waste, reduce personnel and fuel costs, and provide better customer service [Lapalme *et al.* 1992].

Logistical problems are common to many industry segments. Electric utility companies use GIS to produce location maps so that meter readers can plan their daily routes in advance; American Automobile Association uses GIS to support routing analysis and travel information reporting to its members. Car rental firms are increasingly including navigation systems in their rental vehicles. General Motors uses GIS-related technology to provide vehicle navigation systems. Railway companies use GIS for managing rail maintenance history by route and milepost down to each individual track. Federal Express uses GIS for tracking packages along their routes.

1. 6. 4. Spatial Decision Support Systems

Much of what managers do in business relates to planning and making decisions. Strategic decision making generally involves decisions that are broad in scope, unstructured, and focussed on long time frames. Information systems developed to provide strategic decision support to managers have generally been designed to provide access to critical data, analytic and modelling tools, and communication support. Many of these systems, however, represent spatial data and information inadequately [Mennecke 1997]. The term *spatial decision support system* (SDSS) has been proposed to represent easy-to-use systems that have capabilities for manipulating and analysing spatial data [Densham 1991]. SDSSs provide capabilities to input and output spatial data and information. They allow representation of complex spatial structures, and they include analytical tools for spatial, geographical, and statistical analyses. As such, SDSSs are an important class of GIS designed for use by middle- and upper-level managers. GIS can be effectively employed in strategic endeavours such as corporate downsizing, organisational restructuring, site selection (as employed by McDonald's), and competitive analyses (as employed by Time Warner Communications to determine cellular phone market potential).

A software tool, called Spatial Group Choice, developed under a collaborative project of the University of Idaho (Dr Piotr Jankowski) and the University of Washington (Dr Timothy Nyerges), and funded by the National Science Foundation, combines interactive multimedia map visualisation, with multiple critera decision models[1] This Collaborative Spatial Decision-Making (CSDM) system helps problem solving in a group setting, comprised of elected officials, interest group representatives, and the public, enabling the group members to develop consensus-based solutions to decision problems. Several experimental groups are testing the

---

[1] http://www.idaho.edu/~piotrj/
htttp://weber.u.washington.edu/~nyerges/
http://weber.u.washington.edu/~tjmoore/csdm.html

software on a real spatial decision problem of habitat restoration and economic development along the Duwamish waterway, near Seattle, WA, USA.

1. 6. 5. Design and Engineering

Computer aided drafting (CAD) systems have been routinely used by engineering firms to develop archive architectural drawings. GISs have the necessary features which, with suitable modifications, can perform the functions of a CAD system. While CAD systems have rudimentary links to databases and deal with relatively small quantities of data, they do not allow users to assign symbols based on user-defined criteria. They are also constrained by limited capabilities. Organisations that are involved in engineering and design (*e.g.*, Electric utilities, communication engineering, highway construction, *et cetra*) would benefit immensely by coupling their GIS usage with CAD technologies.

ARC/INFO is an extensive GIS with vector and raster capabilities including import, georeferencing, editing, analysis and output. [Mayall & Hall 1994] found that by integrating ARC/INFO®[1] with AutoCAD®[2], and in conjunction with other models, landscape changes could be simulated or predicted. They used ARC/INFO Macro Language to output GDS command files from the graphic and attribute data stored in the GIS. This procedure allows an end-user to obtain quite realistic 2D and 3D digital representations of current landscapes and to produce visual simulations of landscape change. It also allows numerous planning applications, such as site location,

---

[1] ® ARC/ INFO is registered trade mark of Environmental Systems Research Institute, Inc., USA.
[2] ® AutoCAD is registered trade mark of Autodesk, Inc, USA.

environmental impact analyses, transportation routing, and resource allocation, to be undertaken more efficiently than is possible using the traditional, manual methods.

1. 7. CURRENT ISSUES IN GIS

IT is difficult to draw up a comprehensive list of topics which are consuming the interests of those involved in GIS activities. However, they can be grouped into broad areas and the topics in each area be identified as in the sub-sections to follow [Morrison 1994]. The technology being relatively young, it should not be surprising to note that the areas are not disjoint; but with perceptible overlaping among the areas.

1. 7. 1. Data Related Area

The topics that could be linked to this category are

Spatial data integration
Data access methods
Data conversion
Object-oriented approach

1. 7. 1. A. Spatial Data Integration

The majority of GISs currently in use are based on relational databases, wherein attribute data and their spatial references are organized and handled separately In ARC/INFO, for example, the attributes are stored in conventional DBMS, but spatial data are manipulated using conventional file handling techniques. The gain in flexibility attained through this approach is countervailed by the fact that coordinate data are not subject to the same rigorous checkings as in the case of attribute data so that, in these schemes, spatial data suffer degradation in quality control in terms of security and integrity of data. The Starburst project [Lohman *et* *al.* 1991], [Widom 1996] at the IBM Research Centre at Alamaden, California, and the POSTGRES project [Stonebraker *et al.* 1990], [Stonebraker 1995] at the University of California, Berkeley, are developing extensions to standard relational database systems.

There is a great deal of interest in the United States in the creation of large spatial databases that would constitute a national geospatial network. This framework database is likely to be a federation of discrete geographic databases, each maintained by the respective participatory agency The major technical challenge is to synchronize these discrete databases into a federated database system so that they appear as one unified database [Guptill 1994].

1. 7. 1. B. Data Access Methods

In alphanumerical databases, retrievals are normally attribute-based; but in spatial databases we have to retrieve data based on location. Retrievals in spatial databases are made faster through the use of spatial indexing. The subject of spatial indexing in a database context is one of the most difficult problems in GIS.

There are several methods for producing a spatial index, using space-filling curves, cell trees, quadtrees, R-trees, R^*-trees, or sphere trees; but none emerges as easy or tractable. Only a few commercial GISs today provide spatial indexing capabilities [Ooi 1990], [Laurini & Thompson 1994].

Spatial databases often are split into several thematic overlays, each concerning a particular aspect, for example, the street layout, sewerage system, network of canals, *et cetra.* For handling these layers, it becomes necessary to create as many indices as there are layers. When a given area involves several thematic layers, the processing of countless indices becomes time-consuming.

1. 7. 1. C. Data Conversion

The Global Positioning System (GPS) of the navigational satellites give access to positional information in near and real time, thereby enabling accurate and rapid surveys. Position data from GPS will therefore find increasing usage along with existing GISs, for new applications or updating an existing GIS. The integration of GPS data with spatial databases, however, are not easy tasks [Dodson & Haines-Young 1993]. Inconsistencies occur due to discrepancies in the definition of map data, or due to internal mismatch within many of the older data used for the existing GIS

1. 7. 1. D. Object-Oriented Approach

As noted in 1.7 1.A., the general-purpose databases are inadequate when dealing with spatial information. The existing GISs are incapable of storing the earlier states of the databases; previous states are discarded as soon as a transaction is committed. A database associated with a GIS should incorporate time as part of the semantics of the system so that, not only spatial, but also the temporal, information of the relevant geographic objects could be elicited.

Object-oriented (OO) approach persuades one to view objects as basic constructs for modelling the problem domain, rather than taking an implementation-oriented approach to the problem. The Edinburgh group has shown that the static, atemporal limitation of a GIS can be alleviated through a data-centred model, named

TCObject data model, which provides a user-oriented data structuring, along with temporal and spatial semantics [Ramachandran *et al.* 1994]. Notable, among others, who have reported their activities involving object-oriented data modelling or OO databases, are [Worboys *et al.* 1990], [Worboys *et al.* 1993], [Ervin 1993], [Hamre 1994], [Shaw *et al.* 1996], [Sendler 1996], and [Shekhar *et al.* 1997].

The major advantage of OO Technology (OOT) is that it can encapsulate old applications, making the transition to the new system less painful. In practice, however, it has been found that OOT introduces new ways of developing software, requiring new tools, new programming paradigms, and new metrics, thereby masking the anticipated advantages.

While the field-based models of GIS see the world as a continuous layer of surface over which features (*e.g.*, elevations, road crossings, *et cetra)* vary, the object-oriented models treat the world as a collection of recognisable objects corresponding to different user-views. Some investigators have been attempting to couple an OO database containing actual data for class descriptions, semantic relationships, spatial and aspatial attributes, *et cetra,* with a deductive database containing rules and procedures for deriving spatial and aspatial relations [Smith *et al.* 1987], [Jones & Luo 1994], [Ku *et al.* 1994]

1. 7. 2. Technology Related Areas

There are areas besides those covered by 1 7 1, that are receiving intensive attention of researchers involved with geographic information systems, and some of them are listed below

Knowledge-based approaches
Spatial analysis and models
Temporal models
Multimedia Integration with GIS

1. 7. 2. A. Knowledge-Based Approaches

The design of a GIS faces great challenge in the storage, retrieval, and processing of many large and complex data. Knowledge representation in traditional programming is procedural and only complete and structured knowledge, restricted to certain variety, can be represented appropriately In knowledge-based systems, however, domain knowledge is formalized in a declarative form and allows for a natural representation of a large variety of knowledge types (fragmentary, unstructured, uncertain, incomplete, approximate, as well as structured, complete, certain, *et cetra*). For these and other reasons, knowledge-based systems have a reputation of being very flexible and general, and hence suitable for many applications.

For some GIS tasks it will be appropriate if the knowledge is embedded within the procedural code. For example, an algorithm to perform an overlay operation on two sets of spatial data could contain the knowledge of how to perform the operation. In many other tasks, it is more appropriate to keep the knowledge separate, within a knowledge base. A GIS for selecting the site for locating an airport, or a petrochemical complex, would rely on separate knowledge bases relating to different areas of expertise (aviation expert, meteorologist, petrochemical expert, spatial modeller, earth scientist, town planner, land developer) to fill their rôles in an integrated spatial decision support system for users with different backgrounds. The approaches and methods currently available for KBS development, however, are partial or non-systematic [Roman 1990], [Guida & Mauri 1993], [O'Neal & Edwards

1994]. In spite of the efforts for formulating methodologies for the design of knowledge-based systems along the patterns adopted for software engineering, there is a scarcity of concepts and methods regarding the important aspects of development of knowledge-based systems like, formal requirement specifications, evaluation of performance, complexity, and quality. A major reason for this is that the life-cycle of knowledge base design is different from that of software engineering.

1. 7. 2. B. Spatial Analysis and Modelling

Statistical analysis techniques for spatial data are of considerable value in increasing the capability of a GIS in spatial information management and analysis. Even though the need for integrating GIS with spatial analysis has been recognised during the past several years, it is hard to find a commercial GIS today that is equipped with spatial analysis tools. The most important reasons for this lacuna are, the obscurity of the field of spatial analysis, the lack of a comprehensive spatial analysis package, and the complexity of the spatial analysis techniques [Goodchild *et al.* 1992]. One group at the Canada Centre for Mapping has developed a prototype of a package for statistical analysis of spatial data [Zhang *et al.* 1994]. Spatial autocorrelation measures the degree to which a spatial phenomenon is correlated to itself in space. Investigations by Chou [1993] indicate that the effects of spatial weighting functions on statistics of spatial autocorrelation can be identified from correlograms that show higher-order spatial relationships; these were very difficult tasks earlier

1. 7. 2. C. Temporal Models

Space and time—*where* it happened and *when*—are fundamental to decision making and prediction. Most research into spatio-temporal representations has focussed on extending relational database models, treating time as a point or interval. [Raafat *et al.* 1991] introduced an extended time-domain relational algebra into the DBMS, to manage image-processing histories and analyse remotely sensed data over time. More recently, spatiotemporal models have been proposed for recording spatial changes over time, relating to specific geographic objects instead of their locations [Pequet & Wentz 1994], [Worboys 1994], and [Ramachandran *et al.* 1994]

1. 7. 2. D. Multimedia Integration With GIS

Thematic map is a powerful visual medium for presenting geographic data. With the advent of digital maps, many new methods were opened for presenting temporal geographic data, more as a phenomenon, than as a series of flat maps that correspond to data at different points in time. This is further augmented with the arrival of multimedia (MM).

Multimedia allows for interactive integration of sound, animations, text, and video images. While a conventional GIS can work with coordinates, pixels, their attributes, and spatial relations, MM technology offers links to all kinds of other geographic information. These could include text describing a parcel of land, photograph of objects that exist in the GIS database, or a video showing the environment of the area. The image of an old map could conveniently be linked to the system via hyperlinks, thereby obviating the laborious process of encoding the map.

Individual MM elements (video/still images, sound, *et cetra*) can be linked to maps to enhance geographic information for visual exploration, analysis, and presentation. While some of these applications have been realized as stand-alone development projects, they have not been incorporated effectively into full GIS [Kemp 1995], [Krack 1996]. A major burdle for integrating MM and GIS is the lack of a logical data model for spatial and aspatial data that can subsume MM.

In conventional databases, data items are represented by a set of attributes that are of basic data types. Such schema are not suitable for representing MM objects. MM database systems should manage voice, image, video, text, and numeric data, uniformly Because information consists of several types of data, these databases should, in some cases, evaluate a query condition by referring to several data that are of different types. The type of the query condition given by a user, may differ from the target data to be rerieved. To illustrate, consider the changes in landscape of a given area, recorded over a period of time, and stored in a MM database. The attribute is represented by a series of video clips, showing the images of landscape at different points on time. The particular video clip for retrieval and display will depend on the period of time indicated in the query

In other applications, the query may be similarity-based, intended to "find something similar to 'that one' " Instead of *key-based* retrieval, what is needed in MM databases, therefore, are *similarity-based* queries and *content-based* retrievals. In large databases, indexes are crucial for data retrieval at reasonable speed. But conventional indexing techniques like B-trees and inverted files are not effective with similarity-based queries and content-based retrievals. Researchers from different fields, ranging from GIS [Shepherd 1994] to medical imaging [Petrakis & Orphanoudakis 1993], are trying to find solutions to to these problems, using techniques of neural networks [Wu 1997], and object-oriented data model [Yoshitaka *et al.* 1994].

The methodology of management of MM databases are still in a fluid state. While in DBMS, the focus is on search and indexing techniques, in VDBMS (video database management system), efficient retrieval operations of video in a terabyte database remains a great challenge [Yeo & Yeung 1997] Commercial object-relational database systems[1], which are the state-of-the-art for implementing MM database systems have been unsatisfactory in intuitive query environments that are characteristics of the MM usage [Grosky 1997].

1. 8. GIS AND HUMAN FACTORS

BECAUSE of the visual nature of GIS, issues related to the nature of the task, the visual layout and presentation of the display, and the cognitive effects that these issues produce in the user, are all critical considerations in the use of GIS. Not surprisingly, therefore, extensive research on the human factors *vis-à-vis* GIS is under way, under the broad area of computer application known as *Human Computer Interfaces* (HCI) [Adam *et al.* 1997], [Gentner & Grudin 1996], [Vetter *et al.* 1995], [Nyerges 1993], [Turk 1993], and [Benbasat 1986].

[1] DB2 Universal Database (IBM): http://www.ibm.com
Oracle Media Server (Oracle): http://www.oracle.com
CA-Jasmine (Computer Associates): http://www.cai.com/products/jasmine.htm

HCI is less mature as a technology but has lower entry barriers than system architecture. Challenges related to HCI include how to serve a much more diverse and less expert user community than that of professional programmers and how to support a wider variety of applications in an environment of increasingly powerful and sophisticated technology [Adam *et al.* 1997]. HCI can enhance the "convenience-of-use" factor of applications by providing better interfaces and by conducting studies to determine the effectiveness of alternative approaches to interface design.

Fundamental to this area of inquiry is the consideration of the physical characteristics of the system. Issues such as the layout of features on the screen , the colour and saturation of display objects, the number and type of display objects used, the nature of the input and output devices, and the arrangement of the physical components of the system—all have important impacts on the way people interact with the technology [Benbasat *et al.* 1986], [Turk 1993]. With the wide range and scope of applications that GIS may be used for, it is also important to consider task characteristics when studying human factors in GIS. A task framework is needed to provide a better understanding where GIS should be used and how it should be applied to specific applications [Nyerges 1993]. Another important issue in human factors research relates to the cognitive characteristics of GIS users[Turk 1993], [Traynor & Williams 1997]. It has been found that differences in individual spatial cognitive abilities had important impacts on decision maker effectiveness and efficiency [Crossland *et al.* 1993]. A better understanding of cognitive skills and styles, influence of individual cognitive characterstics on user effectiveness — all these factors have impact on a wider acceptance of GIS.

1. 9. APPLICATIONS AND DEVELOPMENT OF GIS IN INDIA

1. 9. 1. Activities at WWF-India

Indira Gandhi Conservation Monitoring Centre (IGCMC) was established in New Delhi, India, by the WWF[1]-India, with support from the Government of India. Using a GIS, built from data acquired *via* satellites, IGCMC has prepared several databases for ten major protected areas, *viz.*,

Buxa
Ranthambhore
Periyar
Palamau
Pench
Kalakad
Mundanthurai & Simlipal Tiger Reserve
Nagarhole (Renamed as Rajiv Gandhi) National Park
Gir National Park
Great Himalayan National Park

These databases will be used to study the forest cover changes, contours, reserve-forest boundaries, and the distribution of endangered plant species. They will also assist in conducting diagnostic studies to identify the frailties of these areas and to launch corrective measures and sustainable developments.

In the case of the protected areas, a project covers wildlife distribution, human settlements, forest-type distribution, river networks, *et cetra.* For a second project on endangered plant species, IGCMC is preparing spatial distribution maps, to be used for assessing conservation threats at regional scale using other spatial data such as forest cover, population pressure, forest contiguity, and the legal status of forest areas[2]

[1] World Wildlife Fund (now renamed as World Wide Fund for Nature).
[2] htttp://www. igc.apc.org/wri/wr-98-99/wr98indi.html

1. 9. 2. Activities at NRSA

At the National Remote Sensing Agency (NRSA), Hyderabad, India, attempts are being made to build an expert system that could be used as an interface to ARC/INFO GIS [Hebbar 1991].

NRSA has developed two systems — one under a project for Drought Monitoring and Prevention, and the other under a project for Land and Water Resources Monitoring. The former system became operational in 1989. The main input data source for both these systems are satellite, and the output for both are in the form of tabular data.

National Informatics Centre (NIC) of the Planning Commission of India has an active project for designing a GIS for administrative and planning purposes. The input data sources used for this project are satellite data, census data, and field data. The outputs will be maps, tables, and summary reports. These are available over NICNET (NIC Network), but with access limited to government officials. The system has been operational since 1990; unfortunately, these websites do not get updated—some are of several months "down-to-date"!

1. 10. MOTIVATION FOR THE PRESENT WORK

IT was pointed out earlier that the majority of the currently popular GISs were built around relational databases (*vide* page 19). Relational databases are severely handicapped when it comes to handling spatial data. The situation is compounded when it comes to formulating queries even for the simplest of spatial operations [Orenstein & Manola 1988]. Information in a relational data model is scattered.

Implementation of the normalisation requirements tends to allocate data to small relations, disseminating the attributes of a particular entity into several separate but connectable tables. These factors make queries in relational databases cumbersome. Because the relational database requires that records must be of fixed lengths, spatial data have to be stored in external files, and this degrades the access times of GISs associated with relational databases. Moreover, the relational data model lacks the ability to specify explicitly the semantic information about relationships [Mohan and Kashyap 1988].

While conventional databases have been successful as data transaction record processing in the business world, the domains of geometric and multimedia GISs require more flexible and expressive approaches than that provided by the relational model. Therefore, object-oriented approaches, developed as a result of the grafting of several roots including artificial intelligence concepts such as 'objects', 'frames', 'abstract data types', 'instances', 'rules', 'inferencing', *et cetra* — collectively known as knowledge-based approach — offer greater promise for the future of GIS. In such an approach, a feature, simple or complex, is reached not *via* attributes but directly; and an entity has an existence separate from the descriptive thematic data.

The progress in computer visualisation and multimedia technology is liberating the field of cartography from the domain of maps, into an entirely new process of communication [Fraser 1994]. With the advent of data acquisition *via* remote sensing satellites, the need to convert increasing volumes of data into usable information is increasing. The computer is becoming a direct source of proliferation of data as well as new ways of visualising it. The driving force in all these changes has been the need for increasing the interactivity between the user and the computer — much more than the new technologies. The urge for increasing the interactivity between a GIS and its user, has also been the principal motivation for the present work.

For realising this objective, a GIS has been designed, invoking the recent techniques in the representation and manipulation of knowledge, while treating the geographic data as "minute pieces of knowledge" that are inter-related within the domain of a map. The knowledge contained in the points, lines, and textual descriptions, are stored in memory as rules, using OPS5, a production system. OPS5 has the facility for analysing the code during execution. It can start the execution with a set of overly specific rules, and generate new rules that are more general and abstract. When the system over-generalises, it would backtrack or specialise some of the rules that are too general.

The major advantages of using a production system in the design of a GIS are the following.

> A production system is modularised at the level of the working memory elements. According to [Harmon 1995], it is only when a developer uses both objects and production rules that he has really modularized an application in the most systematic manner.
> In formulating the rules, it is not required to make a distinction at the top level between spatial and aspatial components.
> Data will be accessible in an application-independent manner to many users,
> All the data in a production system are brought under one umbrella, with consequent advantages in terms of integrity, reliability, and security.

The data model conceived for the present research was based on a three-pronged strategy. Firstly, it would be a vector representation strategy because geographic data entities are stored as a vector of coordinate points that define the geometry and topology of these entities. Secondly, it would be a logic-based structure because two geographic entities, the polygon and the line, are represented as predicate symbols describing certain relationship between a series of objects that are attributes of space [Jayaprakash and Menon 1995]. Thirdly, it is a relational structure because the polygon and the line predicate structures can be interpreted as implementation of two relations (tables).

1. 11. SUMMARY

THIS chapter has provided an overview of the concepts and issues associated with the GIS technology The peculiarities of geographic information processing, *vis-à-vis* general information processing, have been briefly outlined, and the problems and difficulties in building a GIS, high-lighted. The enormous potentials of GIS, brought about by the rapid growth in database, visualisation, multimedia, and communication, *et cetra,* have been pointed out. The importance of GIS has reached such a level that the United States has launched the massive National Spatial Data Infrastructure for coordinating geographic data acquisition and access. Major applications of GIS include automated mapping, design and engineering, demographic and market analysis, environmental monitoring, facilities management, natural resource management, strategic planning and decision making, and transportation and logistics. Current issues that surround GIS, human factors that have a bearing on the quality of GIS, have also been discussed. In the end, the features of the GIS implemented under the present research are described briefly

Chapter 2

A Review of Research in Modern GIS

A GEOGRAPHIC information system essentially consists of a digital database with digital representations of real world entities, called *objects*. Objects have *spatial attributes*, *content attributes*, and may have relationships to other objects. Many entities are captured from the real world by sensors and converted automatically to objects. This conversion is based on a series of geometric corrections applied to the spatial attributes, and on extraction rules for content attributes and relationships. Objects may also have spatial attributes that are derived from global positioning systems. These positions are then attributed from scalar data sources such as censuses and boundary surveys. The digital objects, with their spatial and content attributes and relationships form the digital database.

2. 1. CONCEPTS AND SEMANTICS OF SPATIAL DATA

GISs are considered as systems that require high investments; and a large portion of this investment will be on acquiring, converting, and storing new data. As noted earlier (page 6, para 3), spatial data are distinct from data usually stored in databases.

Ignoring the trivial case of point entities, or the geometries of the one-dimensional objects in Euclidean geometry, lines may be represented by:

Real position, using Cartesian or polar coordinates;
Some generalisations of the vertices for the line features using procedures that depend on coordinates;

- Topology, ignoring geometry;
- Parametric representation.

Some data are discarded if only generalised versions are maintained in a database. But the data volume can be lowered by reducing the number of coordinates, or, under some conditions, by the parameters (intensional representation) or equations. Coordinate-based representations are associated with digitizing or line-following forms of data capture. Procedures for simplification and data reduction are usually applied to those data unless the entities are regular geometric figures.

Normally, however, detailed geometric representations are dealt with by stored data for positional coordinates. In the absence of good reasons for representing phenomena intensionally, *viz.*, by rules applicable to regular geometric figures, then devices are required for adequately and effectively recording position extensionally Quite a number of spatial databases existing today are characterised by Euclidean methods, projective geometry, and by extensional data [Laurini & Thompson 1994], [Burrough 1986].

2. 1. 1. The Terminology

The term **entity** is used to refer to a phenomenon that cannot be subdivided into like units. A house, for example, cannot be subdivided into houses; it can only be split into rooms. An entity is referenced by a single identifier, a city name or a building code. In the spatial information scheme, there is no comparable standard structuring of phenomena as in the physical world's molecules or atoms, that would help us in recognising the **scale** of phenomena as implied by divisibility. While in everyday language the terms entity and objects are synonymous, in GIS, as in the field of informatics, the term entity is used while in the context of the conceptual organisation

Term	Definition	Examples
Entity	Used to refer to a phenomenon that cannot be subdivided into like units. Used in the context of the conceptual organisation of the phenomena.	A house cannot be subdivided into houses; it can be split into rooms only. Referenced by a single identifier.
Object	Used for referring to a phenomena in the context of its digital representation.	The terms 'entity' and 'object' are context-dependent, and used interchangeably.
Feature	Refers to a phenomenon represented on maps.	
Attribute	Defined characteristic of an entity.	Type of bridge / highway, size of a lake, or type of a hotel.
Instance	A particular occurrence of a given type of an entity.	Engineering college or St. Joseph's High School
Spatial type	Point, line, area, or expression for a phenomenon.	
Attribute class	Category of entities with similar attributes	Over-bridges, students, schools.
Attribute value	Particular measure of an attribute for an instance of an entity	Number of students, size of a car.
Entity class	Category of entities with some similarity	All *educational* institutions, all *diesel* cars

Table 2. 1. Definitions of terms applicable to spatial data

of phenomena, and the term **object** is used while referring to the digital representation of the phenomena. In other words, the terms 'entity' and 'object' will be used interchangeably, depending on the context. The term **feature** is slightly more confusing, and is used while referring to phenomenon represented on maps.

An entity may not change in any regard except for its position, for example, a realignment in a highway Positional information is subject to limitations in accuracy, depending on the instruments and methods used for recording the information. Positional information is often used as a basis for spatial properties, like shape, area, or volume; but some spatial characteristics like adjacency, and containment do not require absolute positional data. Shapes are not clearly defined over time—rivers, lagoons, or islets, are examples.

An **attribute** is a defined characteristic of an entity, for example, the type of bridge (suspension, concrete, *et cetra*), type of highway (concrete, four-lane, divided, *et cetra*), category of hotel (five-star, two-star, *et cetra*), and so on Definitions for some more of the terms used in the context of spatial data are shown in Table 2.1.

2. 1. 2. Dimensionality of Entities

In principle, the point, line, area, and volume descriptions refer to the zero, one, two, and three dimensions of geometric figures. The idea of zero, one, or two dimensions is an artefact, for in reality everything is three-dimensional. A dot on a map has size. On the other hand, the dimensionless concept of a point for a city will, on a larger scale (1:200,000 km small scale; 1:24,000 km large scale) of map representation, become an area object. This **scale** of observation or mapping will affect what might be represented in a database. Point entities, defined for a given scale of observation, will appear in a relational database as separate lines of information in a table, with two items of information for position, one or two items for identification and other attributes of interest.

Line entities (also termed polylines, arcs, or edges) in a simple view of phenomena such as railways or roads are unrelated to each other in any way These linear features have the spatial properties like *length* (length of rivers), *sinuosity* (river bends), and *orientation* (compass direction).

Area entities (also referred to as polygons, regions, or zones) may be natural entities like lakes, islands, territory with a particular soil types, *et cetra*. Boundaries may be unclear, multiattribute, changing in time, and may not be directly observable.

Some of the properties associated with area entities are: *areal extent* (size of a lake), perimeter length, being isolated or connected to others (separate elements of Andaman Island or the contiguity for countries of the Middle East, or the Mahe exclave of Pondichery).

2. 1. 3. Referencing Discrete Entities

Features may be located as a relative one-dimensional distance from some named origin. Location of objects or events on highways are often referred to as an offset distance from a specific origin, usually the beginning of a highway, an intersection, or a national boundary. See Figure 2.1.

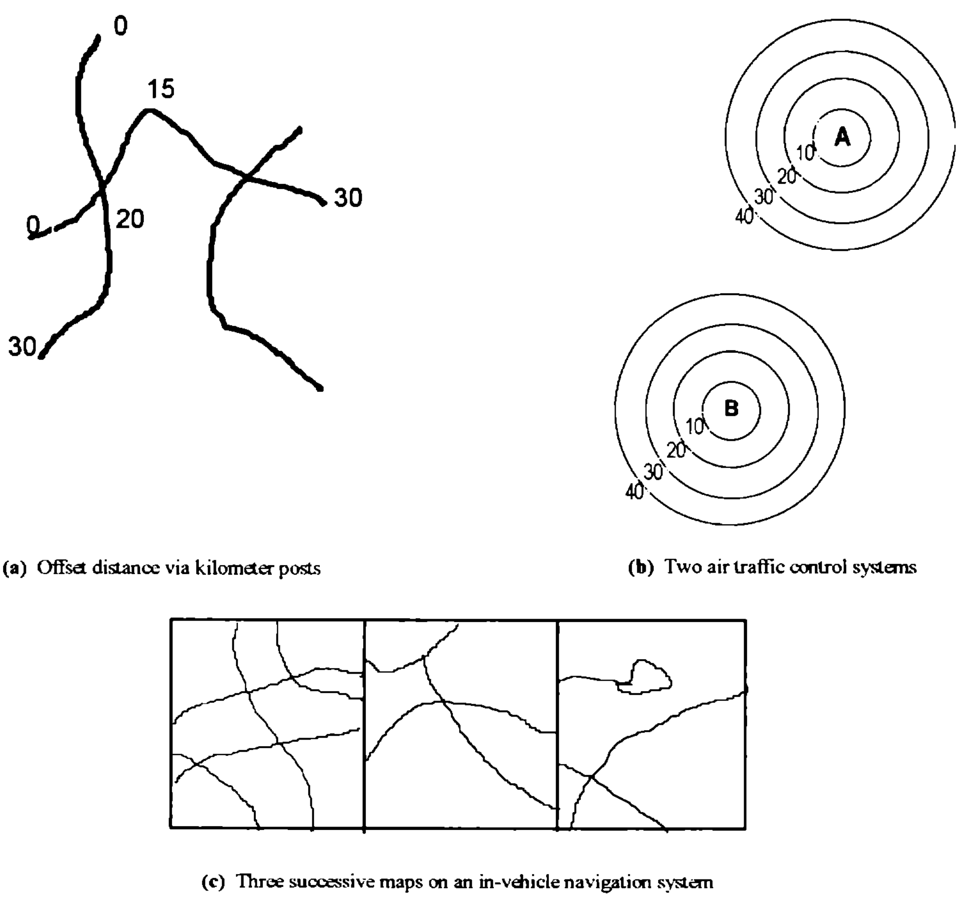


Figure 2.1. Different Origins for Spatial Referencing Systems.
(a) Offset distances; (b) Azimuthal maps; (c) Moving origin

Positional data may be obtained in the field by surveys, by coordinate geometry computations or by satellite surveying. Traditionally, some earth features especially property boundaries, were located by **metes** and **bounds** descriptions. Features were positioned relative to others by distance and bearing, even if the base points for measurement were not too fixed, like trees. Today, space-borne transmitters, known as global positioning system (GPS), are increasingly used to pinpoint places on the earth. GPS allows accurate spatial referencing in continuous space to a precision of few metres, or about the width of an average city street. It should be possible to give every hectare of the earth a unique address. It still remains impractical, however, to collect data for the attributes of such spatial units.

2. 1. 4. Line Simplification

Many cartographic techniques are being applied to reduce the volume of data within some constraints, while preserving the essential shape of the features. The most well-known procedure, used in several commercial software systems, perhaps is the Douglas-Peucker procedure [Douglas & Peucker 1973], [Chrisman 1997].

The procedure works line by line, assuming that the end points are required in the final form of the line. A trend line is constructed by joining the end points. The points between the end points are examined to find deviations of each point from the trend line. Those points whose deviations are less than a chosen threshold are then eliminated. The point having the greatest deviation is then chosen as the next floating point. A new trend line is drawn from the first end point and the new floating point. The process is repeated until there are no points closer to a line than the threshold.

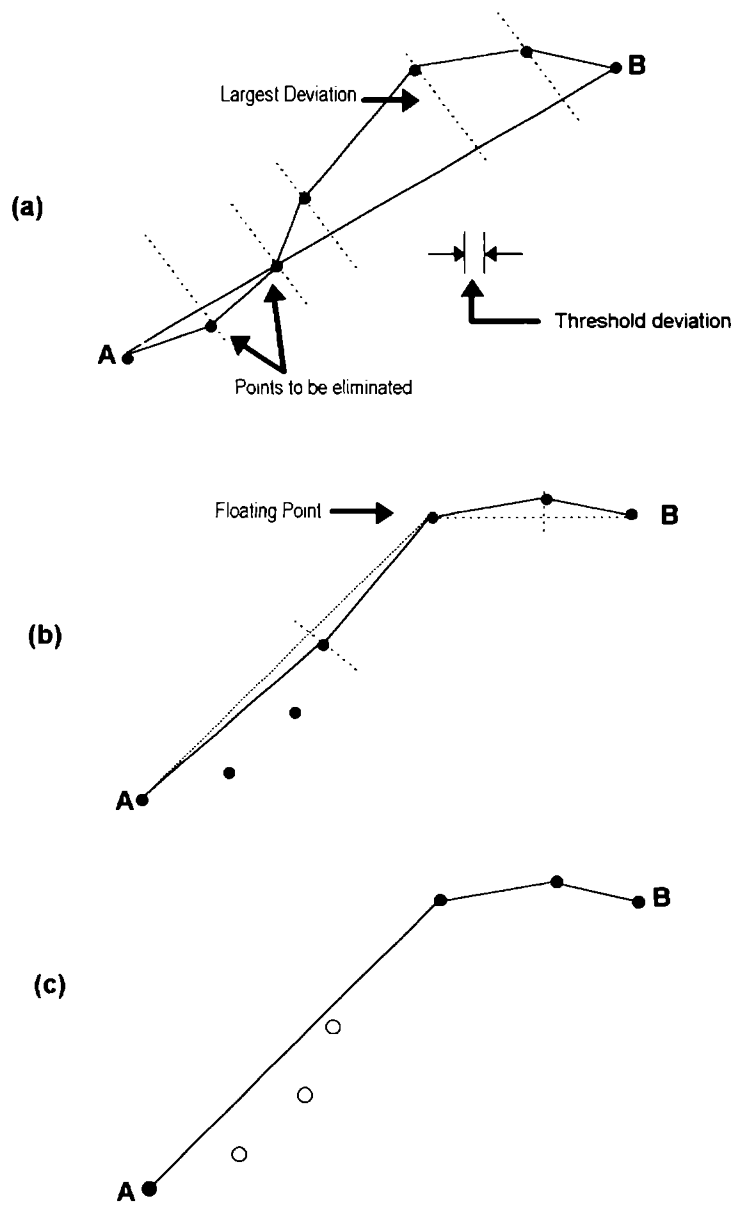


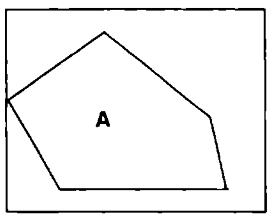
Figure 2. 2. Douglas-Peucker Algorithm. **(a)** The original line with 7 points. **(b)** Three points eliminated after the first processing. **(c)** The final form of line with only 4 points.

Any line reduction technique will fail if extended beyond its intended purpose. The Douglas-Peucker procedure works line by line, and hence does not try to alter the fundamental topology.

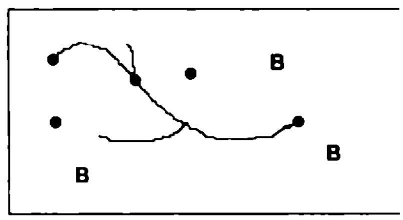
2. 1. 5. Area Representation

The concept of digital line graph was introduced by the United States Geological Survey in the late 1970, for preparing the digital encodings of topographic maps. The phenomena contained in those maps were viewed as point, line, or area objects. The following kinds of areal phenomena had to be distinguished [Burroughs 1986]:

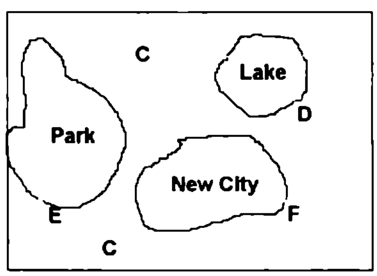
Polynomial units of data such as administrative areas (Figure 2.3 (a).
The area forming a background to point or linear objects (Figure 2.3 (b).
The background for polygon spatial units (Figure 2.3 (b).



(a) Area A is a polygon spatial unit



(b) Area B forming background to point or linear objects



(c) Area C is background for areal objects D, E, F

Figure 2. 3. Types of Spatial Units Handled by Digital LineGraph

2.1.6. Topological Consistency

A topologically consistent map (or database) may be defined as having all spatial entities projected upon a plane surface, with no freestanding features, and having complete topology. Not all features on maps are topological. The curvature of lines, the shape of polygons, and the labels of places are not topological. Several different maps may have the same topological structure. Shape and relative placements of points may differ; but relative placements, the conditions of connectedness, and adjacency should not vary. Lengths and angles may change but four elements must not be altered. These four elements, referred to as the map properties, are:

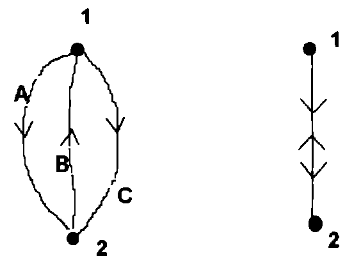
Incidence (two nodes per line or lines at nodes)
Intersections
Adjacencies (neighbours for area units)
Inclusions (points in polygon)

The four map properties, in other words, are invariant under countinous deformation of the map base. Tearing and puncturing can alter some of these in particular cases, but stretching will not do so. This aspect is exploited for substantiating checks on the data integrity during map trasformations. In a topologically consistent map, completeness of inclusion means that there are no isolated non-connected points and that all lines are part of boundaries of polygon. The completeness of incidence means that lines intersect only at points associated with the ends of lines.

Consider the case of two polygons X and Y touching at a common edge, and having two other topological one-cells (lines), as shown in Figure 2.4 (a). Each of the three lines has two nodes and two polygons, including the exterior region O. But if we had only zero-cell information, the graph would consist of three arcs as shown in Figure 2.4 (b), or all coincident, as shown in Figure2.4 (c). Working with only polygon boundary lines (two-cell data) again produces ambiguty (Figure2.4 (d)). A two-cell has a minimum of three edges and three vertices. The complete set of data

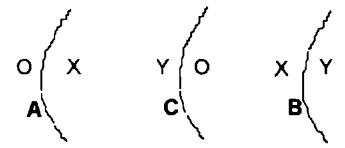
Line	Node From	Node To	Polygon Left	Polygon Right
A	1	2	X	O
B	2	1	X	Y
C	1	2	Y	O

(a) Data for three one-cells



(b) Graph for three one-cells

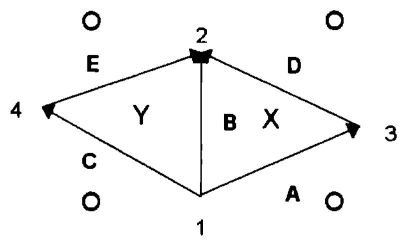
(c) Graph for three one-cells with coincident lines



(d) Only two-cell data used. No node data used. Ambiguous situation

Line	Node From	Node To	Polygon Left	Polygon Right
A	1	3	X	O
B	1	2	Y	X
C	1	4	O	Y
D	3	2	X	O
E	4	2	O	Y

(e) Complete data for five edges and four nodes



(f) Graph for data in (e). Topologically consistent

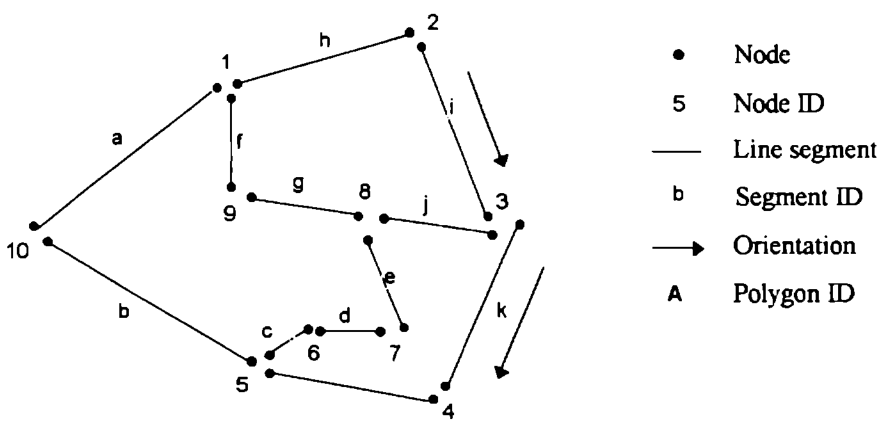
Figure 2. 4. Topological Consistency

must have records for five edges and four nodes. A complete and unambiguous data could be as shown in Figure 2.4 (e), with the corresponding graph as in Figure 2.4(f).

2.1.7. Data for Spatial Relationship

It is important to recognise that some geometric or topological properties cannot be directely encoded; they will have to be derived from other auxilliary data. It is thus impotant to identify what topology is explicit, that is to say, what can be extracted from tables of numbers or other similar sources of data.

The first topologically explicit data model was the **line segment** structure, devised and used by the U.S. Census Bureau for the 1970 census. This had a Dual Independent Map Encoding (DIME) format. In this fomat, a basic record had line segments with from-node, to-node, left polygon, right polygon, and associated information like the coordinates of the nodes, and polygon identifiers. There were no intermediate points, and consequently, a single record could conveniently handle both geometry and topology. One inconvenience posed by this is that major turning points along a road had to be reckoned as nodes as well as road intersection. Further, the features for entities like polygons had to be assembled from the data encoded in the line segment records. See Figure 2.5.



Line	Node From	Node To	Coordinates From	Coordinates To	Polygon Left	Polygon Right
a	1	10	7, 14	1, 10	B	
b	10	5	1, 10	6, 3	B	
c	5	6	6, 3	7, 4	B	C

Figure 2. 5. Explicit Line Segment Encoding of Topology.

The duality for nodes in a DIME (from-node and to-node)structure provided a base for moving through a network by searching for links with matching node numbers. Even if the links and points do not have unique labels, matching can be made with the coordinates of the end-points of edges in the graphs. TIGER, the system used by the U.S. census bureau for its 1990 census adopted this scheme

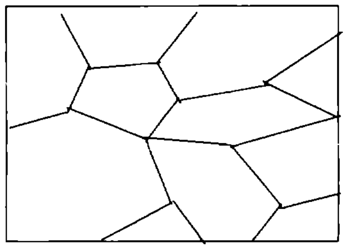
In a **geometric model**, entities can be manipulated completely, in one go, not having to assemble them from seperately connected pieces. A long national highway can be identified and stored as a single linear feature rather than requiring retrievel of a long set of seperate line segments created by the topological junctions at intersections.

Topologically structured data can be manipulated algebraically to produce valuable statistics, for revealing surface properties, or for facilitating error detection. Many people favour the **topology model**, because it directely addresses entities in a 'what-exists' basis. Topological structures avoid redundancy when storing common boundaries and they are very suitable for shortest path computations in a network [Oosterom & Vijlbrief,1994].

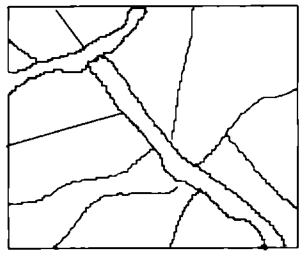
2. 1. 8. Tessellations

The discretization of space takes many forms. Tessellations are sets of connected discrete two-dimensional units. A regular tessellation (e.g., grid squares) is an infinitely repeatable pattern of a regular polygon or polyhedron (three-dimensional). This cellular decomposition implies that every point in space is assigned to only one cell. An irregular tessellation is an infinitely extending configuration of polygons or polyhedras of varied shape and size. Regular tessellations are used in

- Image data from remote sensing
- Data compilation from maps by grid squares
- Uniform sampling of a continuous spatial distribution.



Thiessen Polygons



Land Parcels

Irregular

Low-resolution satellite data

Regular

Figure 2.6. Irregular and Regular Tessellations

Irregular tessellations are used in areas such as

Partitions of a large geographic database
Land parcels
Zones for economic and demographic data

Tessellations are not only to be seen as spatial units for recording data, but also as devices for facilitating access to databases for countinuous space. Of the many regular geometric figures, the squares, triangle, and hexagon are those most likely to be encountered in spatial data contexts. The advantages of the square is related to the popularity of the graph paper and mechanical printing devices. The hexagon has utility in theory and practice where adjacacency is an important property or where packing of geometric figures into space is important. The triangle may be considered as the most primitive polygon, since it cannot be further subdivided into another figure, and also because it can take on different shapes based on angles or length of sides. Regular polygons are characterised by identical lengths of sides and equal angles subtended by

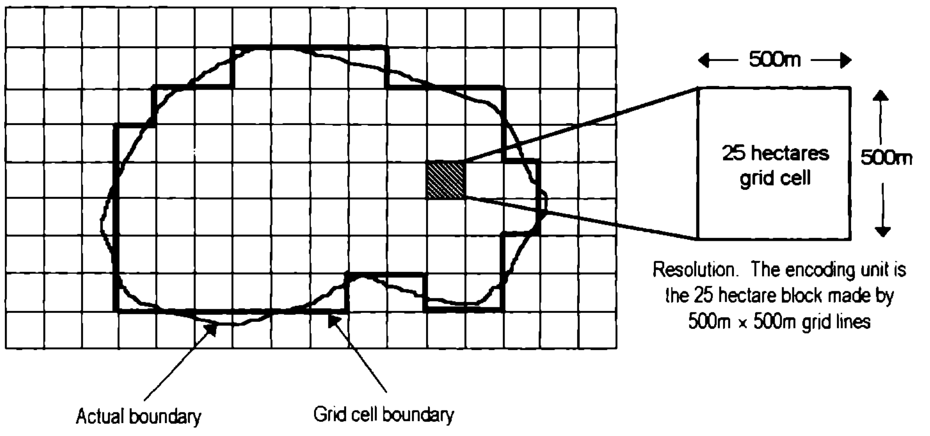


Figure 2.7. Grid Cell Data Encoding

lines from vertices to the centre of the figure; and only the triangle, square, and hexagon regular figures, unlike circles, can fill a plane completely.

The common form for structuring regular tessellation data is a basic tabular record of data value, a column coordinate value, possibly a cell identfier, and a series of attribute values for the cell. If a high degree of precision is *not* required for capturing data, the amount of data can be reduced by having an entire grid cell to indicate that a piece of highway or boundary exists in part of the cell (Figure 2.7).

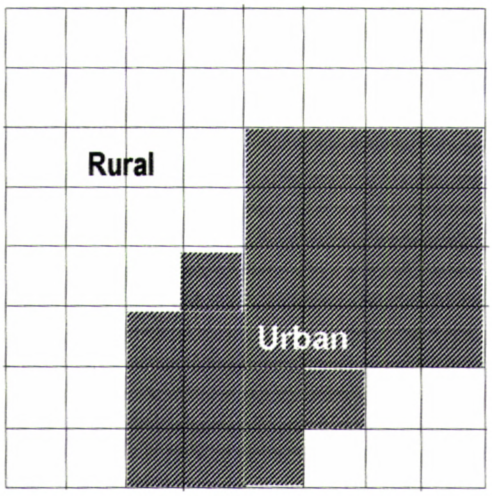
Spatial aggregation can be achieved for tessellation by combining adjacent cells. Aggregation becomes necessary to facilitate generalization from one resolution to another. The number of points in four cells can be added to get the incidence for a set of four cells, or, if the attribute are scalar, an average could be assigned to the combined cells. Spatial deaggregation is dealt with by splitting cells.

2. 1. 9. Variable Spatial Resolution

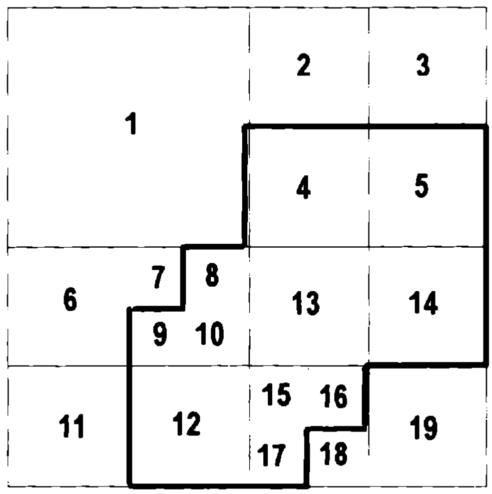
The concept of **variable spatial resolution** is employed in GIS when regular spatial units are involved. The phenomena represented by a space may be aggregated into square blocks of the same or different sizes. Consider the distribution of houses, each represented as a dot. Aggregating these to rectangles containing an equal number of houses produces units of different sizes (of rectangles). The concept of variable spatial resolution implies varying-size units at a given resolution level. Even though the choice of the shape of units should not be a prime concern, the square is particularly handy due to the smooth process of creating blocks of smaller size by dividing a square into four squares, or of larger size by aggregating four adjacent squares into a larger square.

The region can be successively approximated by sets of blocks at different levels. If the process involves systematic splitting of space in two dimensional space by a rule of four, then the structure is known as a **quadtree,** which is one type of hierarchical data model. A three-dimensional equivalent is known as an octree because it involves an eight fold splitting.

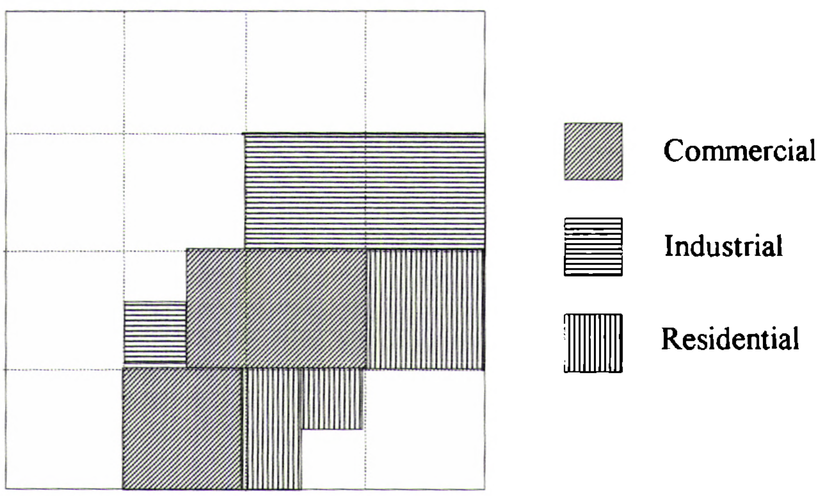
The region quadtree is usually viewed as a tree of out-degree four which divides a $2^n \times 2^n$ image into equal sized quadrants, subquadrants,..., until homogeneous blocks (possibly pixels) are obtained. Consider the region shown in Figure 2.8 (a) which is represented by a $2^3 \times 2^3$ binary array showing an urban area (shaded) located contiguously within a rural area. The maximal blocks resulting from the region decomposition are shown in Figure 2.8 (b), using the quadtree representation. The different types of land use is demarcated in Figure 2.8 (c). The corresponding



(a) A region with binary attributes (rural or urban)



(b) Block decomposition of the urban region in (a). Numbers indicate full quadrants or subquadrants as per the sequence, NW, NE, SW, and SE of **(a)**



(c) Types of land use in the urban sector

Figure 2. 8. A region showing an urban area, contiguous within a rural area. (a) Region represented by a binary array.
(b) The maximal blocks, numbered as per the quadtree scheme.
(c) Demarcation of the urban area based on a hypothetical land use.

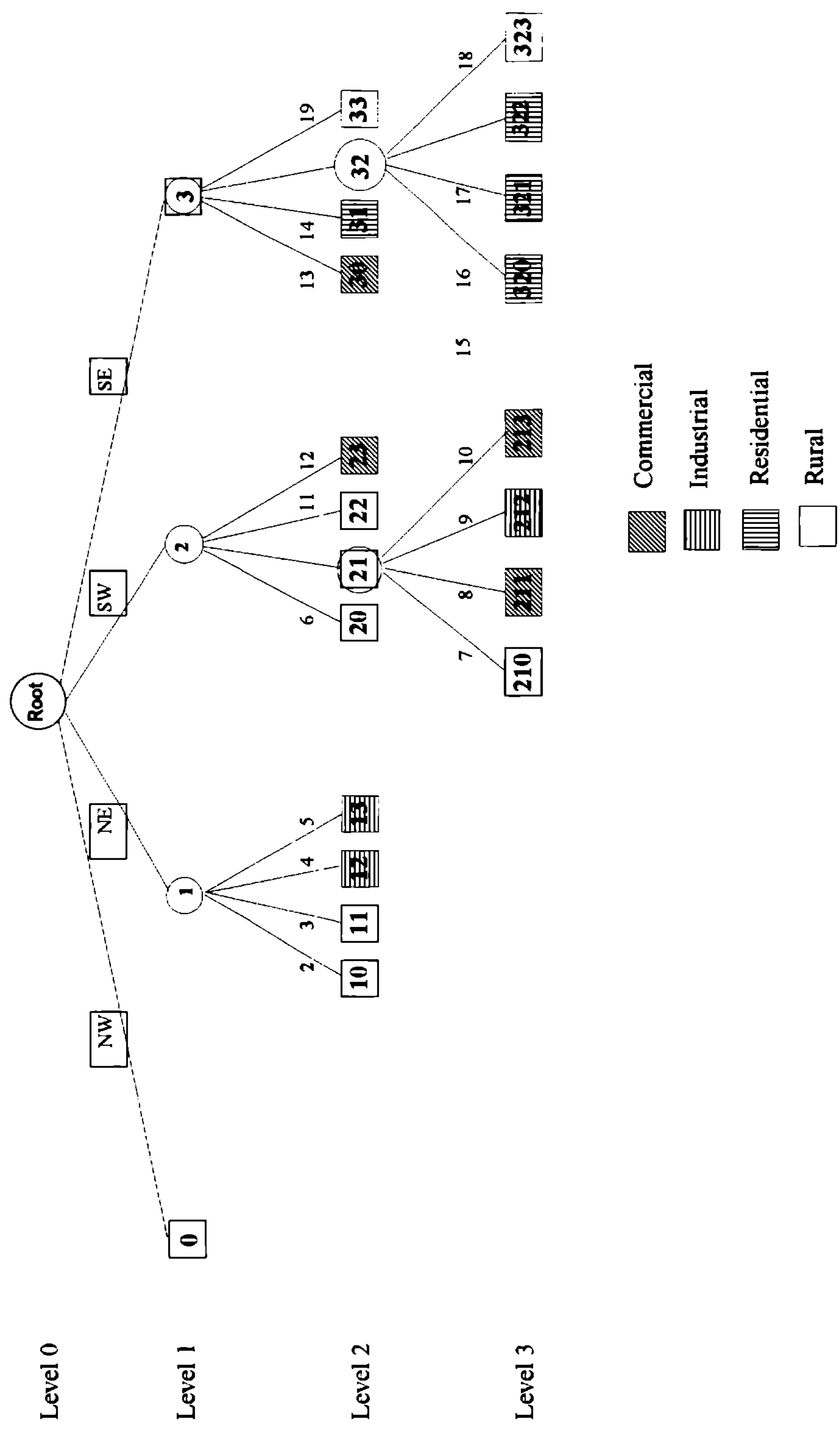


Figure 2. 9. The Quadtree Corresponding to Figure 2.8 (c).

hierarchical arrangement of the blocks and sub-blocks is shown as a quadtree in Figure 2.9. The four blocks (NW, NE, SW, and SE), produced at each level of subdivision can be seen, with the levels marked. The shaded area of Figure 2.8(a), as pointed out earlier, indicates urban area; and this attribute data condition is encoded by the set of quadrants at levels 2 and 3 of Figure 2. 9.

Area computations can be made by counting quadrants of particular conditions and multiplying by the size of the quadrant. Real area unit measures may then be

Level 1	Level 2	Level 3	Land Use	Area[1]
0			Rural	16
1	10		Rural	4
	11		Rural	4
	12		Industrial	4
	13		Industrial	4
2	20		Rural	4
	21	210	Rural	1
		211	Commercial	1
		212	Industrial	1
		213	Commercial	1
	22		Rural	4
	23		Commercial	4
3	30		Commercial	4
	31		Residential	4
	32	320	Residential	1
		321	Residential	1
		322	Residential	1
		323	Rural	1
	33		Rural	4
		Total Area	Commercial	10 Blocks
			Industrial	9 Blocks
			Residential	7 Blocks
			Rural	38 Blocks
		Total Area		**64 Blocks**

[1] Area is indicated as number of level 3 - blocks

Table 2. 2. Table of Data for the Entire Region in Figure 2.9 (b)

obtained by multiplying by a map scale factor. If, for example, the square region shown in Figure 2.8 (a) has a side measuring 800 metres, the total area (urban and rural) will be 64 hectares. Each small squares at level 3 in Figure 2.9 will then measure exactly a hectare.

A complete listing, including the space block identifiers, appears in Table 2. 2. The column under the heading 'Area' shows the area represented by each quadrant. Based on the discussion in the previous paragraph, the unit of area computed in Table 2.2 will be a hectare per block. For the illustration, a numerical location code has been chosen wherein, the sequence of NW, NE, SW, and SE blocks are represented by the numbers 0, 1, 2, and 3. Thus, the code 213 stands for the SW corner at level 1, NE corner at level 2, and the SE corner at level 3

Subdivision continues until a predetermined limit of resolution is reached, or the required detail in the phenomenon is accounted for If the source document contains much curvature detail for linear features or polygon boundaries then, as a practical matter, the decomposition process will be terminated at a chosen resolution, because of computer resource limitations. The hierarchical organisation allows spatial resolution to vary with the complexity of the phenomena being recorded.

In contrast to the regular subdivisions involved in the region quadtrees discussed above, hierarchical decomposition can be made on the basis of the empirical information to be encoded. While the regular subdivision scheme is data-independent, the latter is data-dependent. The point quadtree was proposed by [Finkel & Bentley

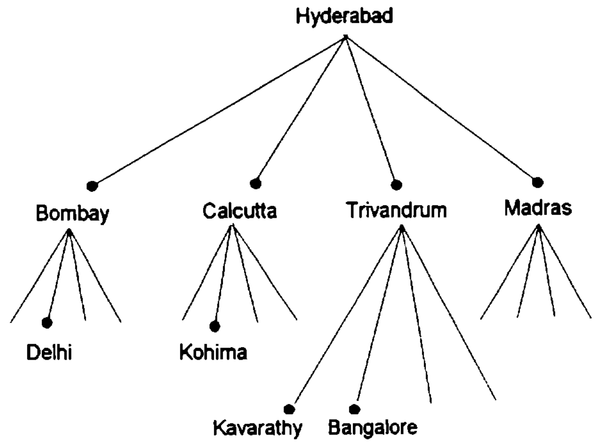
1974] to represent data points in a multi-dimensional space. It can also be viewed as a multidimensional generalisation of the binary search tree [Knuth 1975], [Samet 1984].

In two dimensions, each data point is a node in a tree having four sons, which are roots of subtrees corresponding to quadrants labeled in the order, NW, NE, SW and SE. The process of inserting records into point quardtrees is analogous to that used for binary search trees. We search for the desired record on the basis of its *x*- and *y*- coordinates. At each node of the tree, a four-way comparison operation is performed and the appropriate subtree is chosen for the next test. Reaching the bottom of the tree without finding the record means that it should be inserted at this position. The shape of the tree depends on the order in which records are inserted into it. The tree in Figure2.10(b) is the point quadtree for the sequence, Hyderabad, Trivandum, Kavarathy, Bangalore, Madras, Bombay, Calcutta, Kohima, and Delhi. Deletion of nodes is more complex [Samet 1980]. The efficiency of the point quadtree lies in its rôle as a pruning device on the amount of search that is required. If we wish to find all cities within five units of a data point with coordinates (75,10), there is no need to search the NW, NE and SW quadrants of the root in figure 2.10(c) viz., Hyderabad with coordinates (40,43); the search will be restricted to the SE quadrant of the tree rooted.

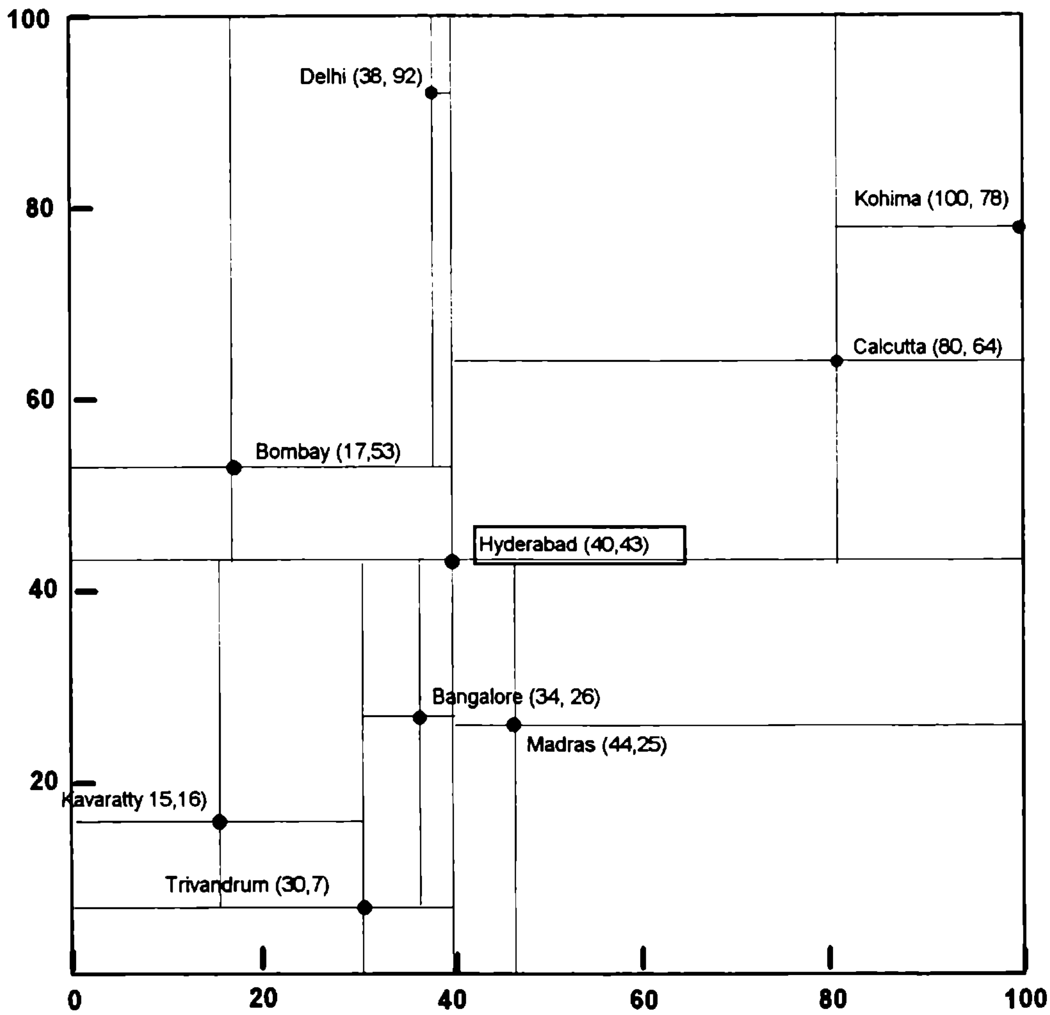
The problem when the number of dimensions become more than 2 is that the branching factor becomes very large (*i.e.*,2^k for k dimensions), thereby requiring much storage for each node as well as many NULL pointers for terminal nodes. The *k*-d tree of Bentley [Bentley 1975] is an improvement on the point quadtree that

City	X	Y
Hyderabad	40	43
Trivandrum	30	7
Kavarathy	15	16
Bangalore	34	26
Madras	44	25
Bombay	17	53
Calcutta	80	64
Kohima	100	78
Delhi	38	92

(a) Data for Selected Cities



(b) Point quadtree



(c) Planar Representation

Figure 2. 10. Point Quadtree

avoids the large branching factor At each node level of the tree a different coordinate is tested for determining the direction (NW, NE, SW, or SE) in which a branch is to be made. In a two- dimensional case (2-d tree), the *x*- coordinate is compared at the root (0 level) and at even levels, and *y*- coordinates at odd levels. Figure 2.11 is the *k*-d tree corresponding to the point quadtree of Figure 2.10, with the same order of records.

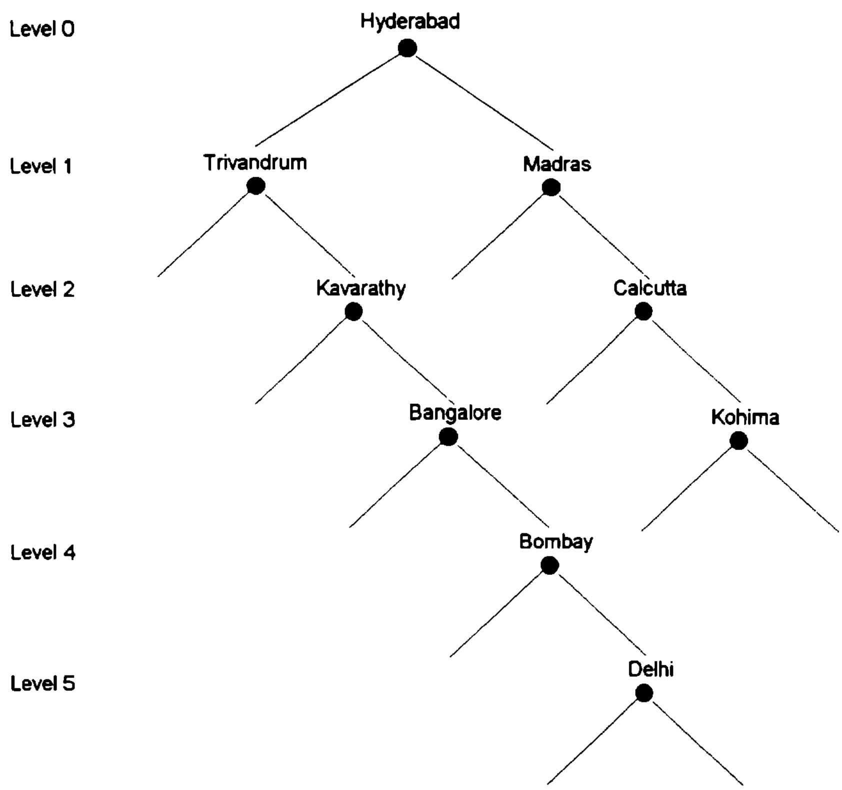


Figure 2. 11. A *k-d* Tree corresponding to Data in Figure 2.10 (a)

In general, *k*-d trees are superior to point quadtrees, with one exception. The point quadtree is an inherently parallel data structure and thus the comparison operation can be performed in parallel for the *k* key values, which is not the case for the *k*-d tree. In short, the *k*-d tree can be characterized as a superior serial data structure and the point quadtree as a superior parallel data structure [Samet 1984].

2. 2. SPATIAL DATA MODELS

FROM the point of view of a casual spatial data user, the terms *data model*, *data structure*, and *file structure*, are often used with the same connotation. From a GIS point, however, the expression **data modelling** is used to denote a comprehensive set of conceptual tools for organising data in the spatial context. Further, data models provide frameworks for information systems that are independent of the particulars of one implementation or another They are at a generic level, extending to the designers and users the basic constructs for building the system.

Integration of spatial data with other kinds of information (e.g., tabular data) involves some formal definitions and classification of spatial relationships and queries, the development of efficient data models and structures for representing spatial data, and the design of efficient algorithms for solving geometric problems involved in answering spatial queries.

Topological relations encode the connecting structure of a map. They can be classified into adjacency, connectedness, boundary, and co-boundary relations among the basic spatial entities. Set-theoretic relations are based on concepts like inclusion, intersection, coincidence, element-of, *et cetra*, and they encode spatial interferences between entities, even from different maps. Metric relations involve the concept of distance and encode spatial proximity [Floriani *et al* 1993].

In an object-oriented approach [Egenhofer *et al.* 1989], [Worboys 1992], representation of spatial entities is independent of their position in space. The entities have distinct identities and are self-contained, each having all spatial attributes described explicitly and exactly (a point by its location, a line segment by its equation and the end-points, a region by its border, *et cetra*). When topological relations between entities are also stored, we have a topological model that provides a complete structural representation of a map.

The literature about formal models for representing and reasoning about spatial information is quite large. The representation of spatial relations, that is the interplay between geometric objects, is one of the key issues in GIS, as pointed out on pages 21 and 23 Topological relations, specifically region-region relation is the thrust of the works of [Egenhofer 1993], [Clementini *et al.* 1995], and [Clementini & Felice 1995]. The formal model for the combination of topological knowledge and the derivation of compositions of binary topological relationships proposed by Egenhofer, is based on concepts of point-set topology with open and closed sets. Topological relationships between two point sets are defined through intersection relations of their boundary, interior and complement [Abdelmoty *et al.* 1993].

An alternative approach to spatial relationship representation is based on the first order formalism for reasoning over space, time, and processes, presented by [Cui *et al.* 1993]. This approach is based on a primitive notion of connection. The primitives include regions where every region coincides with a set of incident points, and is contained in a unique region called the universe. Unlike the Egenhofer model, the basic part of the formalism assumes one primitive dyadic relation C(x,y) read as 'x connects with y', from which a basic set of dyadic relations are defined. Since the connectivity relationship can be computed for the whole space, a systematic derivation of the whole set of specialised relationships can be effected without the application of computational geometry algorithms [Abdelmoty *et al.* 1993].

2. 3. SPATIAL DATABASES

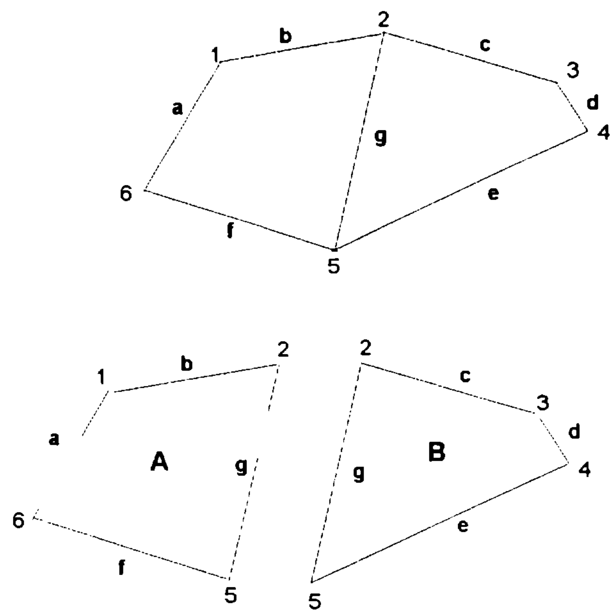
COMPUTER science has seen many generations of data management, starting with index files, and later, network and hierarchical database management systems (DBMS). Subsequently, relational DBMSs (RDBMSs) took the industry by storm by providing *data management* capabilities based on much simpler concepts.. A data management problem may be considered to be three-dimensional in its scope, consisting of **data, object**, and **knowledge** management services. Of this, only the first dimension, viz., the "data management service" is involved in the common business data processing applications where large numbers of instances of fixed format records have to be stored, accessed based on suitable queries, and processed; and RDBMS has been capable of handling them, effectively Object management problems involve non-traditional data types such as bitmaps, icons, text, polygons, and multimedia data. CAD, GIS, and image processing applications come under this category, and RDBMSs have proven inadequate for these areas. Knowledge management entails the ability to store and manipulate non-traditional data types, access and manipulate them based on a collection of rules that are part of the semantics of an application. These rules describe integrity constraints about the application; they also allow extraction of derivatives of data that are not explicitly stored in the database. Examples of situation where all three dimensions are called for in the solution are detailed in [Stonebraker & Kemnitz 1991].

Emergence of new applications has spawned a new generation of database systems. Object-oriented database management systems (OODBMS) are based on the fundamental concept that each real-world entity can be modelled by an object, defined appropriately

2. 3. 1. Spatial Data Models and RDBMS

(a) The Territory

(b) Parcels A and B, their segments and corners



(c) Data tables for the attributes and geometries of the parcels.

Territory

Record No.	ID	Scale
1	East	50,000

Parcels

Record No.	ID	Size
1	A	25
2	B	17

Segments

Record No.	ID	Length
1	a	5
2	b	5
3	c	4
4	d	1
5	e	7
6	f	4
7	g	8

Corners

Record No.	ID	X	Y
1	1	0.1	1.2
2	2	1.1	1.2
3	3	2.0	1.3
4	4	2.1	1.1
5	5	1.0	0.0
6	6	0.0	0.4

Figure 2. 12. A Hypothetical Territory, Split into Two Parcels, A and B, Their Attributes, Geometries, and Tables of Spatial Data

A given territory of Figure 2.12 (a) may be considered to be made up of two parcels of land, A and B. The edges and corners are indicated in Figure 2.12 (b). The relevant data for these parcels are shown in the four tables of Figure 2.12 (c). A relational representation of these data would take the form as shown in the set of three tables of Figure 2.13. The tables, among themselves, can be connected by the fields Vertex ID and Parcel ID.

Segments Table

Segment ID	Length	Vertex ID Begin	Vertex ID End	Parcel
a	5	6	1	A
b	5	1	2	A
c	4	2	3	B
d	1	3	4	B
e	7	4	5	B
f	4	5	6	A
g	8	5	2	A
g	8	5	2	B

Parcels Table

Parcel ID	Size
A	25
B	17

Corners Table

Vertex I	X	Y
1	0.1	1.2
2	1.1	1.4
3	2.0	1.3
4	2.1	1.1
5	1.0	0.0
6	0.0	0.4

Figure 2 .13. Relational Representation of Spatial Data in Figure 2.12

In general, relational databases are popular because of their simplicity of data organisation. The user can join tables as needed; by properly designing the tables, storage efficiency can be improved. Queries can be expressed in terms of what is needed. For some purposes, however, the network model can deliver better performance, although it is necessary to know the paths through the structure for access to particular information.

Currently, most of the GISs available in the market are built over one of the standard relational database management systems (RDBMS). While the RDBMS manages the entire set of classical data types (facts), a proprietary file management system handles the geometrical and/or topological data, while a complex pointer mechanism translates links between facts and geometry. In such a model, known as 'hybrid model', as the file-manager is a single-user, the system becomes a single user.

The systems architectures of most commercial GIS packages may be categorised as one of the following three types: dual, layered, or integrated [Larue *et al.* 1993].

2. 3. 1. A. Dual Architecture

The architecture is shown in Figure 2.14. The integration layer is necessary to process a request linking spatial and aspatial data for some objects. Data integrity is ensured only at the integration level. ARC/INFO (*vide* Page 18) is designed along this scheme, and called "geo-relational" scheme by the Trade Mark holders of ARC/INFO [ESRI 1994]. GeoQL [Ooi 1990] which provides an extended SQL interface at the integration layer can also be considered as belonging to this category

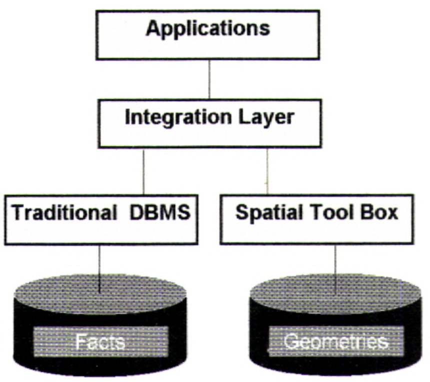


Figure 2. 14. Dual Architecture Scheme of GIS

2. 3. 1. B. Layered Architecture

An improved GIS architecture would result by having a layer built on top of a conventional RDBMS, as shown in Figure 2.15. Geometries are stored in the form of long binary attributes. In contrast to the two separate data managers involved in the dual architecture described above, in the layered architecture, both the spatial and aspatial data are stored in the same RDBMS. A layer of translation software, introduced between the query processor and the RDBMS, facilitates operations on the spatial data. Examples of GIS that employ this architecture are GEOVIEW [Waugh & Healey 1987], and SIRO-DBMS [Abel 1989].

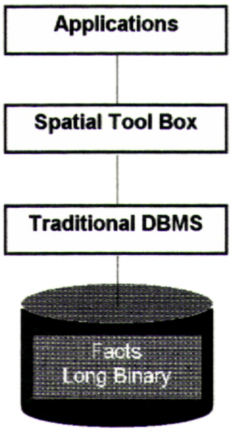


Figure 2. 15. Layered Architecture Scheme of GIS

Spatial semantics of geometric attributes is unknown to the RDBMS which sees these attributes as binary strings. The corresponding operators expressing these relationships must be implemented in the upper layer, far from the database. As a result, the spatial data of query language is more a primitive-call interface than a true language.

2. 3. 1. C. Integrated Architecture

The integrated architecture employs an Extensible Database Management System (EDBMS). These systems accept user-defined data types, or abstract data types (ADT) which provide a complete DBMS with 'hooks' for extensions to be added to the DBMS. These hooks permit the registration of operators and functions that extend the query language and, of access methods that speed up the execution of some of these operators. POSTGRES [Stonebraker *et al.* 1990], [Stonebraker 1995], and Starburst [Lohman *et al.* 1991], [Widom 1996] (*vide* Page 20), are EDBMSs that provide extensibility at one or more levels of the system. POSTGRES provides some geometric data management capabilities. It has an integrated R-tree spatial access method, and it supports the built-in data types: *point, line-segment, path*, and *box*. Some simple operators are also provided such as *box-overlap, point-inside-box,* and *distance.*

One major drawback of the relational data model is that information is scattered. Implementation of normalisation requirements tends to allocate data to small relations, thereby dispersing all information concerning a particular entity The ideal would be to provide an isomorphy *i.e.*, a direct correspondence between real world entities and their computer representation. OODBMSs, which allow large amount of persistent data to be shared by many users, and which model each real-world entity by an object, are gaining closer attention as an alternative to RDBMSs.

2. 3. 2. Spatial Data Models and OODBMS

The data models of OODBMSs provide many new features not present in an RDBMS:

> *Objects:* Data that might span many new tuples in a RDBMS can be represented as one data object.
> *Procedure encapsulation:* Procedures as well as data attributes can be stored with objects in the database, and these procedures can be used as *methods* to encapsulate object semantics.
> *Links:* Relatioinships between objects can be represented more efficiently than in RDBMS implementations, using a more convenient syntax than relational joins.
> *Multimedia data:* Objects that contain large amounts of audio, video, or text can be retrieved and edited more efficiently.
> *Type inheritance:* Object types or classes, can be organised as a hierarchy with inheritance properties, thereby greatly simplifying the description of complex data schemes.

In addition to the data model features for supporting more complex data, most OODBMSs also provide the following two important architectural features:

1. They are designed for higher performance for operations on objects in the new applications. Some can approximate the operational speed of programming languages when the data is brought into main memory.
2. In some OODBMSs, the programming language and data manipulation language have been combined as one language and type system [Cattell 1991].

In an OODBMS the existence of an object is independent of its value. While an RDBMS prohibits two records to have identical in values, two objects in an OODBMS may have equal values and still be identified unambiguously Each object is labeled by an object identifier (OID), created by the system, in order to guarantee system-wide uniqueness. It is not visible to the user and does not change during the lifetime of an object.

As seen earlier, an OODBMS is a DBMS with an OO data model. The OO data model provides a framework for a uniform treatment of arbitrary data types, including spatial data types. It makes it possible for the attribute definitions to be systematically reused. Even the application code can be stored in the database and systematically reused. The OO data model allows the domain of an attribute of a class (roughly corresponding to a table in a relational database) to be an arbitrary, user-defined class or a system-defined type (e.g., integer, string, *et cetra*).

It is not sufficient for a system to have just an OO data model if it does not have the full functionality of a DBMS. Specifically, this requires persistence, secondary storage management, concurrency control, recovery, and an *ad hoc* query facility. Optional features include data distribution and integrity constraints

The class City in Figure 2.16 has an attribute Wards, which is an enumeration of its wards, and, therefore, being set-valued (indicated by {}). The Shape of a ward is described by a set of edges. An Edge has two vertices V1 and V2, which are split up into X- and Y- components. These components are integers and, therefore, atomic. For a system to support composite objects, it should provide atomic types ,viz., integer, character, and certain type constructors viz; tuple, set, list, so that users can build application-specific complex object structures.

In an OO data model, similar objects may be grouped together into a *class*. The structure of a class may be defined by types as well, such that all values of a

City

OID	**Wards** (Set of Wards of the city)
Kochi	{Ward1, Ward2, ..., Ward45}

Ward

OID	**Shape** (Set of edges of the boundary)
Ward1	{W20,N11, E31, S6}

Street

OID	**Shape** (Set of edges of the street)
Main Street	{E10, E25, E30}

Edge

OID	V1		V2	
	X (integer)	Y (integer)	X (integer)	Y (integer)
E10	2	5	57	8

Figure 2. 16. Structural Object-Orientation Model

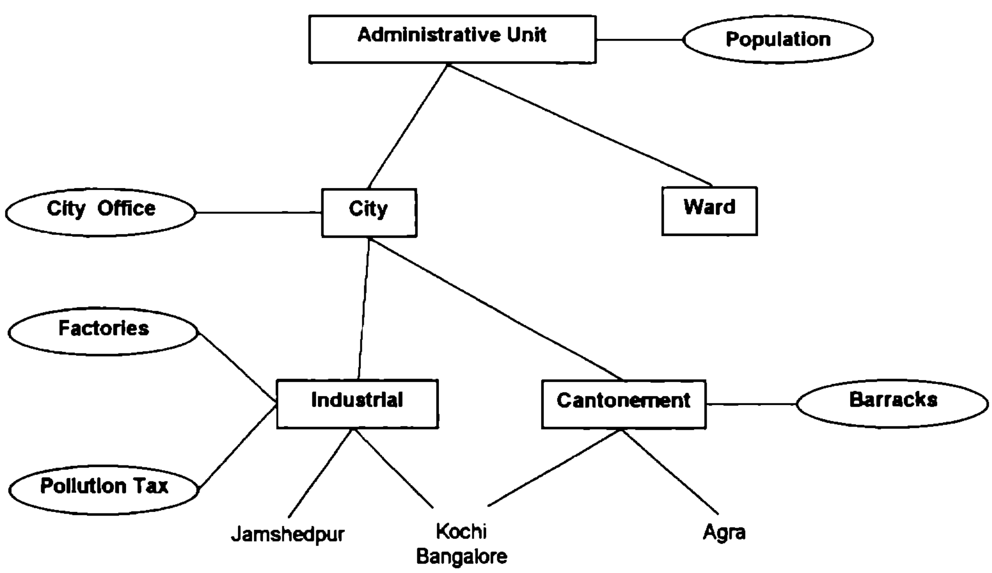


Figure 2. 17. Classes, Class Hierarchies, and Inheritance

particular type form a class. Classes may form a hierarchy, such that objects in any given class are automatically members of all its superclasses. An object class o inherits the attributes and methods of an ancestor class a if and only if they do not conflict with the attributes and methods that have been defined for o directly or for any ancestor class of o that lies between a and o.

Figure 2.17 illustrates a class hierarchy A class is represented by a rectangle. Ovals attached to rectangles indicate attributes that are defined on these classes. A City has an attribute CityOffice and an attribute Population inherited from its ancestor AdministrativeUnit. These attributes are inherited down to the descendants of the class City, viz., to Industrial (with specific attributes Factories and PollutionTax), and to Cantonement (with the specific attribute Barracks). An object may belong to multiple classes, in which case one requires a mechanism to resolve possible conflicts between various inherited attributes. For example, the cities of Kochi and Bangalore are industrial cities and both are cantonments, having therefore attributes of Factories, PollutionTax and Barracks, besides all the other attributes attached to City The city of Jamshedpur is an industrial city, therefore having the attributes Factories and PollutionTax, besides those attached to City.

The object-oriented paradigm was introduced primarily in the field of programming languages. Several object-oriented programming languages (OOPLs), such as Smalltalk, CLOS (an extension of Common Lisp), and C++ (an extension of C), came about, as a result of these efforts. The OODBMSs share many characteristics with OOPLs. Some OODBMSs, in fact, were defined starting from existing OOPLs. OODBMSs that are prominent today, can be grouped into two broad categories, depending on whether they evolved from programming-language or database-system architectures:

- O_2 [Deux *et al.* 1990], [Deux *et al* 1991], ObjectStore [Lamb *et al.* 1991], and GemStone [Butterworth *et al.* 1991], provide "database programming languages." That is, they are designed as an extension of object-oriented programming languages to provide persistence, concurrency control, queries, and other database features for programming language objects.
  Starburst [Haas *et al.* 1990], [Lohman *et al.* 1991], [Widom 1996] and POSTGRES [Stonebraker *et al.* 1990], [Stonebraker & Kemnitz 1991] extend relational database system functionality with procedure calls, efficient object references, composite objects, and other object-oriented programming language capabilities.

Even though the second of the above two categories of DBMSs are also called "object-oriented and "extended relational" database systems, respectively, these terms have not been used consistently in the literature, and the data models of the systems are very similar in any case. The relevant details of the five systems are summarised in Table 2.3

2. 4. SOME LANDMARK GISs USING DATABASE TECHNOLOGY

DESIGN of GIS using database technology has so far been along one of the following three options [Günther & Lamberts 1994]:

1. Extending an existing GIS by adding database functionalities to it;
2. Coupling of a GIS with an existing DBMS; and
3. Augmenting an existing OODBMS with geometric and geographic functionalities.

In the following sub-sections, each of the above options will be illustrated, from the point of design of a specific GIS product.

System	Organisation	Announced In	Type	Query Language	Programming Language Basis	Remarks
O_2[1]	O_2 Technology, Altair Research Consortium (INRIA, Siemens-Nixdorf, Bull, CNRS, Univ. of Paris)	1991	Object-Oriented	Hybrid Algebra	O_2C	O_2 uses a newly designed query language, the mosr expressive in the field. An O_2 query can result in tuples, lists, sets, or any type of data value.
ObjectStore[2]	Object Design, Burlington, MA, USA	1991	Object-Oriented	C++ -like	C++	
GemStone[3]	Servio Corporation, Alameda, CA, USA	NA	Object-Oriented	Smalltalk-like	Smalltalk	
POSTGRES[4]	University of California,Berkeley, CA, USA	1990 (ver 1) 1991 (ver 2.1)	Extended Database	Extension to INGRES QUEL	None	Version 2.1 has about 180,000 lines of code in C, and is available free over the internet.
Starburst[5]	IBM Almaden Research Center, San Jose, CA, USA	1990	Extended Database	SQL Extensions	None	Approximately 100,000 lines of C source code. Extensible at every level—data storage methods, access methods, execution strategies, query optimization.

Table 2. 3. A Comparative Table Showing the Summary Details of Major OODBMSs

[1] [Deuce *et al.* 1990], [Deuce *et al.* 1991].
[2] [Lamb *et al.* 1991]
[3] [Butterworth *et al.* 1991]
[4] [Stonebraker *et al.* 1990], [Stonebraker &Kemnitz 1991]
[5] [Lohman *et al.* 1991], [Widom 1996]

2. 4. 1. Extending an Existing GIS by Adding Database Functionalities

SICAD system by Siemens AG, is a sort of CODASYL structure. It has its topological structure governed by a Master-Detail system. Internal address mechanism generate topological relations between the Master and Detail elements. A second set of addresses link all the elements of the same specification to allow flexible data access. SICAD can integrate vector and raster graphics. This hybrid processing is done either at the level of graphics display by superimposing two kinds of data, or at the level of interpretation by vector-to-raster, and vice-versa conversions.

2. 4. 2. Coupling of a GIS with an Existing RDBMS

This kind of dual architecture stores non-spatial data in commercial DBMS, (usually a RDBMS is chosen). ARC/INFO (*vide* page 18*)* uses the term *coverage* to denote the basic unit of data storage for a layered thematic database. A coverage represents a set of spatial units, called features where each feature has a location defined by coordinates, and topological pointers to other features, and possibly some non-spatial attributes. Some spatial data and all attributes are stored in feature attribute tables, oriented to either point, node, arc, or polygon spatial units.

A *region* is used to create geographic features from a set of polygons, while a *route* represents a feature composed of a set of arcs.
Grid is the main method for representing raster data. Grids can represent areas, points, lines, and continuous surfaces.
Continuous surfaces are represented using TIN, grid, or lattice.
The coverage stores line and polygon features topologically. This optimises data storage by reducing coordinate redundancy and facilitates spatial operations such as polygon overlay, and pathfinding through connectivity of arcs. Spatial searching in ARC/INFO is implemented through quadtrees [ESRI 1994].

Due to lack of an integrated file manager, as well as a homogeneous query language, this system can operate only in a single-user mode. ARC/INFO was originally designed for thematic mapping. Naturally enough, different attribute themes are conceived and organized as layers (coverages). Separate set of attribute data for each coverage corresponds to a different layer

ARC/INFO Version 8 has been implemented as a client to the spatial data server called Spatial Database Engine, designed by ESRI[1] This Spatial Database Engine integrates tabular and spatial data in an RDBMS. With the added functionality of a client-server mode of operation, the single-user constraint becomes less critical. The new version of ARC/INFO integrates with the tabular DBMS data, data from vector, raster, photographs, scanned documents, GPS, satellite images, CAD drawings, and videos in MPEG[2] format. ARC/INFO Open Development Environment enables the user to custom design applications using development tools such as Visual Basic, C++, PowerBuilder, Motif, ARC Macrolanguage, and the built-in scripting language of ARC/INFO ARC/INFO has been made "Year 2000-compliant"

In contrast to the layered topological structuring of ARC/INFO, SYSTEM 9[3] has a single layered topological structuring with multiple thematic content and integrated with a RDBMS. The topological structure is built from nodes, lines, and polygons. Like ARC/INFO, SYSTEM 9 assigns separate management and processing environment to the attribute and spatial data [Laurini & Thompson 1994].

[1] "ARC/INFO Version 8-First Look," http://www.esri.com/ updated on 26 Oct 1998
[2] Motion Pictures Experts Group Standard
[3] ® Prime Computer Inc., Natick, MA, USA

In SYSTEM 9, SMALLWORLD[1] and TIGRIS[2], the RDBMS is expanded to manage geographical objects by adding spatial data types, spatial functions, and predicates. This integration of RDBMS eliminates the disadvantages such as single-user, impedance mismatch between two manager levels and non-homogeneous query languages[David *et al* 1993]

The drawbacks of RDBMS still apply collectively to all the systems mentioned above. Data type system is weak, data have to be normalized in first normal form when hierarchical structures are needed, semantics are lost and have to be rebuilt through integrity constraints. Moreover, relational query languages such as SQL do not render portability to applications [Stonebraker 1989].

2. 4. 3. Augmenting an Existing OODBMS with Geometric and Geographic Functionalities.

In this option, all OO programming benefits such as object identity, complex objects, inheritance, and methods, become readily available. However, it is still necessary to enrich the OODBMS basic structure with geometrical data types(that is, point, line, area, *et cetra*) and add associated functions (union, distance, rectangle, *et cetra*) and predicates (cross, adjacency, and overlapping) to the query language.

GODOT has been a Project to build the prototype of a GIS on top of a commercial OODB by adding appropriate classes and methods [Günther & Lamberts 1994]. The system has a four-layer architecture, consisting of:

1. The commercial OODB, ObjectStore (*vide* Table 2.3, Page 69).

[1] ® Smallworld Systems Ltd., Cambridge, UK
[2] ® Intergraph Corporation, Huntsville, AL, USA

2. An extensible kernel with classes and methods for the representation and management of complex spatial and non-spatial data objects.
3. A collection of base components for query processing, graphics, database administration, and data exchange.
4. User interface modules for C/C++ programme interface, UNIX[1] command interface, and graphical user interface based on X Windows[2]

The GODOT data model is partitioned into three categories of objects: thematic objects, geometric objects, and graphic objects. Thematic objects represent real-world objects. They may be simple or complex (composed of several other thematic objects). Geometric objects are used to describe the geometric features of geoobjects. The elementary geometric objects form the class Geometry, with its three subclasses: Region, Arc, and Point. A geometric object can either be a singular point, arc, or region, or a collection of such singulars. Graphic objects are used for the display and interactive updating of thematic objects, represented by geoobjects.

The OODB technology, employed in the GODOT has helped in solving some of the limitations associated with the traditional GIS, *i.e.*, difficulty in handling large amount of data, problems associated with user requirements such as: structured object modelling, concurrency, and recovery Modelling of complex geographic objects was made much easier by the class hierarchies and structural object orientations available in OODB. Moreover, the behavioural object-orientations allowed the introduction of application-specific data types and operators. In effect, the flexibility is the major advantage of GODOT, as compared to ARC/INFO, developed around an RDBMS

[1] ® X / Open Company Ltd., USA.
[2] ® Common Law trademark of Massachusetts Institute of Technology, Cambridge, MA, USA.

2. 4. 4. Distributed GIS

It is becoming increasingly evident that all interesting computing systems are distributed. As soon as a system attracts multiple users, we find that the latter wish to collaborate—to share objects with each other—using a variety of computers in a multiplicity of locations.

It is recognised that GIS technology provides an ideal tool for industrial site selection, with the target domain ranging from a small town to a large state, or even the whole country South Carolina Infrastructure and Economic Planning (SCIP) project was launched in 1987, jointly by the University of South Carolina, Columbia, SC, USA, and the South Carolina Department of Commerce [Cowen & Shirley 1991]. The entire system is made up of a distributed network of UNIX workstations running ARC/INFO and operates on the most extensive geographic database ever created for the state of South Carolina, U S. These data include existing industrial facilities, sites, and buildings, details of infrastructural facilities, demographic and land cover data.

An expert system that is a major constituent of the SCIP decision support system, has a standard communication interface between a C language production system (CLIPS) and the ARC/INFO database [Cowen & Ehler 1994]. Information in the form of ASCII text files are exchanged between the expert system and the main GIS. Most of the final system was programmed in the ARC macro language, thereby facilitating further customisation of the system. The rules for the knowledge-base were coded in CLIPS which has its own script and also available in the public domain.

2. 5. KNOWLEDGE MANAGEMENT

IT WAS observed in Section 2.3 that most real-world data management problems require data, object, and knowledge management services, and that knowledge management requires the ability to store and enforce a collection of rules. Knowledge-based systems typically use rules to embody knowledge of a human expert. It is common to use the concept of a *knowledge base* and a *rule base* interchangeably One application area of KBS, considered in the work reported by this thesis, was designing a GIS, covering the attributes of a city In such an application, the map of the concerned city is treated as a collection of features which are represented in terms of points, lines, or areas.

The possible alternatives available for combining the rules and data within the application are:

1 Put the rules in an application programme and the data in a database.
2. Put the rules in an expert system shell and the data in a database.
3 Put both the rules and the data in a composite data-rule base.
4. Put the rules and the data in an expert system shell.

Option 1 is the favoured implementation adopted for general data processing. The disadvantage of this approach is that the rules are buried in the application programme and hence, they become hard to code, hard to change, and hard to enforce in a consistent fashion.

The second alternative is advocated by the vendors of expert system shells and is termed *loose coupling*. In this scheme, the shell environment which contains the rules and the inference engine, reside in the main memory. As and when required, this engine runs a query against a database to gather extra information needed. In effect, rules are stored in a rule manager and data in a separate data manager. These two subsystems are coupled together by a limited control mechanism. KEE[1] is the most widely used expert system building tool for sophisticated expert systems. However, loose coupling will fail on a wide variety of problems. Suppose one wants to monitor a certain data item in the database, and whenever it changes, a second data item is to be set equal to the value of the first data item. By the time the first data item is fetched from the database it will become outdated, and hence must be fetched anew. Such repeated querying of the database will consume resources needlessly and rarely will the values be consistent.

In contrast to the loose coupling, the third alternative is to have *tight coupling*, wherein a single rule-data system manages both rules and data [Sellis *et al.* 1993]. Such a system must perform asynchronous operations to enforce the rules. The current breed of commercial DBMSs are passive in that they respond to user's requests; but are incapable of independent action. An *active* system can tag the data item being monitored and send a message to the application program whenever the data item changes. Such a data manager will automatically support sharing of rules, the ability to add and drop rules, and the ability to query the rule set. POSTGRES is designed along these lines, with one general-purpose rules system inside the data manager [Stonebraker 1995].

The last option is to put both data and the rules in an expert system shell, so that facts that are to be examined by the rule engine are resident in the main memory

[1] ® IntelliCorp, 1975 El Camino Real West, Mountain View, CA, USA

along with the rules. Such a system will be infeasible when the number of facts become too large to be in the main memory Also it will not work in a multi-user mode [Stonebraker 1989]. These topics will be discussed further in Chapter 3

2. 6. PROGRESS IN OTHER RELATED AREAS

WHEN maps incorporating sound are used to present spatial information they tend to be less interactive; but they are effective when used to analyse or explore. In a multimedia atlas, for example, pointing to a specific country starts the national anthem of that country Sound might also be applied as background music to embellish different map phenomena such as industry, small video clips relating to the area under browsing, or starting short sentences spoken in the dialect of the region as the screen pointer is moved across the screen.

2. 6. 1. Multimedia, Hyperdocuments, and Hypermaps

Integrated use of text, graphic, image, and sound data has become a common place activity, in the domain of corporate advertising, educational materials, *et cetra*. When we have Multimedia documents for which a geographic access is important, we have more than a combined use of multimedia data. In such a situation, we should be able to retrieve all information, pertaining to an area or a point identified by a random mouse-click on a map image. The target information may be contained in documents, satellite images, seismic measurements, architectural drawings, land-use zoning maps, *et certa*, for which access is based on coordinates. When the principle of hypertext, applicable to textual material, is extended to documents in other formats, we have hyperdocuments and hypermedia.

Animation is an effective mode to present the structure or history of a city, by showing the thematic layers, one after another, starting from the basic geographic relief, followed by hydrography, soil structure, infrastructure, changes in land use over a period, *et cetra*. Animations are most appropriate to introduce temporal components of spatial data, indirectly In spite of the perceived importance of the individual multimedia elements (animation, sound, still images, text, and video), they are being realised today only as stand-alone projects. Incorporating them effectively and seamlessly into a full GIS environment is still under study [Kraak 1996].

Of the three common database management systems *viz.*, centralized, master-slave, and federated, that can be identified for MM applications, the most suitable one for large-scale MM applications are the federated and master-slave types. [Little & Ghafoor 1991] have laid out a methodology for spatial and temporal composition of MM objects

Hypermedia applications can be constructed either by adding hypermedia functionality to existing (or third-party) applications or by developing new hypermedia applications from scratch Hyperform [Wiil 1995] is a dynamic, open and distributed multiuser hypermedia application development environment, providing a hypertext database for storing nodes, links, and anchors [Isakowitz et al 1995].

There are two fundamental problems limiting the success of software libraries for processing digital audio and video.

1. Developing software components (like media players. Netscape plug-ins. ActiveX controls) which decode and process digital audio and video is time-consuming and costly Reuse of such code is not practical because high performance

from these components requires tight coupling of the code to the processing environment

2. The software decoder is required to handle a wide range of processors with speeds differing by orders of magnitude.

Since the processing units are normally shared by a number of applications, the availability of these units to the software decoder will not be guaranteed on demand as the system load fluctuates. This variation will result in undesirable skips or distortion in the output signal, thereby degrading the presentation quality Object-oriented application frameworks is an emerging technology for reusable design of all or part of a system [Johnson 1997]. Systems like OLE, and OpenDoc are frameworks; Java is spawning new frameworks like Beans. [Posnak *et al.* 1997] have described a framework for developing software components that decode and process digital audio and video data. Given that compression technology is still evolving and presentation processing is a major bottleneck in communication performance, improvements in this area should have a positive impact on the structure of future hypermedia applications.

There is currently a dearth of tools for multimedia data modelling. This situation has to change if multimedia information is to be intelligently and efficiently managed. The continued development of the MPEG-7 standard on the content-based description of multimedia data should spur developments of such tools [Grosky 1997].

On the security front, a multi-level security model classifies multimedia documents into such levels as "Secret" or "Top Secret" and formulates various rules concerning the security levels of related documents. However, much work has to be done to formalize the presentation of multimedia objects with selected objects masked for security purpose[Grosky 1997].

To summarise, the main ideas of the 'hyper' concept are:

> A *hypertext* is a logical network of textual content portions that can be accessed electronically.
> A *hyperdocument* is a logical network of audio-visual content portions that can be accessed electronically, and displayed over audio-visual equipment, on-line.
> All the features of the hyperdocument are present in a *hypermedia*, but the latter has additional facilities for digitized speech, digital audio recording, movies, film clips, *et cetra*.

The term *hypermap* refers to multimedia hyperdocuments with a geographic coordinate-based access *via* mouse or its equivalent. Navigating in a hypermap has two aspects: thematic navigation, as is usually done in hypertext and in hypermedia, and spatial navigation which is particular to hypermaps. In thematic navigation, several modes of navigation can be implemented, to serve users with levels of competence varying from 'novice' to 'expert'.

For solving the spatial navigation, the region to which a point P_1 belongs, can be determined by by comparing its x-coordinate to the x-coordinates of various labelled rectangles, and then repeating the process with the corresponding y-coordinates. As a result, we obtain two lists of possible rectangles. The answer is given by the intersection of of both lists. The region query can be answered in a similar manner [Laurini & Thompson 1994].

2. 6. 2. Web-Related Developments

In this section, a few World Wide Web-related developments in the area of GIS are discussed.

2. 6. 2. A. GIS Queries on the Web

The development of Web technology (browsers, servers, and the like) is having a major impact far beyond the world of Web sites. Legacy systems would be more effective if they

> could be delivered to people's desk over corporate intranets without concern for hardware platform,
> had access to databases distributed across the corporation, and
> could be transformed into a modern graphical user interface.

As distributed applications proliferate on the Internet and World Wide Web, GISs ranging from corporate information systems to spatially enabled applications have become mainstream. The Web's heterogeneous environment requires that query mechanisms be both software- and hardware-neutral. This creates a need for spatial query mechanisms that are distributed and open [Coetzee & Bishop 1998]. Coetzee & Bishop have compared spatial queries for GISs using different approaches: executing native C/C++ APIs locally using the Java Native Interface (JNI); and using Remote Method Invocation (RMI) remotely for distributes-object execution from a Java middleware server. The authors have found that one key advantage of using RMI over JNI was that RMI could run in a Java applet, thus allowing for spatial querying in the client Web browser. According to them, executing GIS queries locally gives superior performance over executing distributed queries *via* the Web.

2. 6. 2. B. OO GIS on the Web

Even though OO databases have been around for many years, industry does not use them heavily. Java has enhanced OO's benefits by providing a vehicle for running such applications across several platforms without the need for extensive cross-platform coding. [Cobb *et al.* 1998] has developed a Web-based system that lets users view digital mapping data. They used an OO database (GemStone) to store mapping information and Common Object Request Broker Architecture (CORBA) to distribute the database objects to Web clients. The stored data included Digital Nautical Chart, the World Vector Shoreline Plus, the Vector Smart Map, and the Urban Vector Smart Map, all produced and distributed by the National Imagery and Mapping Agency (NIMA) of the US Department of Defense. Applets request information from a Smalltalk server, which passes complex objects back and forth.

2. 6. 3. New Standards for GIS Software

The Open GIS Consortium (OGC) has started working on a standard for GIS software. The OGC expects the standard geoprocessing interface specification to yield several benefits. The standards could enable the development of products like Web-enabled multimedia Yellow Pages, so that a user could submit a multiple query to the on-line Yellow Pages and get a map showing where the stores selling a required item are located. The lack of a standard has been preventing, the merging of geographical data from heterogeneous databases into a complex map [IEEE Computer Society 1996].

2. 7. SUMMARY

THIS chapter has reviewed the development of concepts, methodologies, and tools, originating in disciplines as varied as geography, computer science, and information technology The first generation of computer applications involved numerical computing. The second generation, responding to the needs of commercial data processing, led to the development of new data structures, algorithms for searching and sorting, *et cetra*. We are now in the third generation of computer applications, dominated by computing with geometries, pictorial objects, and map data.

The development of data structures due to these three phases have brought out the following distinctive aspects of spatial data over the rest:

> Proximity access based on location in space: objects are typically not accessed *via* a unique identifier, but by the fact that they lie in or near some region in space.
> Object representation is an integral part of the data structure. Spatial data applications involve not only points, but many kinds of other primitives (line segments, triangles, polygons, *et cetra*) and composites of these primitives.

In the 1970s and 1980s, the database community lumped every kind of data (other than fixed-format records), like pictures, voice, and spatial data, into a heterogeneous collection called 'non-standard data', and extended relational database technology with its simple conceptual structure, to handle all kinds of data. It soon became evident that RDBMS was not suited for this, because the latter was oriented towards access algorithms that were designed to operate on data organised in tabular forms. Forcing spatial data into tabular form, through relations like *polyhedra*, *faces*, *edges*, and *vertices*, have harmful consequences. Geometric proximity is not reflected by proximity in memory. All vertices, no matter how far apart in space, get stored contiguously in the same relation, whereas a vertex and its incidental edges and faces are scattered all over the storage. Instead of accessing one entire object as a unit, bits and pieces of this object have to be gathered in several disk accesses.

The use of OODB technology has been an attempt to solve some of the limitations of RDBMS. Modelling geographic objects became relatively easier through the class hierarchies, and the structural object orientations provided by OODBs allowed introduction of application-specific data types and operators.

With so many complexities associated with the geographical data, we require reasoning and deduction facilities in GIS, more than in any other computer application. But a robust methodology employing the paradigm of knowledge-based system is yet to be realised. Some of these points have been discussed.

The growth of technologies of multimedia, hyperdocuments, and their refinements have been so fast that they would be ready as an industry, much before GIS would become one.

Chapter 3

Knowledge-Based Systems

AMONG the software systems that have been created in the area of artificial intelligence (AI), some are heavily oriented toward emphasizing sophisticated approaches to their control structures, while a majority of them emphasizes the rôle of the knowledge related to a specific subject. Successful systems are characteristically "knowledge-rich even if they are methods-poor [Feigenbaum 1977]. The latter category of systems are called *knowledge-based systems* (KBS) because they are heavily oriented toward specific and narrow knowledge domains. Common knowledge representations used are:

> Predicate logic — For relatively small systems where the data and rules are known with certainty;
>
> Semantic networks, frames, and object-oriented approaches — For systems where inheritance of attributes among classes is important;
>
> Rule-based approaches — For systems where the control structure (known as "inference engine") determines the direction of the deduction

Rules or implications, stated in the form of *if-then* rules, are one of the most common ways of expressing the domain knowledge, so much so that many people associate knowledge-based systems with rule-based systems. The rules in a KBS are mostly heuristic rules. Heuristic rules are statements that we accept as true based on our real-world experience, and that can be generally applied to solve problems pertaining to a particular domain. Heuristic rules are often referred to as *production* *rules,* and a system that operates using a set of production rules is known as a *production system*.

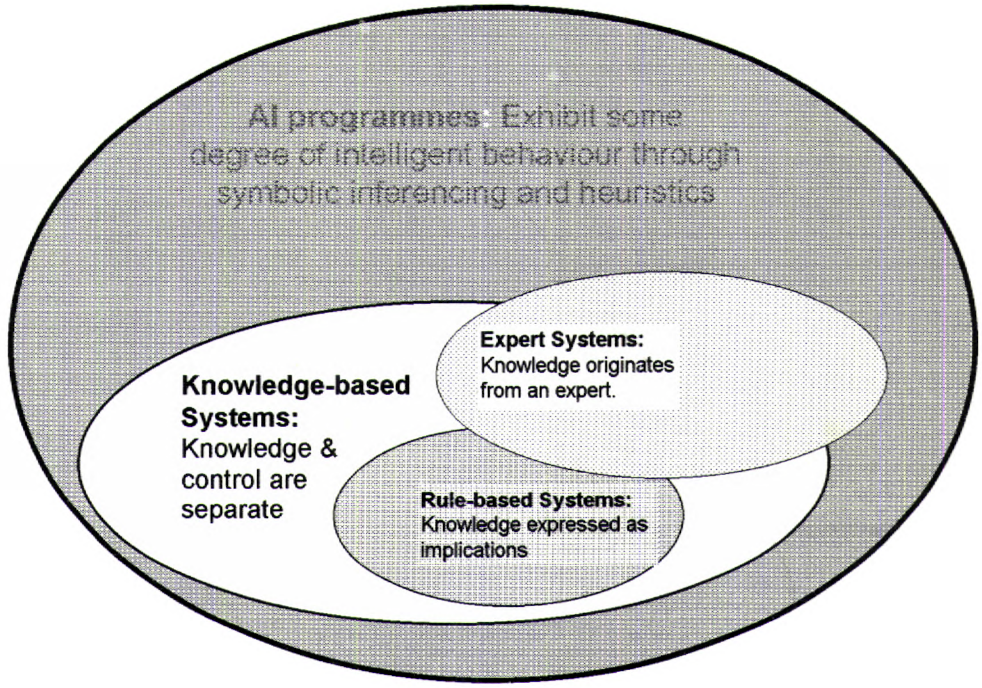


Figure 3. 1. An overall Classification of AI Programmes

The overall classification of AI programmes is shown in Figure 3.1 Any programme that exhibits performance that may be called "intelligent", come under the category of AI programmes. When procedures for reaching a solution are not well-defined, we introduce "heuristics" which, while not guaranteeing solutions at all times, do find them often enough—and quickly enough—to warrant their use and, as stated earlier, KBSs form a subclass of these, and rule-based systems are a specialisation of KBSs. As shown in Figure 3.1, expert systems may exist in all the three categories.

3. 1. SYBOLIC LOGIC AND AI

IN DEVELOPING knowledge representations, one of the most useful logical connectives is the implication (IF) operator. The implication operator is used to formulate *conditional statements*, that is, statements of the form

if antecedent
 then consequent

Symbolic information encoded *via* (possibly compound) 'antecedent' and 'consequent' statements, together with the implication connective, form the basis for many AI implementations, including rule-based systems.

The implication or conditional operator is written as ← or → depending upon the direction of the implication. When written as $p \rightarrow q$, the term on the left, that is, the antecedent *p implies* the term on the right, that is, the consequent *q*. If the implication is TRUE when the term on the left is TRUE, the term on the right must also be TRUE.

3. 1. 1. Logical Properties of Implication

The logical properties of implication are summarised in the accompanying truth table (Table 3.1).

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Table 3. 1. Implication Truth Table

Implication yields a fundamental basis for rule-based inference systems. If we assume that implication is TRUE, this knowledge may be used to constrain the allowable truth values of the antecedent and consequent. Consider the antecedent p being TRUE and the value of the implication ($p \rightarrow q$) also being TRUE. (This is the first row of Table 3.1) This constrains or forces q to assume the value TRUE.

It is noteworthy that implication is used in a weak sense, in logic. For example,

1. p and q together do not need to make sense.
   if 4+ 2 =3
   then America is within India.
   This corresponds to row 4 of Table 3.1. A FALSE hypothesis (antecedent) implies any conclusion.

2. A simple statement (that is, containing no connectives) could be viewed as an implication with no antecedents.

3. The implication statement, $p \rightarrow q$ does not signify that 'p causes q'; there is no cause-effect relationship. From the truth table, the fact that the implication is TRUE signifies that
   a.) p implies q.
   b.) q is TRUE if p is TRUE.
   c.) p is TRUE only if q is TRUE.
   d.) p is a sufficient condition for q.
   e.) q is a necessary condition for p.

4. A Karnaugh map for the implication, $p \rightarrow q$, yields the result, $(p \rightarrow q) \equiv (\neg p) \cup q$. The significance of this result is that, given an implication statement is known to have the value TRUE, the fact that the antecedent is also TRUE forces the consequent to be TRUE. This is the logical basis for *forward chaining* in a rule-based system. The antecedent statement may, however, be itself a compound statement, involving the conjunction of several simple statements.

3. 1. 2. Inferencing

Inferencing is the process of deducing (new) facts from (other) existing facts. The process of reaching a conclusion from a set of propositions is called deductive reasoning. One deduction strategy is known as *modus ponens,* expressed as a compound statement involving implication: $(p \cap (p \rightarrow q)) \rightarrow q$ Modus ponens provides a rigourous and justifiable mechanism for making inferences.

Another mathematically tractable process is known as **abduction**. Abduction is a mathematically justifiable, practical, and reasonable way of generating hypotheses. It is therefore closely related to the analysis of backward chaining and implication.

$p \rightarrow q$	a rule; assumed TRUE
q	an observation; assumed TRUE
p	an explanation; possibly but not necessarily TRUE

Example:

Rule:	If traffic density is high, the road is concrete.	(3.1)
Observation:	Crescent Lane is concrete.	(3.2)
Possible explanation (Abduction):	The traffic density on Crescent Lane must be high	(3.3)

Here, the inference of high traffic density is only a *possible explanation*. 3.1 only explains 3.2; it is not necessarily true

A more rigorous logical formulation of explanation is provided by *modus tollens*, defined as $((p \rightarrow q) \cap \neg q) \rightarrow \neg p$ (Fourth row of Table 3 1). Modus tollens provides a logically sound explanation which is necessarily TRUE.

When two facts are always concomitant, by induction we can affirm the rule that expresses the relationship between them. Knowing that when p is TRUE, r is also TRUE, we can derive (induce) a rule, expressed as $(p; r) \rightarrow (p \rightarrow r)$.

Reasoning by **transitivity** is as follows: $(p \rightarrow q\ ,\ q\ \rightarrow\ r) \rightarrow\ (p\ \rightarrow r)$.

While rule-based systems are based on the four types of reasoning, *viz*., modus ponens, modus tollens, induction, and transitivity, the more commonly used are sets of deductions, also called **forward chaining**, or sets of abductions, also called **backward chaining**, by transitivity Forward chaining is the way to test the consequences of a set of starting assertions (facts).

Consider a series of assertions or **if-then statements**. Variables p_i and q_i are used to represent the truth values of the antecedents and consequents, respectively, in the following implications:

Implication 1

p_1 A township is planned near lake L.

q_1 There are no houses near lake L, at present.

Implication 2

p_2 There are no houses near lake L, at present.

q_2 Lake L is clean.

Fact

p_3 Townships near lakes pollute the lakes.

Conjecture

q_3 Lake L will be polluted when houses are built near it. (3-4)

The consequents of some implications in (3-4) are the same as the antecedents of others. The set of following assertions, each of which we assume to be TRUE, form a rule base, or knowledge base.

$$\begin{array}{l} p_1 \rightarrow q_1 \\ p_2 \rightarrow q_2 \\ p_2 \quad\ q_1 \\ p_3 \end{array} \qquad (3\text{-}5)$$

This knowledge base may be rewritten using variable substitution and simplifies to

$$\begin{array}{l} p_1 \rightarrow q_1 \\ q_1 \rightarrow q_2 \\ p_3 \end{array} \qquad (3\text{-}6)$$

While it is assumed that the assertions in (3-6) are all true, the same cannot be said about the components of those statements. To determine the truth of statement q_3, given that p_1 is TRUE, forward chaining can be used to "chain" production of new TRUE facts from a starting one, and arrive at a truth value for q_3. The logical sequence is as follows:

$(p_1 \rightarrow q_1)$ and (p_1 being TRUE) $\rightarrow$ (q_1 is TRUE)
(q_1 is TRUE) and (($q_1 = p_2$) being TRUE) yields (p_2 is TRUE)
$(p_2 \rightarrow q_2)$ and (p_2 being TRUE) $\rightarrow$ (q_2 is TRUE)
(q_2 is TRUE) and (p_3 being TRUE) yields (q_3 is TRUE) (3-7)

The sequence of productions employed in deducing "the lake L will be polluted when houses are built near it" from "A township is planned near the lake" is shown in Figure 3.2. This type of diagram is referred to as an **inference net.** Inference

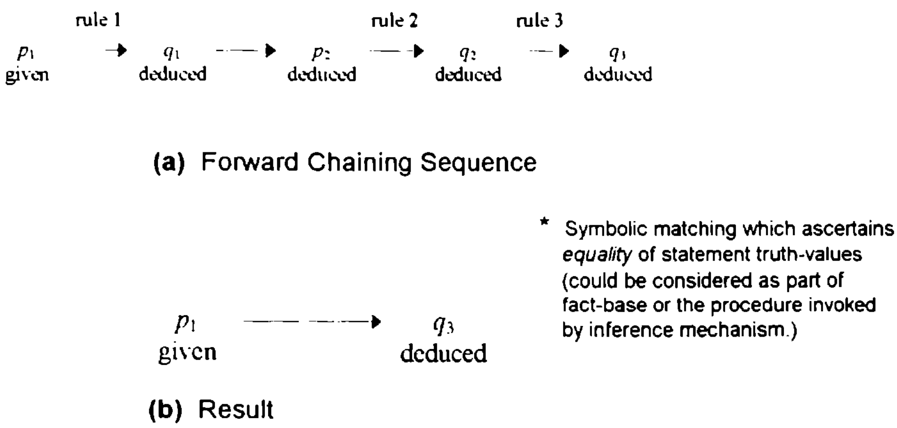


Figure 3. 2. Inference Net for Three-Rule Deduction Example. (Given p_1 is TRUE)

nets are seldom as simple as this; there could be hundreds of rules (instead of five in the present case) in a typical rule-base. The action indicated by following an arrow from the antecedent of a rule to its consequent (and thus producing a statement whose truth value is known to be TRUE) is shown in the Figure. This is referred to as *firing the rule*. The process of deciding which rule to fire is fundamental to the inference control strategy

3. 1. 3. Inference Rules

The semantics of predicate calculus provide a basis for formal theory of *logical inference*. The ability to infer new correct expressions from a set of true assertions is an important feature of the predicate calculus. These new expressions are correct in that they are *consistent* with all previous interpretations of the original set of expressions.

Modus ponens is a sound inference rule. Given an expression of the form p --> q and another expression of the form p such that both are TRUE under an interpretation I, then modus ponens allows us to infer that q is also TRUE for that interpretation; and q will be TRUE for *all* interpretations for which p and p --> q are TRUE.

3. 1. 3. A. Resolution

The resolution approach yields new clauses from an initial set. Resolution is a powerful and indirect method of inference. The utility of resolution in inference application is subtle. Often, resolution is used to show that a set of clauses is *inconsistent*, in the sense that the resolution process produces a logical contradiction. A clause is proven to be true, in the context of a set of clauses known to be true, by appending the logical NOT of this clause to the set and seeking a contradiction. If the resolution process fails to find a contradiction, the negative of what we seek to prove is *logically consistent* with the database, and thus the clause cannot be TRUE.

Resolution proceeds by augmenting the database with the negation of the desired hypothesis. Then clauses are resolved pairwise in the augmented database until a contradiction is found. If none is generated, we conclude that the situation is consistent, and therefore the hypothesis is FALSE. First, all implications and equalities are to be removed from the well-formed formulae.

$(p \rightarrow q)$ is replaced with $((\neg p) \cup q)$
$(p \equiv q)$ is replaced with $((\neg p) \cup q) \cap (p \cup (\neg q))$

Pairwise resolution of clauses uses the strategy of verifying that

$$(a \cup b) \cap (\neg a \cup c) \rightarrow (b \cup c)$$

Example: A database has its initial condition (D1) is consistent and its contents are given below. All statements are assumed to be TRUE.

1 p_1
2. $p_1 \rightarrow q_1$
3 $q_1 \rightarrow q_2$

The goal is to prove, using D1 that q_2 is TRUE. Therefore, $\neg q_2$ is added to D1, clauses are simplified to a suitable form, and pairwise resolution proceeds. The modified database D2 is:

1. p_1
2. $\neg p_1 \cup q_1$
3. $\neg q_1 \cup q_2$
4. $\neg q_2$

Resolution of (2) with (3) yields $(\neg p_1 \cup q_2)$ which then becomes the 5th element of the database (D3). Resolution of (4) with (5) yields $\neg p_1$, and the new database (D4) has the following configuration

1. p_1
2. $\neg p_1 \cup q_1$
3. $\neg q_1 \cup q_2$
4. $\neg q_2$
5. $\neg p_1 \cup q_2$
6. $\neg p_1$

Resolution of (1) with (6) produces a contradiction. Therefore, $\neg q_2$ is inconsistent with D1. Hence, *by refutation*, q_2 is TRUE.

More details on Resolution can be found in Section 11.2 of [Luger & Stubblefield 1989].

3. 1. 3. B. Unification

In order to apply inference rules such as modus ponens, an inference system must be able to determine when two expressions are the same or match. In propositional calculus, this is trivial. two expressions match if and only if they are syntactically identical. In predicate calculus, the process of matching two sentences is complicated by the existence of variables in the expressions. Unification is a systematic procedure for instantiation of variables in wffs. Since the truth value of predicates is a function of the values assumed by their arguments, the controlled instantiation of values provides a means of validating the truth values of compound statements containing predicates. The basis of unification is *substitution*.

> **Unification** is the process of making two expressions *identical* (that is, unify them) by finding *appropriate substitutions* or *bindings of variables* in these expressions.
>
> A **substitution** is a set of *assignments of terms* to variables, with no variable being assigned more than one term.
>
> A set of expressions is **unifiable** if and only if there exists one or more (unifying) substitutions that make the expressions identical.

In a simplified manner, substitution can be illustrated as follows: If a has *a* property *P* and all objects that have property *P* also have property Q, we conclude that *a* has property *Q* [Schalkoff 1990].

$$\frac{\begin{array}{c} P\,(a) \\ \forall\, x\, P\,(x) \rightarrow Q\,(x) \end{array}}{Q\,(a)}$$

In concluding $Q\,(a)$, x was substituted with a. This was made possible by the implication $P\,(x) \rightarrow Q\,(x)$ that was TRUE for all x, and in particular for $x = a$.

Substitutions permit simplifications or reduction of expressions through the cancellation of complementary literals.

One of the most important and logically sound inferencing techniques involves using modus ponens with statements (especially rules) containing variables. A single substitution may be represented by a pair of the form (t/R), denoting a substitution of the variable R with the constant value t, in a statement. Application of a set of substitutions $\theta = (t_i \; R_i)$, $i = 1, 2, \ldots, n$, to a statement P as $P\theta$, where every occurrence of variable R_i in P is replaced by t_i. Two statements, P and P_1 are unifiable, that is, they have a unifying substitution, θ, if $P\theta = P_1\theta$. In other words, a set of substitutions, θ, exists that makes P and P_1 identical.

If statement P_1 contains no variables, then $P_1 = P\theta$ represents a unifying substitution. For example, the two statements

P_1 = (*block_1 left_of block_2*)
P = (A *left_of* B)

wherein A and B are variables, unify with

θ = { (*block_1* / A), (*block_2* / B) }

Assume the following axioms in the initial database:

(A left_of B) → (B right_of A) (3-8)
(block_1 left_of block_2) (3-9)

The following inference procedure is logically valid:

1. Unify antecedent of implication statement (3-8) with fact (3-9).

These unify with θ= {(block_1 / A), (block_2 / B)}.

Therefore, (3-8) and *modus ponens* allow deduction of the new fact

(B left_of A) θ

and, with substitution

(block_2 right-of block_1) (3-10)

3. 2. PRODUCTION SYSTEMS

PRODUCTION systems represent symbolically tractable, solutions to many AI applications. A production system consists of

1. An unordered set of production rules (Production Memory) that modify the existing database of facts and / or production memory and whose applicability is conditioned on the current database of facts (or production memory).
2. A database of facts (Data Memory), maintained separately from the production memory.
3. A control mechanism or a rule interpreter , also known as the Inference Engine, that determines the applicability of the rules , the selection of appropriate rules, and the resolution of conflicts that may arise when more than one production become applicable at the same time.

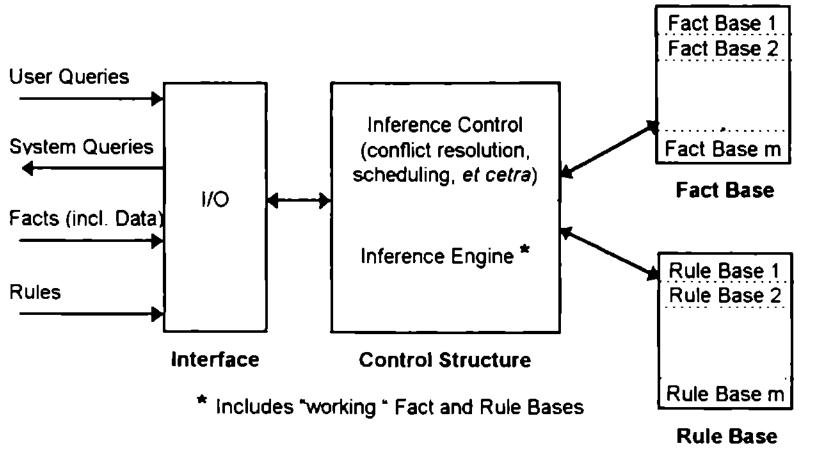


Figure 3. 3. The Structure of a Rule-Based Production System

Production systems are a subset of **pattern directed systems**—systems whose production applications are driven by input (or initial) data patterns. Specifying the production conditions in the form of **if** statements and actions *via* **then,** leads to a rule-based system paradigm. The term **expert system,** is used to indicate a subset of production systems that are restricted to specific task domains. Figure 3. 3 shows the structure of a rule-based production system.

3. 2. 1. The Inference Engine (IE)

A rule in the Rule Base is said to be *triggered* when all of the antecedents of the implication are satisfied — that is, when these antecedents are present (or asserted) in memory The inference engine can be treated as a finite state machine (FSM) with a cycle consisting of the action states:

1. Match rules.
2. Select rules.
3. Execute rules.
4. Check stopping condition (goal satisfaction).

The IE somewhat exercises control over the production system operation in the sense that it either regulates the production of new databases (forward chaining) or the verification of hypothetical information (backward chaining) Fundamental to the operation of the IE is the process of matching. Since more knowledge may be encoded with rules using variables, this involves matching with variables and thus, a suitable unification algorithm.

3. 2. 2. Rule Selection

Another critical aspect of IE operation is in the selection of rules. One type of specialised production system is the commutative system. A rule is defined to be "applicable in the context of a database, denoted by $\mathcal{D}$", if the conjunction of the antecedents for that rule are satisfied by $\mathcal{D}$, that is, the rule is eligible for firing. A *commutative* production system has three important properties [Nilsson 1980]:

1. Any rule appllicable to $\mathcal{D}$ is also applicable to any database derived from $\mathcal{D}$ by successive applications of applicable rules.
2. If a goal is satisfied by $\mathcal{D}$, then it is also satisfied by any database produced by applying applicale rules to $\mathcal{D}$.

3. The database generated by firing of a sequence of rules applicable to $\mathcal{D}$ is invariant to permutations in the firing sequence. (This does not imply that if we find a sequence of rules that change the initial state of the system to the goal state, then we can arbitrarily reorder this sequence and still find that the reordered sequence will achieve the goal.)

A production system having the commutative property enables the IE to avoid many of the solution paths that differ only in the order in which initially applicable rules are applied. For large rulebases, this may be a significant number This allows the IE to select and apply applicable rules without provision for reconsideration at a later point in the inference sequence. This *irrevocable control strategy* is in contrast with the *tentative control strategy* which selects and applies rules, but has a provision for reconsideration at a later point in the inference process. A control strategy that allows backtracking, PROLOG's inference mechanism, for example, adopts the tentative strategy A practical or generally useful control strategy is not likely to be a simple matter of choice.

The best example of a non-commutative production system is a planning system. In planning systems, the order in which productions are invoked is usually critical to the outcome of the system. Achievements of a plan that meets the prescribed goal is highly dependent upon the chosen sequence of operations that make up the plan.

3. 2. 3. Conflict Resolution

Conflict resolution is a selection process for determining "good" rules, from among all applicable rules. There are several approaches to conflict resolution. Some of them are the following:

1. In choosing "rule a" versus "rule b", examine the antecedents of both rules. If the antecedents of "rule a" are a superset of those of "rule b" then the former is more specialised than the latter, because more constraints are applicable to the former, "rule a" As per the strategy, known as **specificity-ordered conflict resolution**, the rule with the stricter precondition is chosen.
2. Examine the conflict set, and choose a rule whose firing takes the system closer to the goal, as compared to other rules. This implies that the goal has already been specified.
3. Choose the rule that has the largest number of consequents, assuming that, more the additional information available, closer the system would be to the goal.
4. Rank rules according to an *a priori* firing desirability

If rules contain variables, it is possible for a rule to be used repeatedly This makes room for possible additional conflict resolution strategies, as listed below

5. Choose the most recently used rule.
6. Choose the least recently used rule.
7. Choose the rule with the least (or most) number of variables.

3. 2. 4. Inference Strategies

When a rule is said to have fired, as a result of the conflict resolution process, the rule is removed from consideration, and the cycle of considering all currently triggered rules, including any triggered by the result of the rule just fired, continues. This approach, called *refraction*, is to avoid looping on the same rule. Rules removed after being fired are re-incorporated into the production memory when conditions relating to their antecedents have changed. This iterative process is referred to as **recognize-act-cycle**, and is represented graphically in Figure 3.4.

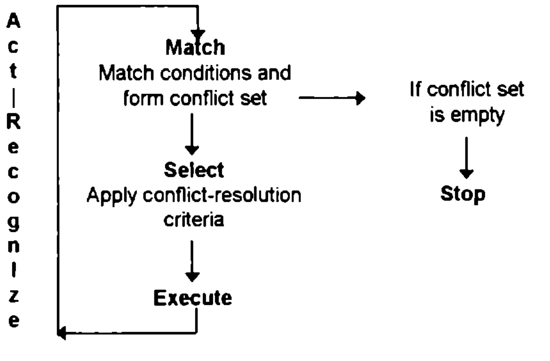


Figure 3. 4. The Recognize-Act Cycle

The computational difference between the two inferencing strategies, namely, the *forward chaining* approach and the *backward chaining* approach, merits brief discussion. In the forward chaining approach, the inference process proceeds exhaustively from the starting set of facts to a set of new facts. Thus, all new facts are generated, unless a control or stopping mechanism is embedded inside the process. A consequence of this is the checking and firing of a substantial number of rules; some of these rule firings may contribute very little to advance the system towards a goal.

The backward chaining paradigm, on the other hand, seeks only to prove the validity of a chosen fact or expression whose truth value is not known *a priori*. Computationally, it is perhaps more efficient than forward chaining, since it represents a goal-directed strategy that may eliminate checking of many superfluous paths. If more than one hypothesis is involved, backward chaining attempts to verify each one independently

Forward chaining is appropriate when

There exist many equally acceptable goal states, a narrow body of relevant information, by way of facts and rules, and a single initial state.
All or most of the required facts are in the initial database.
It is difficult, initially, to form a goal or hypothesis to be verified.

Backward chaining is appropriate when

There exists a single goal state (like in diagnosis, where we desire to confirm the occurrence of a single event or malady) and a large amount of potentially relevant initial information [Brownston *et al.* 1985].
Relevant data must be acquired as a part of the inference process, for example, "asking questions"
Large numbers of applicable rules exist. This potentially leads to the production of many extraneous facts.

3. 2. 5. The OPS5 Production System

An important family of AI languages comes directly out of the production system language research at Carnegie Mellon University (CMU), USA [Brownston *et al.* 1985] These are the OPS languages; OPS stands for Official Production System. These languages have proved highly effective for programming and designing production systems, expert systems, and other AI applications. OPS5 is the most widely used interpreter, released by CMU in 1981.

From a programming viewpoint, an OPS5 implementation consists of a set of suitably coded facts in *working memory* (WM) and a set of rules in *production memory*. These are shown in Figure 3 4. A production is designated by a *list* whose **car** is the symbol p. The *n*th element of the list is the rule name find-segment (as shown below), and the antecedents are a series of lists ending with the symbol --> In the simplest form, the LHS of a rule is a set of *condition elements* that specify patterns to be matched against facts in working memory. The RHS of the rule (indicated after the arrow, as shown below) is a sequence of actions. A simple rule, for example, for finding in WM a line segment whose endpoints are the values of the variables (<x0>, <y0>, <x1>,<y1>) in OPS5 is

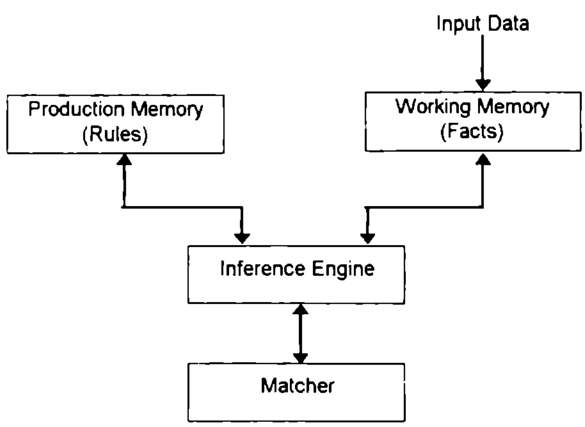


Figure 3. 4. OPS5 Production System Architecture

```
(p find-segment
        (segment ^x0 <x0> ^y0 <y0> ^x1 <x1> ^y1 <y1>)
        (segment ^x1 <x1> ^y1 <y1> ^x2 <x2> ^y2 <y2>)
-- >
        (write A longer segment has been found with coordinates
                (<x0>, <y0>, <x2>, <y2>)))
```

Further details on OPS5 programme structure is given in Appendix B.

It was stated in Section 3.2.1 (Page 97) that the inference engine of a production system can be treated as a FSM with a cycle consisting of three action states, namely, Match, Select, and Execute. In the first state Match, the machine finds those rules which are triggered. The triggered rules are all potential candidates for execution; and they are collectively referred to as the *conflict set*. The conflict set is passed along to the second state Select, which applies the selection strategy (determined by the specific production system strategy) to determine which rules are suitable for execution.

3. 2. 5. A. The Rete Match Algorithm

In the OPS5 interpreter, the recognize-act cycle (*vide* Page 99) has been changed to

> 1. **Conflict resolution.** Input is the conflict set. Select one production with a satisfied LHS. If there are no productions with satisfying LHS, return control to the user.
> 2. **Execute.** Perform the actions specified on the RHS of the selected production.
> 3. **Match.** Evaluate the LHSs of the productions to determine which are satisfied given the current contents of working-memory. Output is the conflict set.
>
> If **Halt**, return control to the user or go to step 1 [Digital 1988].

The Rete Match [Forgy 1982] is an algorithm for computing the conflict set. That is, it is an algorithm to compare a set of LHSs to a set of elements to discover all the instantiations. The cycle is more convenient to the user because when it ends, the conflict set is consistent with the current contents of WM. The algorithm exploits two properties common to all production systems in reducing the effort of performing the match. First, it exploits the fact that only a small fraction of working-memory changes in each cycle. Therefore, information about the LHS matches and partial matches in one cycle is saved and used in the next cycle, updating the information as necessary to reflect the changes made to working-memory on each cycle. Thus the amount of effort expended by the matcher depends primarily on the rate of change of working-memory rather than the absolute size of working-memory.

Second, it exploits the commonality between condition-elements of productions, to reduce the number of tests performed. Since the rules have to work together, they must be able to access the same working-memory elements. Hence their LHS must contain many similar conditions and terms if their patterns are to contend for access. Exploiting this requirement, Rete processes the patterns *before* the system is interpreted. The productions are processed by a rule compiler that locates these common terms and eliminates as many of them as possible. This allows many operations for the entire rule set to be done just once rather than doing them over and over again. These patterns are compiled into a special data-flow graph called a Rete[1] graph. The nodes in the graph designate computations to be performed by the matcher. The edges in the graph indicate the way the data is to flow from node to node through the graph. Redundant computations are eliminated by constructing only

[1] Rete means 'net'

a single node to perform a given a computation, linking the node to all the places where its result is needed.

3. 2. 6. Major Advantages of Production Systems

The production system offers a general framework for implementing search. The major advantages of production systems include:

> **Separation of Knowledge and Control.** The production system is an elegant model of separation of knowledge and control in a computer programme. Control is provided by the recognize-act cycle of the production system loop, and the knowledge is encoded in the rules themselves. This renders the knowledge base easier to modify without disturbing the program control code and, conversely, the program code can be modified without affecting the rule set.
>
> **A Natural Mapping onto State Space Search.** The successive states of working memory form the nodes of a state space graph. The production rules represent the set of possible transitions between states, while conflict resolution managing the selection of a branch in the state space. These factors simplify the implementation, debugging, and documentation of search algorithms.
>
> **Modularity of Production Rules.** There is lack of any syntactic interactions among production rules; rules may only effect the firing of other rules by changing the pattern in working memory. A rule does not invoke another one directly as if the latter were a subprogramme; nor a rule can set the value of variables in other rules. The scope of the variables of these rules is confined to the individual rules. This syntactic independence supports incremental development of applications by successively adding, deleting, or modifying the knowledge (rules) base.
>
> **Tracing Facility and Explanation.** The modularity of rules and the iterative nature of their execution make it easier to trace execution of a production system, by forcing each stage of the recognise-act cycle to display the selected rule. Moreover, since each rule represent a fragment of the problem-solving knowledge, each rule becomes self-explanatory regarding its part in the system as a whole.

3. 2. 7. Production Systems for GIS Design

Based on investigations to-date, two distinct classes of GIS have become perceptible. One class uses map-based data, especially in vector-format, and finds applications in engineering, town planning, decision support system, boundary analysis, and thematic representations. The second class uses image-based data, especially in raster format, and finds applications in image analysis and remote sensing.

The work reported in this thesis was an attempt towards designing a GIS of the first category, that is, a GIS based on data extracted from an authoritative map. As already stated, the characteristic feature of geographic data is that they are spatially indexed. Two basic alternatives are available in the construction of a data model that incorporates spatial addressing:

> Objects may be represented with each object having spatial location as an essential property
>
> Locations may be represented with each location being characterised by a set of object-properties.

These alternatives complement each other, and have resulted in the vector and tessellation models, respectively [Smith *et al.* 1987].

The spatial data model adopted for the present work is the vector model. The basic logical unit in a vector model is the line, used to encode the locational description of an object, and represented as a string of coordinates of points along the line. Closed areas, modelled as polygons, are represented by the set of lines that constitute their boundaries.

The vector model of geographic space can be classified as unlinked or topological. In the unlinked model, also termed the **spaghetti model**, each map entity is encoded separately in vector form without referencing any of its neighbouring entities. Spatial relationships are not encoded, rendering spatial analysis of data cumbersome. This approach, nevertheless, is adequate for routine analysis of geographic data, involving the distribution of objects and their display [Smith *et al.* 1987].

To recapitulate, the present work is aimed at

1. Designing a GIS based on data extracted from a planar map;
2. Representing each identifiable geographic primitive as a distinct object with its spatial properties defined;
3. Formulating composite spatial objects made up from primitives, based on rules defining such compositions.

The facts and rules are then organised into a production system, OPS5 to be specific, that can be used to produce thematic, or other mapping. It should also be capable of answering queries about locations of some class of spatial objects within a given spatial window; or about the identities of objects found within a given spatial window; or about the measurements such as, point-to-point distance, or perimeters, *et cetra*.

Even though GIS problems are well-defined, data in a GIS application are ill-structured. The use of production systems, in which control is exercised by choosing the most applicable rule in a knowledge base of rules, is a natural method for modelling such diverse tasks as the search of complex data bases of maps, reasoning about the data, and producing outputs in the desired format. The adopted scheme derives all the advantages of OPS5, pointed out in Section 3.2.6, like ease of modification, clarity of design (since the properties of objects are encoded directly into fact / rule), self-documenting code, *et cetra*. The benchmark studies conducted on several expert system tools, indicate that OPS5 is the fastest [Gevarter 1987], [Mettrey 1991].

3. 3. SUMMARY

THE COMPLEXITY of spatial objects suggests the applicability of AI techniques for representing the spatial objects, for developing procedures for reasoning about spatial objects, and for answering various queries about the objects. There are several ways of solving problems and representing knowledge in AI applications. Among them, production systems render the representation of knowledge systematic and modular. The implementation of control mechanisms is much more easier than in a procedural method.

The chapter has discussed the logical foundations of production sysems in general, with particular reference to OPS5. Even though GIS problems are well-defined, data in a GIS application are ill-structured. The use of production systems, in which control is exercised by choosing the most applicable rule in a knowledge base of rules, is a natural method for modelling such diverse tasks as the search of complex data bases of maps, reasoning about the data, and producing outputs in the desired format.

Chapter 4

Functional Units of a GIS Workstation

AS HAS been pointed out in Chapter 1, GIS technology finds application in a wide spectrum of problems, in research, in small or self-contained projects, and as an integral part of corporate activity and planning. Potential applications for GIS range from single, well-focussed ones like route planning, to computer-supported collaborative work systems wherein, groups of users, separated geographically, simultaneously develop a single cognitive map based on a problem/solution [Mennecke 1997].

Because the range and scope of the applications vary widely, configuring GIS workstations should take into account not only the wide range of *uses*, but also the wider range of *users*. The capabilities of GIS users vary from technically competent professionals interested in tackling complex problems, to individuals of below-average ability whose job is confined to repetitive, uninteresting tasks. For systems intended for tightly focussed applications in a stand-alone environment, an essentially static, uncustomisable user interface would suffice. The 'spatially aware professional', on the other hand, would rather work in a sophisticated GIS environment with complex visualisation user interfaces equipped with a large number of available commands [Raper & Bundock 1993]. It will therefore be prudent to examine the most comprehensive GIS which could set the standard for other systems.

4. 1. THE SEQUOIA 2000 PROJECT

THE requirements of a modern, state-of-the-art computing environment for global change researchers, scientists involved in studying ozone depletion, environment toxification, species extinction, simulating and forecasting world climate ten or even hundred years ahead, were formulated in June 1990, and a long-term project got underway in 1991. The focus of research under the Sequoia 2000 project was a high speed, broadband network spanning University of California (UC) campuses from Berkeley to Santa Barbara, Los Angeles, and San Diego; a massive database; massive storage; visualisation system; and electronic collaboration.

The participants in the Sequoia 2000 project included

Computer Science Division, UCBerkeley;
Computer Science Department, UCSan Diego;
San Diego Supercomputer Centre;
Department of Geography, UCSanta Barbara;
Atmospheric Science Department, UCLos Angeles;
Climate Research Division, Scripps Institution of Oceanography;
Department of Earth, Air, and Water, UCDavis;
Department of Water Resources, State of California;
Department of Forestry, State of California;
Coordinated Environment Research Laboratory, US Army;
NASA, USA,
National Oceanic & Atmospheric Administration, USA;
US Geological Survey;
Digital Equipment Corporation, Hewlett-Packard, Hughes Communications, MCI, Siemens, and TRW

The objectives of the project and the nature of its participants have made Sequoia 2000 the converging point of a great deal of technological resources. The status of the project as of 1995 has been described in [Stonebraker 1995], [Larson *et al.* 1995], [Kochevar & Wanger 1995], and [Pasquale *et al.* 1995].

4. 1. 1. Specifications for Sequoia 2000

The set of product requirements that were validated through discussions with the potential users included

A computing environment built on a data-centric system.
A database that handles a wide variety of non-traditional objects such as text, audio, video, graphics, and images.
Support for a variety of traditional databases and file systems.
The ability to perform necessary operations from computing environments that are intuitive and have the same look and feel. The interface to the environment should be generic, very high level, and easily tailored to the user application.
High-speed data migration between secondary and tertiary storage with the ability to handle very large data transfers.
Network bandwidth capable of handling image trasmission across networks in an acceptable time frame and error tolerance.
High quality remote visualisation of any relevant data regardless of format; the user must be able to manipulate the visual data interactively.
Reliable, guaranteed, delivery of data from tertiary storage to the desktop.

Used as a GIS, Sequoia 2000 has extensive capabilities for manipulating and displaying spatial information. It supports spatial data such as points, lines, and polygons. In addition, the DBMS supports the large spatial arrays in which satellite imagery is stored in its natural form. These characteristics are beyond the realm of popular, general-purpose relational and object-oriented DBMSs. The nearest setup that met the above requirement was either a special-purpose GIS or a next-generation OODBMS. Since POSTGRES (*vide* Pages 63, 68, & 76) was an object-relational system, the Sequoia 2000 project chose it as the database. It has since been extended to support basic spatial data processing tools that are adaptable to standard spatial models. Unlike many commercially available GIS software wherein, spatial data handling functionalities are provided as separate sets of executables operating on private data structures, Sequoia 2000 provides these functionalities as an integral part of the DBMS design.

Providing this level of functionality in the DBMS requires implementing geoprocessing primitives as POSTGRES operators and providing visualisation tools for examining the outputs. To address this, a set of representative queries has been developed. The queries are organised according to four basic spatial data types *viz.*, raster, network, polygon, point, singly as well as in combination, as follows:

raster:	multiple image selection by region, spectral analysis, resampling;
network:	graph traversal;
polygon:	clip to region, boolean selection by attribute;
point:	selection by region;
joins:	raster + polygon, raster + point, polygon + point

Presently POSTGRES supports coordinate data in both vector (polygon and line) and point formats. Raster data are represented as large objects, which are interpreted and processed, based on identifiers specifying number of dimensions, size, and organisation of arrays (row or tile), and pixel or cell format. Figure 4 1 provides an overview of the Sequoia 2000 geodata framework, highlighting the relationships among feature-based views and map coverage perspectives [Gardels 1994].

In spite of the enrichment of POSTGRES through augmented functionalities, it is bridled with the legacies of a typical relational database, particularly with respect to the lack of isomorphism that information on entities are subjected to, when they are dissipated over several tables during the process of normalisation. And the effect of this is most awkward in the case of geographic data.

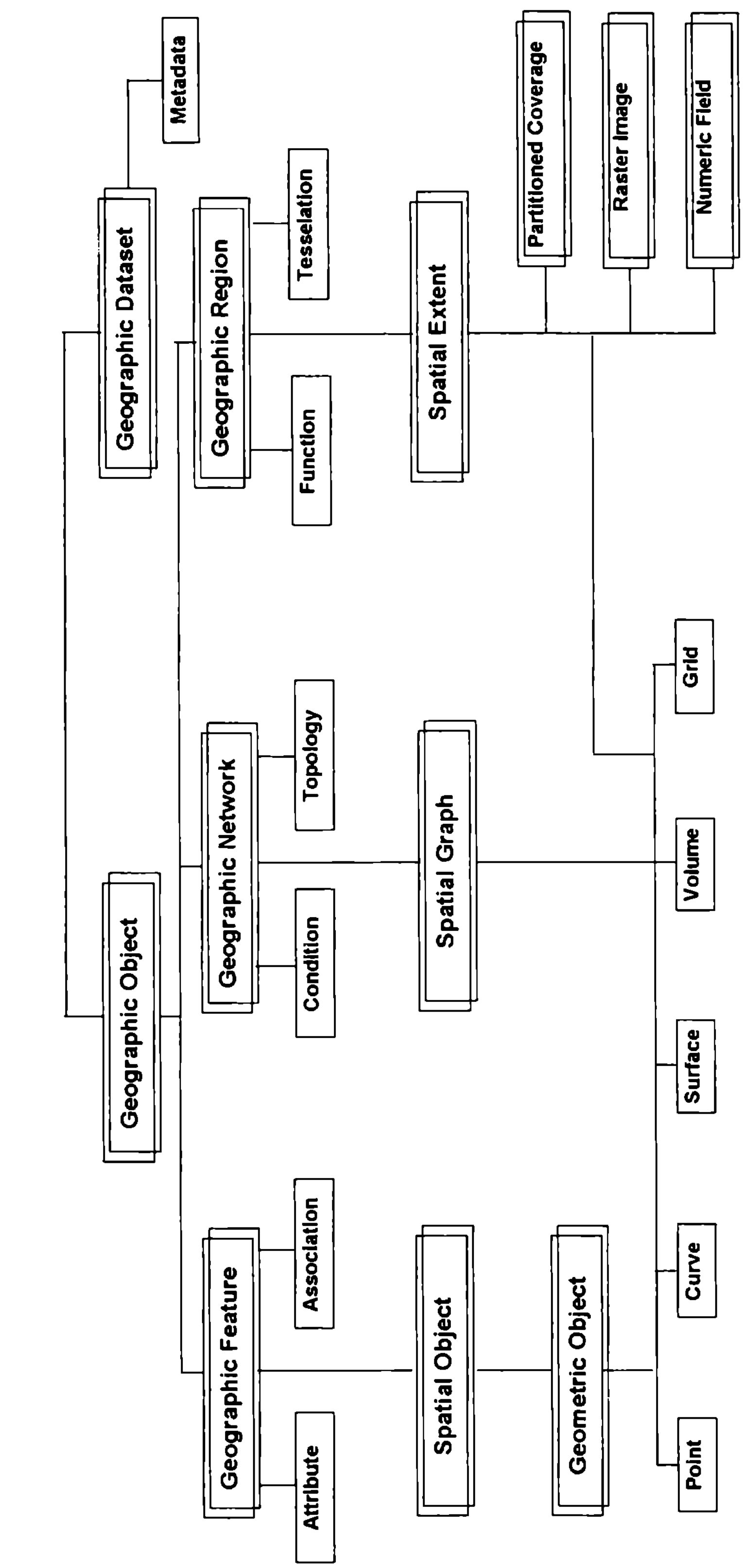


Figure 4. 1. The Geodata Frame Work Adopted for the Sequoia 2000 Project

Adapted from [Gardels 1994]

4. 2. SCOPE OF THE PRESENT WORK

AS INDICATED in Section 4.1, Sequoia 2000 is a project unmatched in its immensity of collaborative expertise and the sweep of technological features that are part of the project specification. The Sequoia project is relevant to the present work, insofar as it defines the state-of-the-art in GIS. For obvious reasons, the frame work adopted for KBGIS, could only be a small subset of that for the Sequoia 2000. It correspomds to the left-most branch of Figure 4.1 and, as shown in Figure 4.2, it focusses on spatial and geometric objects, *viz.*, points, lines, chains, curves, and polygons—objects that occur most frequently in common GIS applications such as route planning; urban/rural planning & development; resource allocations concerning schools, hospitals, police stations, *et cetra*; site selection of industries, schools, electrical transmission lines, and many others. The subset chosen can provide distribution of existing objects, and distance analysis. Since roads are treated as chains, with junctions acting as nodes connecting the short links, the shortest route from one location to another can be determined and reported.by the system. It can

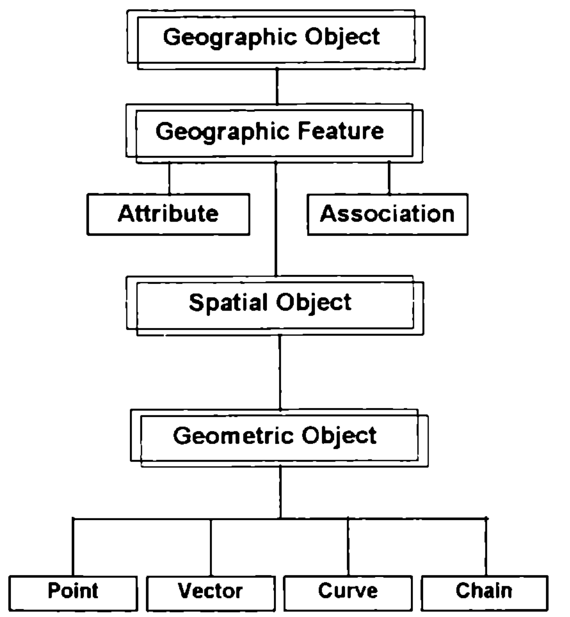


Figure 4. 2. The Data Frame Work Adopted for KBGIS

generate thematic maps using overlays. A schematic of the functional units of such a system is shown as Figure 4.3.

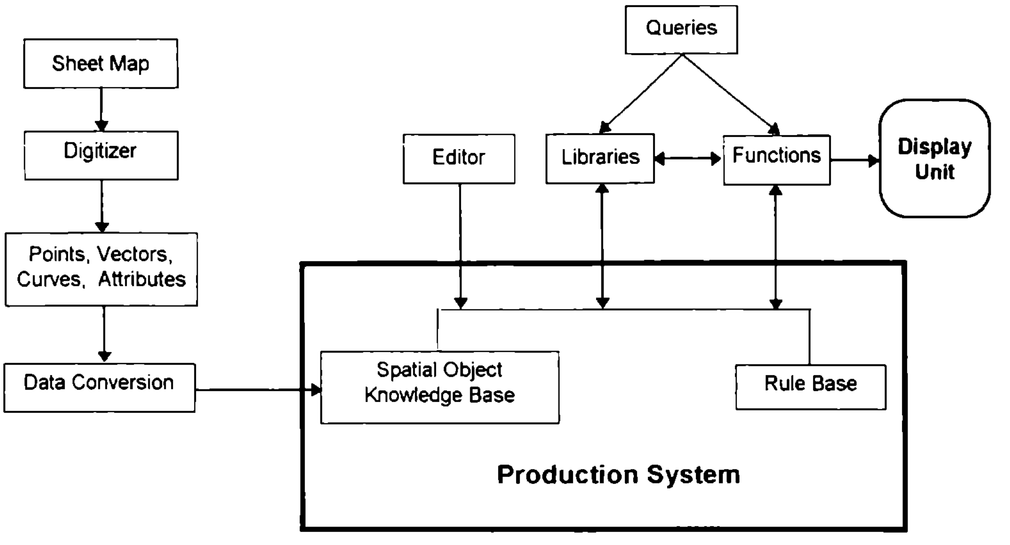


Figure 4. 3. Functional Units of a Knowledge-Based GIS

4. 2. 1. Specifications of the Proposed GIS

The requirements that are to be met by KBGIS are as follows:

1. The system will be designed as an experiment to study the feasibility of using a knowledge base in place of a relational data base that has been the tradition for most GIS designs.

2. The source of input spatial data is mostly sheet map that are to be manually digitised.

3. It should have suitable data handling facilities for converting the digitised data into formats suitable for storing as working memory elements (WME s) in a production system.

4. Points, lines, curves, and polygons, are to be handled appropriately, along with their relevant attributes.

5. An **editing** subsystem should be capable of editing the entries made as WME. The editing subsystem should also have the capability of inputing new digitised data into the working memory. It should also facilitate entering and editing the productions in the rule base.

6. The **query** subsystem should hold a number of standard **parametric query** statements, any one of which can then be

invoked by the user, either in the original form, or modified through changing the parameters.

7. The **functions** subsystem should have routines for customising and manipulating the screen layout, window layout, form layout, and for interacting with the GUI, so as to effect various display manipulations (zoom, pan, window sizing, scrolling, *et cetra*. The functions subsystem should also act as an interface between the query and the display subsystems.

8. A **library** subsystem should act as a repository for the various routines, queries, files containg the coordinate pairs of linear objects like roads, rivers, *et cetra*. When the query process finds as the result of a search, a road and its corresponding road-number, it could fetch the file of that road and have the coordinates and other relevant attributes of the road displayed on the screen.

The methodology adopted for realising the requirements defined by the above specifications are covered in the next chapter.

4. 3. SUMMARY

THIS CHAPTER has examined the features of a technologically sophisticated GIS, developed as part of the Sequoia 2000 project. The GIS is based on POSTGRES. POSTGRES functions essentially as a relational database system; but the relational model has been modified in the areas of data structure and object management. One goal of the project was the development of a multiterabyte storage system that would be available to the users over high-speed network.

A small subset of that system is chosen as the defining scope of the present work. The requirements of the proposed system have been specified. It is built around a production system, with separate knowledge base and rule base.

Chapter 5

Design Methodology

IT IS noted in Section 1 10 that the field of GIS is beset with the following difficulties:

> The majority of the current GISs are built around relational databases which have severe limitations in handling spatial data.
> Formulation of queries, even for the simplest of spatial operations in such GISs is cumbersome.
> Relational data model lacks the ability to specify explicitly the semantic information about relationships.

The motivation of the present work was to design a GIS that did not suffer from the drawbacks and limitations of the relational data model. Knowledge-based approach offers greater promise in realising that goal. In such an approach, a feature can be manipulated directly, and not *via* its attributes, as in relational models. An entity has an existence, separate from the descriptive thematic data. The knowledge content of the points, lines, arcs, polygons, and textual descriptions are stored in memory as rules that conform to the production system, OPS5. In formulating the rules, it is not necessary to make distinctions at the top level between spatial and aspatial components of the data. OPS5 can start with a set of overly specific rules and generate new rules that are more general and abstract. When the system over-generalises, it would backtrack or specialise some of the rules that are too general. The convenience and suitability of OPS5 in designing a knowledge-based GIS, have been explained in Chapter Three. This chapter explains the methodology adopted in the design of a knowledge-based GIS (KBGIS), as per the requirements specified in Chapter 4.

5. 1. Data Modelling in KBGIS

A VECTOR spatial data model is used for storing the graphic objects in the KBGIS where the basic unit is a pair of (*x*, *y*,) coordinates. Using these cordinate pairs, points, lines, and polygon (region) are represented. Thus, the map features can be identified in three basic graphic primitives:

Point: An (*x*, *y)* coordinate pair with a unique identifier

Boundary chain: A set of ordered coordinate pairs with a chain number, and the unique polygon codes of the polygon that it bounds on the left and on the right. Only boundary chains can be used later to form region features automatically

Network chain: These chains represent the linear features such as roads, rivers, main pipelines, *et cetra*. Every chain in the network is represented as a set of ordered coordinate pairs with a unique chain number In network chains, both the junctions and the dead-ends are stored as network nodes with numbers assigned. These nodes are then named and stored along with the associated attributes in the spatial object KB [Jayaprakash and Menon 1995].

To make the searching of point objects much faster, locational values of point objects are stored directly as vector points in the spatial object KB. The non-graphic attribute data about point, line, and area features are also associated with each object in the spatial object KB.

Thus, the data model adopted in the design of KBGIS can be summarised in the following three steps:

Firstly, it is a vector representation strategy because the geographic data entities are stored as a vector of coordinate points that define the geometry and topology of these entities precisely

Secondly, it is a logic based structure because two geographic entities, *viz.*, polygon and line are represented as logic clauses, more specifically as predicate structures describing certain relationship between a series of objects that are attributes of space.

Thirdly, it is a relational structure because the polygon and the line predicate structures can be interpreted as implementation of two relations (tables). Clauses of predicates can be seen as the relations' tuples. Polygon and line relations can be joined, when necessary, using common identifiers.

5. 2. THE ARCHITECTURE OF KBGIS

BASED ON the findings of previous research outlined in the earlier chapters, one can conclude that a good GIS should meet the following criteria.

> Ability to handle large, multi-layered, heterogeneous types of spatially indexed data;
> Ability to query such data sets interactively about the existence, location, and relevant properties of a wide range of spatial objects;
> Flexibility in configuring the system so as to accommodate a variety of applications and users;
> Ability of the system to learn, in a significant way, about the spatial objects in the knowledge and data bases during the use of the system.

The architecture of the knowledge-based GIS, reported in this work is shown in Figure 5.1

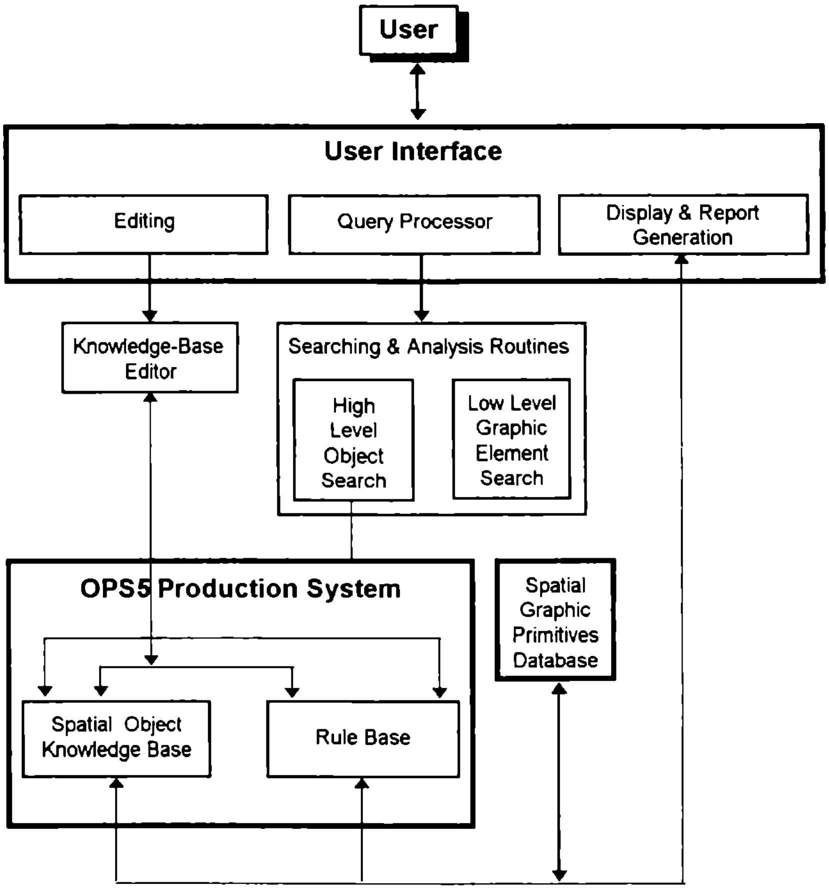


Figure 5. 1. Architecture of the Knowledge-Based GIS

5. 2. 1. Spatial Object Knowledge-Base

The spatial object KB contains knowledge about the high level spatial objects. The basic unit in the KBGIS implementation presented here is the entity representation, class. Each spatial information is treated as an object. An object should belong to one of the many classes which are pre-defined using the literalize declaration of OPS5 The class representation naturally captures the inherent structure of the data. The classes can be created at different levels of spatial resolutions, and information about the hierarchical links between classes can be stored along with classes themselves. Thus a spatial tree structure can be generated through class-subclass relationship formalism.

A class is defined with various attributes, and the instances of classes are stored as WMEs, in the spatial object knowledge-base, using the literalize declaration as in the following example:

```
(literalize education_inst
        type
        name
        location
        x_cord
        y_cord)
```

where "literalize" is the OPS5 declaration specifying the format of a WME that has attributes; "education_inst" is the name of the object, that has the attributes, "type", "name", "location", "x_cord", and "y_cord."

An instance of the above object "education_inst" can be created as an element in the working memory by the "make" action as follows:

```
(make education_inst
        ^type           college
        ^name           |University College|
        ^location       Palayam
        ^x_cord         1583.0
        ^y_cord         2385.0)
```

where the accent symbol ^ is used to denote an attribute that is immediately followed by its value.

Two different instances of the same object can be created similarly as follows:

```
(make  education_inst
       ^type           school
       ^name           |Cotton Hill High School|
       ^location       |Cotton Hill|
       ^x_cord         12148.0
       ^y_cord         8647.0)
```

and

```
(make  education_inst
       ^type           others
       ^name           |Ayurveda Research Institute|
       ^location       Poojapura
       ^x_cord         13844.0
       ^y_cord         6955.0)
```

In this manner, a spatial object knowledge-base of any arbitrary size, and consisting of any complex spatial objects can be created by the "literalize" declaration and the "make" action. Each element of the spatial object KB is thus a class (object) with a class name and a number of attribute value pairs. The attributes may be scalar or vector The value of a scalar attribute is an atom, and the value of a vector attribute is a list of one or more atoms.

5. 2. 2. The Rule-Base

While the information about the spatial objects and their definitions are stored in the spatial object KB, the information about search heuristics, object classification, object complexity, *et cetra,* are stored in the Rule-base, which, as stated in Chapter 3, is an unordered set of *if-then* rules, called productions.

To find and display all the schools known to the system, we could use the following production:

```
(p  find_school
       (education_inst
              ^type          school
              ^name          <schlname>
              ^location      <loc1>)
-->
       (wrte (CRLF)          <schlname>  <loc1>))
```

Here, "p" indicates a production, and "find_school" is the name of the production. The statement following the production name is the *condition element* of the production (LHS). This production is satisfied by all the objects in the spatial object KB whose class name is education and the value of the attribute "type" matches the pattern "school." If the production is satisfied, all the schools in the spatial object KB will be selected, and the action on the RHS will be executed whereby, the names and locations of all the schools known to the system get listed on the printer

Constants and variables are used as attribute values in the condition elements on the LHS of a production to constrain the the search. Constants can be strings, integers, or floating point numbers. A variable is enclosed in angle brackets < >

The production for the query for finding the location of the hotel named "Luciya," will take the form.

```
(p  find_hotel
        (hotel
                ^name           Luciya
                ^location       <loc>
                ^x_cord         <x_cord1>
                ^y_cord         <y_cord1>)
-->
        (write (CRLF)           <loc>  <x_cord1>  <y_cord1>))
```

Comparisons to be made for values that an attribute may take in a conditional element of a production is implemented using the relational operators, as illustrated below:

```
(block
            ^block_name    <blk>
            ^area          >= 200)
```

Conjunction is implemented by enclosing the list of conditional tests in braces {}, as shown below:

```
(block
                ^block_name     <blk1>
                ^area           <area1>
                ^population     {>20000  <30000})
```

Compute and external functions are implemented to perform mathematical calculations and to assign attribute values to the conditional element as in the following:

```
(block
        ^block_name     <blk1>
        ^population     {>20000  <30000}
        ^area           <a1>)

(place  ^place_name     <blk1>
        ^area1          (compute  <a1> — 1000))
```

5. 2. 3. The Query Processor

In query processing, the user can enter a query, either through the keyboard or by selecting options from the menu. The query entered is parsed and checked for syntactic correctness, and the user can modify the query, if required. The query processor module consists of Search and Analysis Routines. First the search routines are invoked which itself is divided into two levels of searching, *viz.*, the High Level Object Search, and the Low Level Graphic Element Search. In answering a query, the high level object search is invoked first.

5. 2. 3. A. High Level Object Search

The high level object search is performed on the spatial object KB, and is implemented using the OPS5 inference engine which, as explained in Chapter 3, is a forward chaining inference mechanism. Exiting from IE occurs after the match state, either because of an explicit halt or because there are no more rule instantiations in the conflict set. It is also possible to have a limit on the number of rule-firings by setting break points on particular rules.

If the match step produces a conflict set containing more than one rule instantiation, the conflict resolution is performed to select one instantiation for firing. In the KBGIS, the MEA strategy is used in this step. (Details and alternative strategies are given in Appendix B). With an instantiation selected, the execution phase is entered. In this phase, variables are bound to values and the action on the RHS of the production associated with the selected instantiation, is executed. When a given query is satisfied, the answers to the query can be output by invoking the Display and Report Generation Module.

5. 2. 3. B. Low Level Graphic Element Search

If the search on the spatial object KB is unsuccessful in producing the required answer to the query, then a low level graphic element search procedure is invoked for a detailed search on the spatial graphic primitive entity database. If the search turns out to be a success, the Display and Report Generation module will handle the output.

5. 2. 3. C. Multiple Queries

An important feature of the user interface of the KBGIS is that, in addition to the keyboard input and the menu-based input of queries, it also provides a set of pre-stored queries pertaining to the domain city, with facility for inputting optional parameters through submenus. This is achieved by selecting the *Standard Query* option of the Main menu (To be described in Chapter 6). Seventeen such standard queries have been identified, labelled, and stored. When the Standard Query option is selected, these "intermediate queries" will be listed on the screen, starting with Query 1, and the submenu associated with it, for effecting the choice of parameters. Other queries in the series can be browsed (and selected) by manipulating the vertical scroll button on the right side of the window Each one of the "intermediate queries" is associated with a submenu that enables the user to exercise different parameters that are appropriate to the query This feature enables a novice to quickly get the hang of navigating the GIS and get the required information. Some of the standard queries, and the submenus associated with them, are illustrated in Chapter 7

The exercise of options through parameters, serve another purpose *viz.*, reducing the clutter on the display screen. With several roads criss-crossing, placenames and location identifiers can get crowded. For certain queries, therefore, a priority on the roads to be displayed is enforced so that, the display can be controlled selectively, layer-by-layer The situation is the same with the placenames, and the same strategy is employed whenever display of placenames are called for The discussion on Query 1 in Chapter 7 should clarify the points further

5. 2. 4. Learning Module

The main purpose of learning procedures in the KBGIS is to reduce query search time. In a production system, incremental changes in performance can be effected by adding new production rules to an existing rule-base. Learning is accomplished in two ways, either by remembering the results of previous search or by learning the definition of an object more precisely so that the search space may be pruned rapidly The details of learning, implemented as an integral part of OPS5, is given in section B.7 of Appendix B.

5. 3. SUMMARY

THIS CHAPTER has enumerated the desiderata of a good GIS, borne out by the earlier researches in the field. The choice of the vector model for representing the geographical data for the particular application envisaged in the present work has been reasoned out. The relevance of using OPS5 in implementing the KBGIS has been highlighted. The chapter has described the architectural features of KBGIS, especially with respect to the rule-base, the spatial object facts- base, and the query processor

Chapter 6

Implementation of KBGIS

THIS CHAPTER will discuss the implementation details pertaining to the present work. Of the three graphic primitives mentioned in Section 5 1, only the Point and Network chain have been implemented, the Boundary chain was not found very relevant to the application presently on hand.

6. 1. DATA CAPTURING

KBGIS WAS prototyped using a map of Thiruvananthapuram City[1] (also known as Trivandrum City), the capital of Kerala State, and its environs, as the primary source of data. The map (Scale 1 125,000), authorised by Survey of India, was enlarged and brought to Scale 1:20,000 The enlarged map was divided into 64 × 64 blocks and each block was digitised using a digitizer of accuracy 0.025 mm, and coded separately so that any one of the small area covered by a single block could be displayed, when needed [Menon *et al.* 1996]. Close attention was paid to natural features like rivers, lakes, and coastal lines; physical features like railway, roads and their intersections, highways, airport, and public buildings, parks, schools, colleges, tourist spots, *et cetra.*

The digitized data that is stored as the map has a scale of 1:20,000. However, the resolution at this scale is too high, and hence awkward while displaying any meaningful area on the monitor For this reason a scale conversion routine has been

[1] Latitude 8° 29' N, Longitude 76°' 55' E (Source: Britannica Atlas © 1974 Encyclopaedia Britannica, Inc., Chicago, USA)

written, so that displays could be effected using either of the two standard scales, 1:24,000 and 1 100,000, suitable for display on the monitor, as well as for the output on the printer. The last two scales have become *de facto* standards for street maps and region maps in the United States [Morrison 1994].

6. 2. DATA DEFINITIONS

THE DATA model adopted for this work, as stated earlier, is the vector model. The data definitions for the different geographic objects are detailed below:

6. 2. 1. Network Chains

Entities coming under this category are: roads, rivers & canals, railwaylines, *et cetra*.

6. 2. 1. A. Road

Object Name	road
Cardinality	1-N
Type	Chain

Attribute Name	Type	Constraints	Cardinality	Description
1. rd-name	String	Alpha	1	Name of the road
2. rd-no	Integer		1-N	Identifying number of the chain
3. pr-no	Integer	1-4	1	Priority assigned to the road
4. st-point	String	Alpha	1	Location name where the road start (eg. Palayam)
5. end-point	String	Alpha	1	Location name where the road end
6. junction	String	Alpha	N	Location name of a junction along the road
7 st-xcord	Real	2 decimal places	1	*x*-coordinate of starting point
8. st-ycord	Real	2 decimal places	1	*y*-coordinate of starting point
9. end-xcord	Real	2 decimal places	1	*x*-coordinate of ending point
10. end-ycord	Real	2 decimal places	1	*y*-coordinate of ending point
11. rd-cord	String		1	Name of file holding list of road coordinates
12. rl-cros	Boolean		1	Crossing railway line or not
13. crplcrl	String	Alpha	0-N	Location name where the railway line crosses
14. rr-cros	Boolean		1	Crossing a river or not
15. rname	String	Alpha	0-N	Name of crossing river
16. crplcr	String	Alpha	0-N	Location name where the river crosses

Each road is identified with its name (rd-name). A road is treated as a linearly connected chain, interspersed with nodes. The starting and terminating points of the road, and the junctions along its way, are treated as nodes. The segment of the road from its starting point to the first node (usually a junction) is treated as a chain with the number 1 (Attribute 2, rd-no, integer data, cardinality 1-N). The last chain for the road is the segment between penultimate node and the terminating point of the road. A dead-end road with only one entry point will have only one chain. The ordered coordinate pairs of data for the entire road is maintained in a file by name rd-cord (Attribute 11), with separate references to its chains. This way, a road can be tracked from one junction to another during analysis for finding the shortest distance or for plotting the required sections of a road.

When a road crosses a river, Attributes 14 will get a True value and Attributes 15 & 16 get appropriate values, for each such crossings. The same narration holds valid when a road crosses the railway line, except that in this case, there is no equivalent to Attribute 15, as in the case of the road crossing more than one rivers.

6. 2. 1. B. Junction

Object Name	**Junction**
Cardinality	**1-N**
Type	**Node**

Attribute Name	**Type**	**Constraints**	**Cardinality**	**Description**
1. jn-name	String	Alpha	1	Name by which the junction is known
2. x-cord	Real	2 decimal places	1	x-coordinate of the junction
3. y-cord	Real	2 decimal places	1	y-coordinate of the junction
4. aprch-roads	String	Alpha	1-5	Name(s) of roads meeting at the junction

The fourth attribute (aprch-roads) has a cardinality of 1-5, so as to account for junctions which form large squares or circles.

6. 2. 1. C. *River (Includes canals)*

Object Name	**river**
Cardinality	**1-N**
Type	**Chain**

Attribute Name	Type	Constraints	Cardinality	Description
1. rrname	String	Alpha	1	Name of the river
2. st-point	String	Alpha	1	Location name where the river starts within the city
3. end-point	String	Alpha	1	Location name where the river ends within the city
4. rd-cros	Boolean		1	Crossing a road or not
5. rdcf	String		1	Name of file with list of crossing road coordinates
6. crplcrd	String	Alpha	O-N	Location Name where the road crosses
7 rr-cord	String		1	Name of file holding list of river coordinates

6. 2. 1. D. *Railway*

Object Name	**railway**
Cardinality	**1**
Type	**Chain**

Attribute Name	Type	Constraints	Cardinality	Description
1 rlname	String constant	Alpha	1	Name of railway "Southern Railway"
2. st-xcord	Real	2 decimal places	1	x-coordinate of starting point of railway within city
3. st-ycord	Real	2 decimal places	1	y-coordinate of starting point of railway within city
4. end-xcord	Real	2 decimal places	1	x-coordinate of ending point of railway within city
5. end-ycord	Real	2 decimal places	1	y-coordinate of ending point of railwaywithin city
6. rd-cros	Boolean		1	Crossing a road or not
7 rd-cros-rdf	String		1	Name of file holding coordinates of crossing roads
8. rl-cord	String		1	Name of file holding coordinates of railway

6. 2. 2. Point Objects

Under the point objects are well-known institutions, public buildings, offices, places of tourists interest, and hotels. Since their attributes differ due to the nature and functionalities of the objects, these are treated in separate categories. An important attribute regarding a point object is criteria. This attribute pertains to the locational characteristics of a point object with respect to other objects, point as well as chain types, in the domain. Activating a query, for example, to "search and identify the post office nearest to a specific hotel", the nearest-to criteria is employed in the production. Another query may involve searching "for all schools located between 1 km and 3 km" in which case, the criterion, between will be the right choice. When this criterion is selected in the query menu, the values for the lower and upper ranges (in km) will be sought from the user during run time.

6. 2. 2. A. Educational & Research Institutions

Object Class	**edn-research**
Cardinality	**1-N**
Type	**Point**

Attribute Name		Type	Constraints	Cardinality	Description
1	inst-type	String constant	"school", "college", or "others"	1	Restricted to one of three categories – school, college, others.
2.	name	String		1	Name of the institution
3.	location	String	Alpha	1	Name of location where institution is situated
4.	x-cord	Real	2 decimal places	1	x-coordinate of the location
5.	y-cord	Real	2 decimal places	1	y-coordinate of the location
6.	schl-no	Integer		0-1	Identifying number of school
7	clg-no	Integer		0-1	Identifying number of college
8.	oei-no	Integer		0-1	Identifying number of other institution
9.	criteria	String constant	"adjacent-to" "away-from" "between" "farthest", "nearest-to", and "within"	1-6	Used for tagging the object based on any of the six locational characteristics relative to other objects

Attribute 1 refers to the type of institution. An institution can be *one* of the three allowed types, *viz.*, school, college, or others. The cardinality is therefore 1 For an institution that is a school, attribute 6 (schl-no) will be loaded with the identification number of the school, and attributes 7 & 8 will be loaded with 0s. When the institution is a college, attribute 7 will be set to the identification number of the college, setting the remaining two to 0s.

6. 2. 2. B. Public Buildings and Offices

Object Class	**building-office**
Cardinality	**1-N**
Type	**Point**

Attribute Name	Type	Constraints	Cardinality	Description
1. office-type	String constant	"Govt" or "Other"	1	Type of building or office
2. name	String	Alpha	1	Name of the office
3. location	String	Alpha	1	Name of location where situated
4. x-cord	Real	2 decimal places	1	x-coordinate of the location
5. y-cord	Real	2 decimal places	1	y-coordinate of the location
6. criteria	String constant	"adjacent-to" "away-from" "between" "farthest", "nearest", and "within"	1-6	Used for tagging the object based on any of the six locational characteristics relative to other objects

6. 2. 2. C. Tourist Places

Object Class	**tourist- place**
Cardinality	**1-N**
Type	**Point**

Attribute Name	Type	Constraints	Cardinality	Description
1 tp-name	String	Alpha	1	Name of tourist place
2. tp-no	Integer		1	Identifying number of the tourist place
3. location	String	Alpha	1	Name of location where situated
4. x-cord	Real	2 decimal places	1	x-coordinate of the location
5. y-cord	Real	2 decimal places	1	y-coordinate of the location
6. criteria	String constant	"adjacent-to" "away-from", "between" "farthest", "nearest" and "within"	1-6	Used for tagging the object based on any of the six locational characteristics relative to other objects

6. 2. 2. D. Hotels

Object Class	**hotel**
Cardinality	**1-N**
Type	**Point**

Attribute Name	Type	Constraints	Cardinality	Description
1. name	String	Alpha	1	Name of the hotel
2. type	Char	"5", "4", or "3"	1	Hotel category: 5 star, 4 star, or 3 star
3. x-cord	Real	2 decimal places	1	x-coordinate of the location
4. y-cord	Real	2 decimal places	1	y-coordinate of the location
5. m-type	String	"veg" "non-veg"	1-2	Type of food served
6. r-type	String	"ac" "nonac"	1-2	Type of room availability
7. htl-no	Integer		1	Identifying number of the hotel
8. criteria	String constant	"adjacent-to" "away-from" "between", "farthest", "nearest", and "within"	1-6	Used for tagging the object based on any of the six locational characteristics relative to other objects

6. 2. 2. E. Places

Object Class	**place**
Cardinality	**1-N**
Type	**Point**

Attribute Name	Type	Constraints	Cardinality	Description
1 pname	String	Alpha	1	Name of the locality, e.g., East Fort, Palayam
2. x-cord	Real	2 decimal places	1	*x*-coordinate of the location
3. y-cord	Real	2 decimal places	1	*y*-coordinate of the location
4. priority	Integer	1-4	1	Indicates importance of the place; used for display
5. criteria	String constant	"adjacent-to" "away-from" "between", "farthest", "nearest", and "within"	1-6	Used for tagging the object based on any of the six locational characteristics relative to other objects

6. 3. HARDWARE USED

KBGIS HAS been implemented as a prototype on a MicroVAX II™ AI workstation, running on VMS™[1] operating system. The workstation had the following configuration:

System Type:	VAX station II / GPX
Processor type:	MicroVAX II, KA630, 32-bit
Firmware:	VAX Microcode emulation
Processor clock.	23.4 MHz
Memory (RAM)	16 MB
Operating system:	VAX / VMS Version 5.2
Monitor:	VR290 19" colour
Video format	RGB colour, composite video
Resolution:	1024 × 864 pixels
Picture size:	326 × 275 mm
Hard Disk:	**RD 54**
Total capacity:	191 MB
User capacity	159 MB (Formatted), 512 bytes/sect, 17 sect/track
Transfer rate:	5 MB/s
Tape Drive:	**TK50**
Total capacity	131 MB
User capacity:	94.5 MB (Formatted)
Recording density:	16,934 bits / cm
Tape speed:	190.5 cm / sec
Streaming data rate:	500 KB/s
Digitizer:	**KD4610**
Effective digitizing area:	460 × 310 mm

[1] MicroVAX II and VMS are registered trade marks of Digital Equipment Corporation, U.S.A.

Total capacity:	131 MB
User capacity:	94.5 MB (Formatted)
Recording density:	16,934 bits / cm
Tape speed:	190.5 cm / sec
Streaming data rate:	500 KB/s
Digitizer:	**KD4610**
Effective digitizing area:	460 × 310 mm
Resolution:	0.025 mm
Point reaching accuracy:	± 0.25 mm
Transfer rate:	65 data/s (maximum)
Colour Printer:	LJ250 companion inkjet printer
Resolution:	180 × 180 dpi (Text, colour graphics)
Printing speed:	90 characters (167 burst) per sec at 10 cpi (B&W text) 36 characters per sec at 10 cpi (colour text) 16.7 inches per sec (colour graphics)
Mouse	
Resolution:	79 counts / cm
Tracking speed	73.5 cm / sec
Baud rate	4800
Accuracy	± 3 % 0 - 24.5 cm / sec in any direction ± 15 % 24.5 - 49.0 cm / sec in any direction ± 30 % 49.0 - 73.5 cm / sec in any direction

6. 3. 1. Workstation Software

The workstation supports a proprietary graphics user interface called User Interface Services (UIS), the features of which can yield a high degree of user friendliness. This utility provides windowing facility KBGIS is built with the help of OPS5. Specific user interfaces and other utilities for the GIS were designed using VAX Fortran and C programming languages, which came along with the workstation.

6. 4. KBGIS SOFTWARE

ALL THE user interaction with KBGIS are controlled by event-driven commands based on menus, and in few specific cases, through the keyboard input. The various menus, and the characteristics of the modules connected with each one of the sub-menus are detailed in the following section and illustrated in Figures 6.1 through 6.4.

6. 4. 1. The Start Menu

The menu presented as soon as a successful logging-in is over, is the "Start" menu that has three items (sub-menus). An item can be selected by moving the mouse over it and clicking the left mouse button. For all the windows **below** this highest level, clicking the right mouse button returns the command to the immediately previous menu / sub-menu.

6. 4. 1. A. Map Input

The data input module of KBGIS, called "dig," enables the user to input data and organise it in a form suitable for the database. This module could be invoked either within the menu structure or outside it. The command **dig** allows the user to input a new map or part of the old map. The user will be prompted for input of the object names and filenames. Points, lines and area features of the map can be digitized and organised according to the data model consistent with that adopted in KBGIS.

6. 4. 1. B. Exit

This item, when selected, takes the control out of KBGIS and the opening window of the system will be presented.

6. 4. 2. The Kerala Districts Menu

All the fourteen districts of the state are shown; but only the item "Thiruvananthapuram district" is implemented at present. By clicking the left mouse button anywhere within the boundary of Thiruvananthapuram district, the sub-menu of Thiruvananthapuram City will be rolled down.

6. 4. 3. The Thiruvananthapuram City Menu

This has seven items, including one for queries predicated on value or range of values of a chosen parameter. At the top of the screens for all the sub-menus *below*

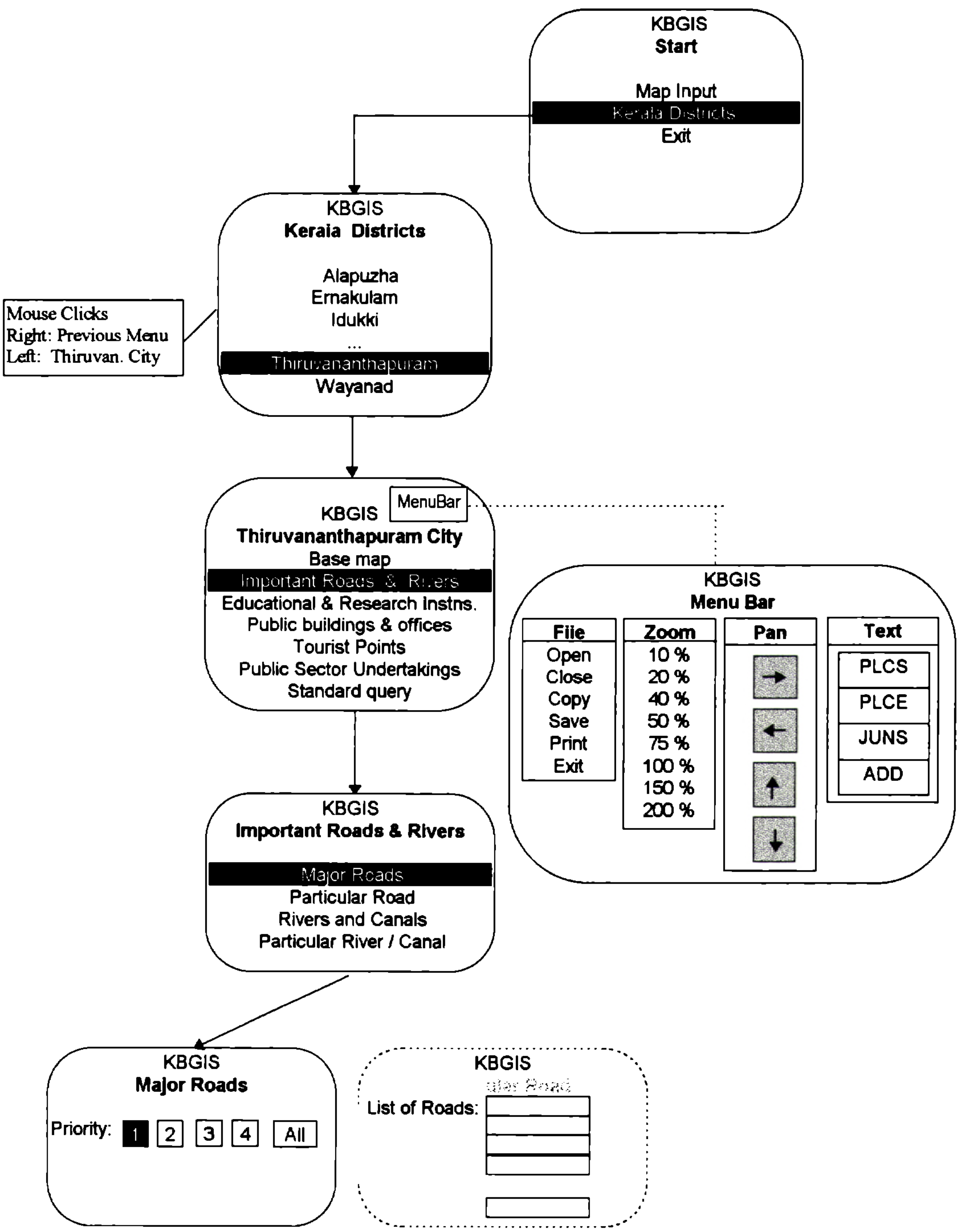


Figure 6. 1. A Segment of the Menu Operations in KBGIS

the level of this menu, *viz.*, Thiruvananthapuram City, will carry a pull-down **Menu Bar.**

6. 4. 3. A. Menu Bar

As shown in Figure6.1, the menu bar has four menus, *viz.*, File, Zoom, Pan, and Text.

File Menu. The File menu provides the user with facilities for editing a new file or a file that is already known to KBGIS, and save the edited file while remaining within the KBGIS environment. The Open and Close commands of this menu let a user open a file in the memory, continue working on it. When he decides to close the session, he clicks the left mouse button on the Close command. At this point, the map will be deleted, the screen cleared, and the control brought back to the previous menu, *viz.*, the Thiruvananthapuram City menu.

The Copy command of the File menu allows the user to suspend the working on a map he has been currently engaged in. The Copy command will keep the current map as an **icon.** The screen will be cleared except for the small icon, displayed at the bottom of the screen on the left corner The user could very well start a new session by opening another file. At any time, the user could bring back the first map that was copied, by clicking the left mouse button on the corresponding icon.

The Save command of the File menu allows the user to save the map that is currently under use as a UIS file (Refer Section 6.3.1). The user will be prompted for the *filename* in which the file is to be saved.

The Print command enables the user to take hard copy print out of the mapthat is currently being worked at.

Zoom Menu. This menu allows the user to zoom up the map, or a selected area within the map. The zoom window is selected by placing the mouse pointer at the top left corner of the intended window, clicking the left mouse button, dragging the mouse pointer to the bottom right corner of the intended window, and clicking the left mouse button again. The area thus selected can be zoomed by one of the zoom ratios selected from the menu. The enlarged view of the selected area will be displayed, laid over the original map. The zoomed view can be deleted by clicking the right mouse button while keeping the pointer anywhere over the zoomed map

Pan Menu. This menu allows the user to pan through a large map in small steps, in all the four directions, by clicking the left mouse button on one of the four icons representing the particular direction desired By clicking the right mouse button, the original state of the map could be restored.

Text Menu. The commands in this menu provide the facilities for adding place names, junction names, *et cetra* on the map. There are four commands: **PLCS**, **PLCE**, **JUNS**, and **ADD**. The PLCS command lets one to see more than one place name on the city map, on a priority basis. A priority sub-menu will be pulled down on the screen showing a list of four priorities, 1 to 4 (1 being the highest). The priority option allows one to view (or print) a map without being overcrowded with place names. When a particular priority is chosen, the place names that are assigned the selected priority will be displayed on the map. The place names belonging to a priority already chosen, can be deleted by placing the mouse pointer over the particular priority item and clicking the right mouse button.

The PLCE command allows the user to view a given place name, by having it displayed on the appropriate spot on the map. In this command, however, the user will be prompted to enter the particular place name in the dialog box.

The JUNS command works in a manner similar to that for PLCS, except that the operations pertain to street junctions within the city, instead of place names. The priority options are also applicable for this command.

The ADD command makes allowance to add new objects and placenames in KBGIS interactively

6. 4. 3. B. Base Map

This will display a map of the city with all the transportation network such as roads, highways, railway, rivers, and canals. This is intended to provide an overall view of the layout of the domain, viz., the target city

6. 4. 3. C. Important Roads & Rivers

This command allows the user to display the major roads in the city, on a priority basis. Clicking on this item in the City menu, pulls down the sub-menu Important Roads & Rivers, which has the following four items:

Major Roads
Particular Road
Rivers and Canals
Particular River / Canal

- Major Roads
- Particular Road
- Rivers and Canals
- Particular River / Canal

Clicking on the Major Roads opens a priority menu with four options corresponding to the four priorities (1 - 4) and the"All" option which overrides the priority and forces the display of all the roads. As indicated in the paragraph for Text Menu in Section 6.4.3.A, enforcing this priority will reduce the cluttering on the screen with too many details.

After the roads are displayed, a dialogue window opens up on the screen, showing the message on the right. If overlay is opted for, information retrieved subsequently, will be overlayed with the previous information.

Selecting Particular Road from the sub-menu, will open another sub-menu, showing a list of roads that are available for display. The list can be scrolled up or down using the combination keystrokes, CTRL+ U or CTRL+D, respectively. When a road is selected from the list and clicked upon, it will be displayed, along with the dialogue window for the "overlay option."

The menu items, Rivers and Canals, and Particular River/Canal behave *mutatis mutandis* in ways similar to those concerning the roads.

It should be noted that the facilities available on the Menu Bar can also be used for exercising various features over the displays.

6. 4. 3. D. Educational and Research Institutions

The rôle of this command in the Thirivananthapuram City menu is to pull down the sub-menu with three items, Colleges, Schools, and Others. When College is selected, the sub-menu for College will be presented; and this has two items:

Distribution of Colleges;
Particular College

Selecting the first item, a map of the city with the names of all the colleges and their locations will be displayed. If the "Overlay Information" option has still been active from the previous selection, the new map will be laid over the previous one and the combination map will be displayed. Once again, the "Overlay Information" prompt will be displayed, allowing the user to have a further set of objects laid over the current combination map.

Selecting the second menu item, *viz.*, Particular College, will display a list of all the colleges in the city from which, the user could select the desired one. Scrolling of the list up or down is again through the combination keystrokes, CTRL+U and CTRL+D, respectively

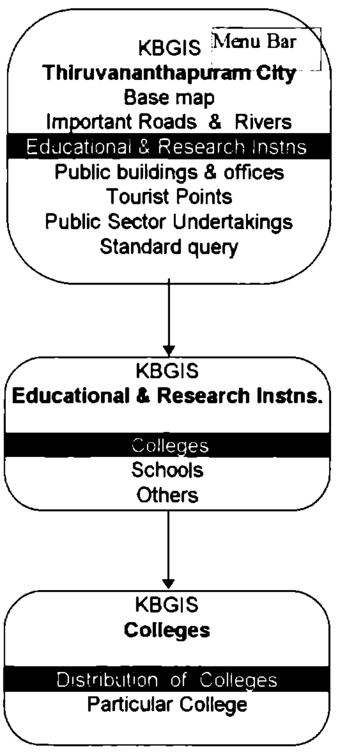


Figure 6. 2. Partial Menu Sequence for Educational & Research Institutions

The other two items of the Educational & Research Institutions menu, Schools and Others work in similar ways.

6. 4. 3. E - G. The Rest of the Commands

An overview of the flow of control through the menus for the Thiruvananthapuram City, except the Tourist Points, Public Sector Undertakings, and Conditional Query, are shown in Figure 6.3 Those relating to Tourist Points, Public Sector Undertakings, and Conditional Query, are shown separately as Figure 6.3 1, due to the large size of a combined figure.

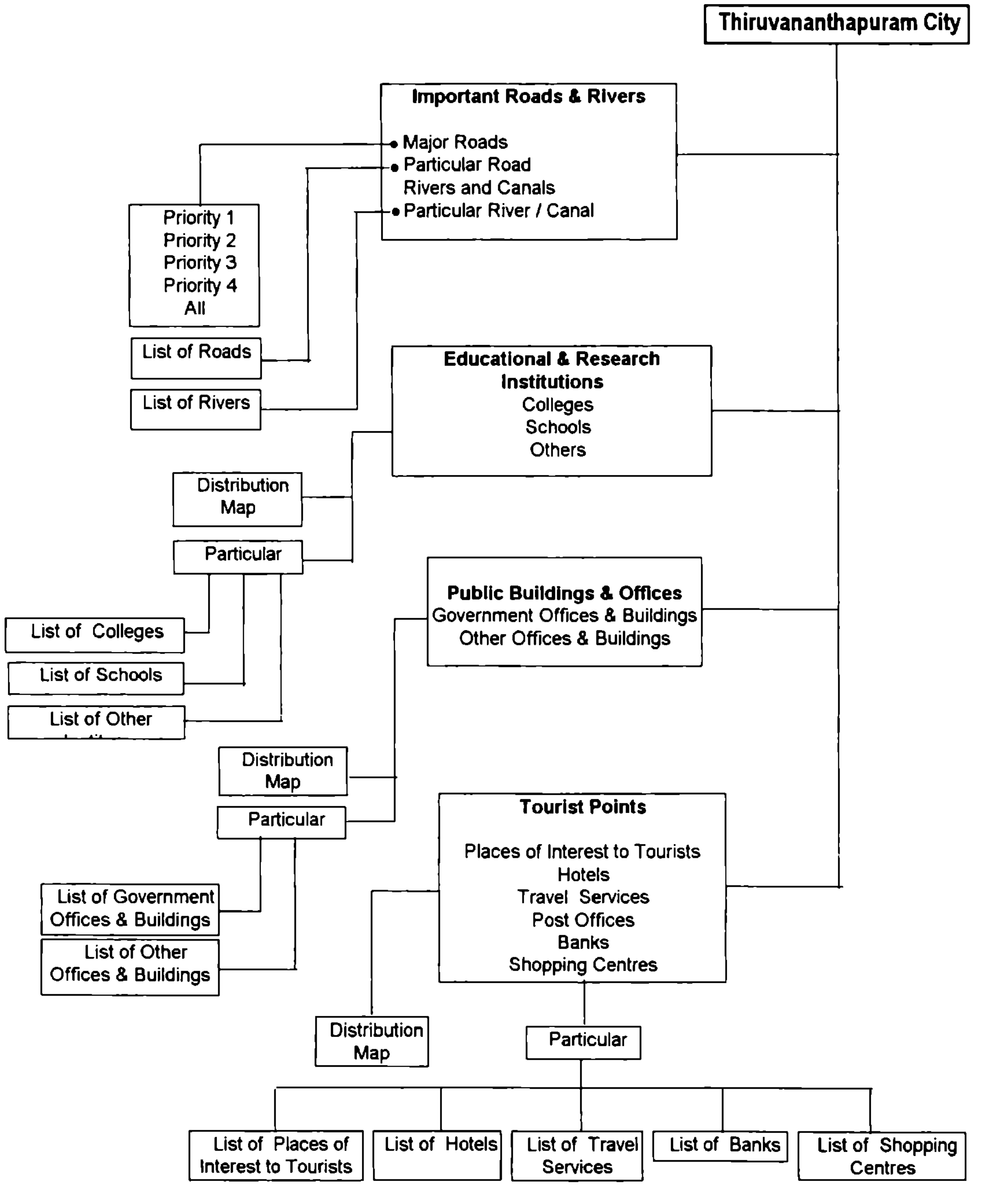


Figure 6. 3. An Overview of Menu Structure for Thiruvananthapuram City

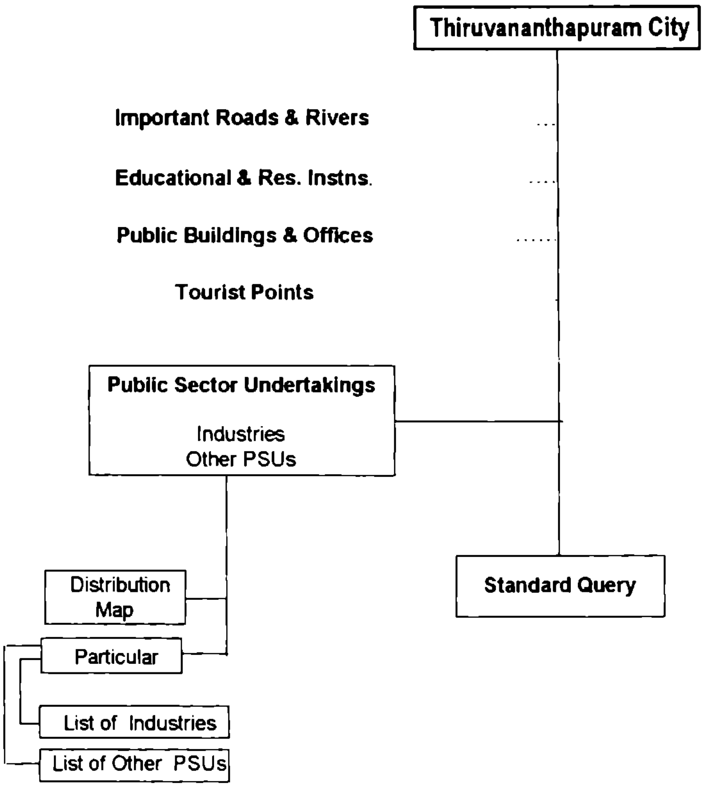


Figure 6. 3. 1. Another Partial Overview of Menus for Thiruvananthapuram City

6. 5. SUMMARY

A DISCUSSION on the query processing, the characteristics of the various menus and their hierarchical structers have been presented in this chapter. The details of the data definitions used in the implementation of KBGIS are also given.

Chapter 7

Results and Discussion

THE PRESENT implementation of KBGIS is based on 83 rules and 490 objects. Few of the plates that are output by the system, in response to some of the queries presented to it, are shown in the following pages. A narration of the query, the salient features of the corresponding production which processed the query are also presented for each plate. The background for all the output screens are white, due to problems with the colour printer As the latter is of low resolution (180 × 180 dpi), continuous areas in colour show awkward striations.

7. 1. QUERY 1

7. 1. A. Input Screen

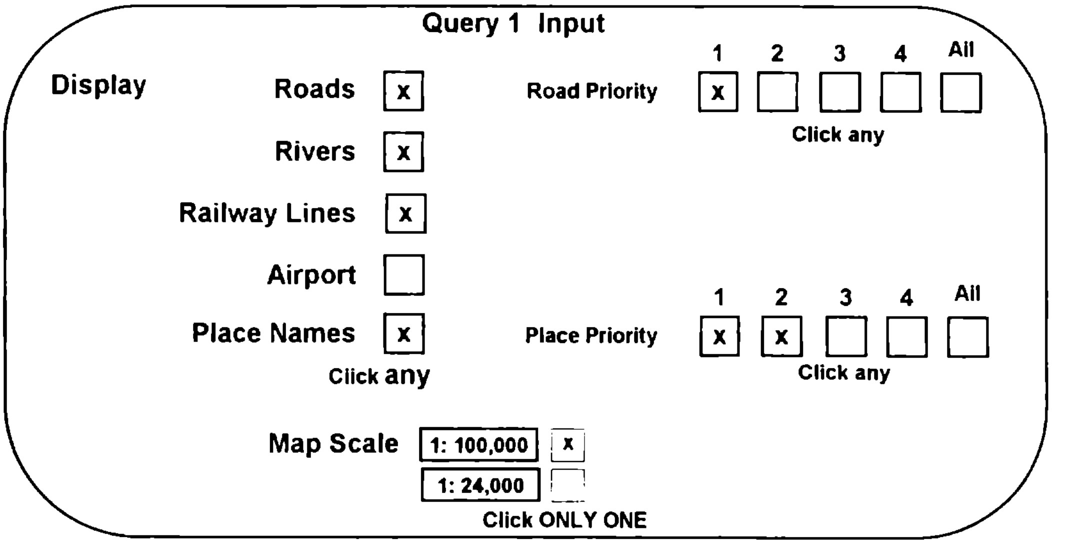


Figure 7. 1. Screen for Selecting Option Parameters for Query 1

Thiruvananthapuram City and Environs

File Text Pan Zoom Window

Query Window

Qry1 Display the Transportation network of Thiruvananthapuram city with roads (priority 1), rivers, railway lines and place names (priority 1 , priority 2)

Result Window

Roads-Priority 1

National Highway

Anayara Road

Kovalam Road

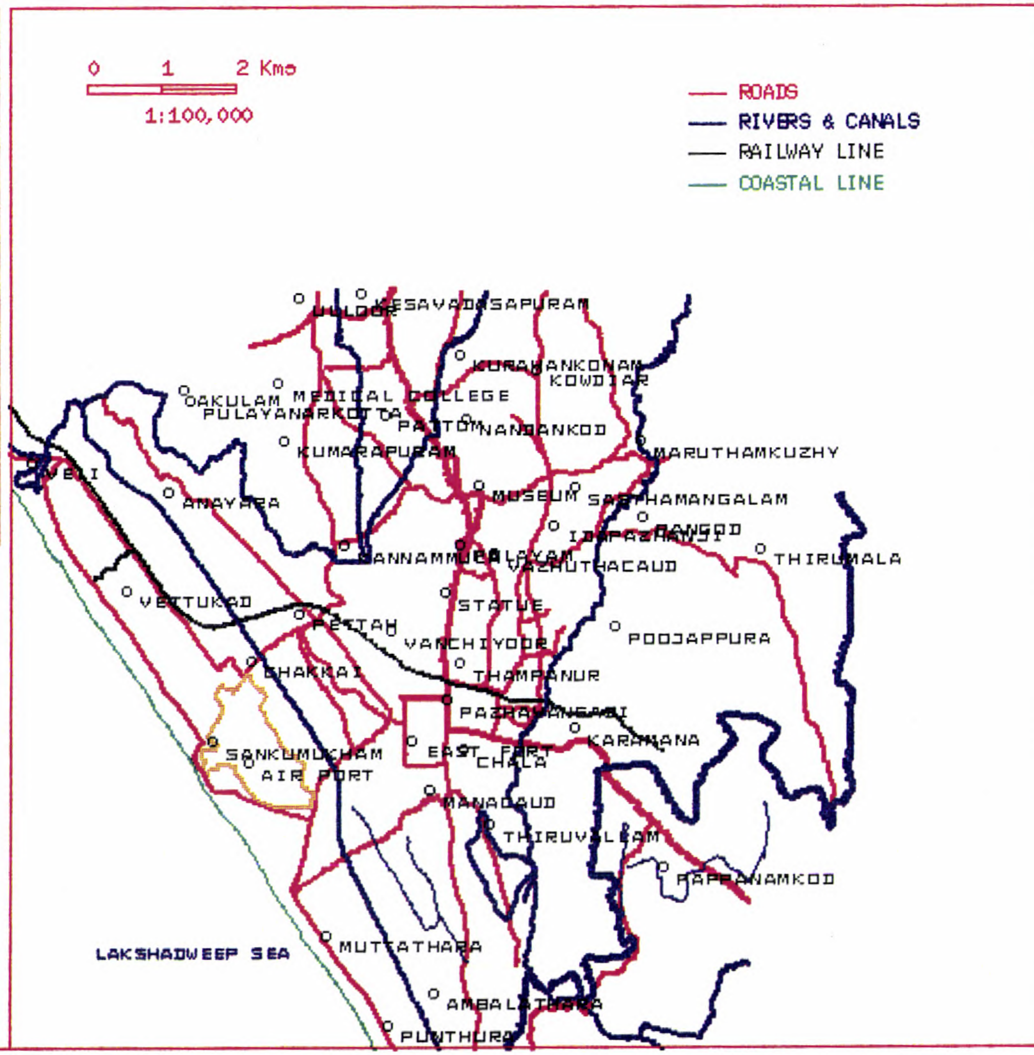
Airport Road

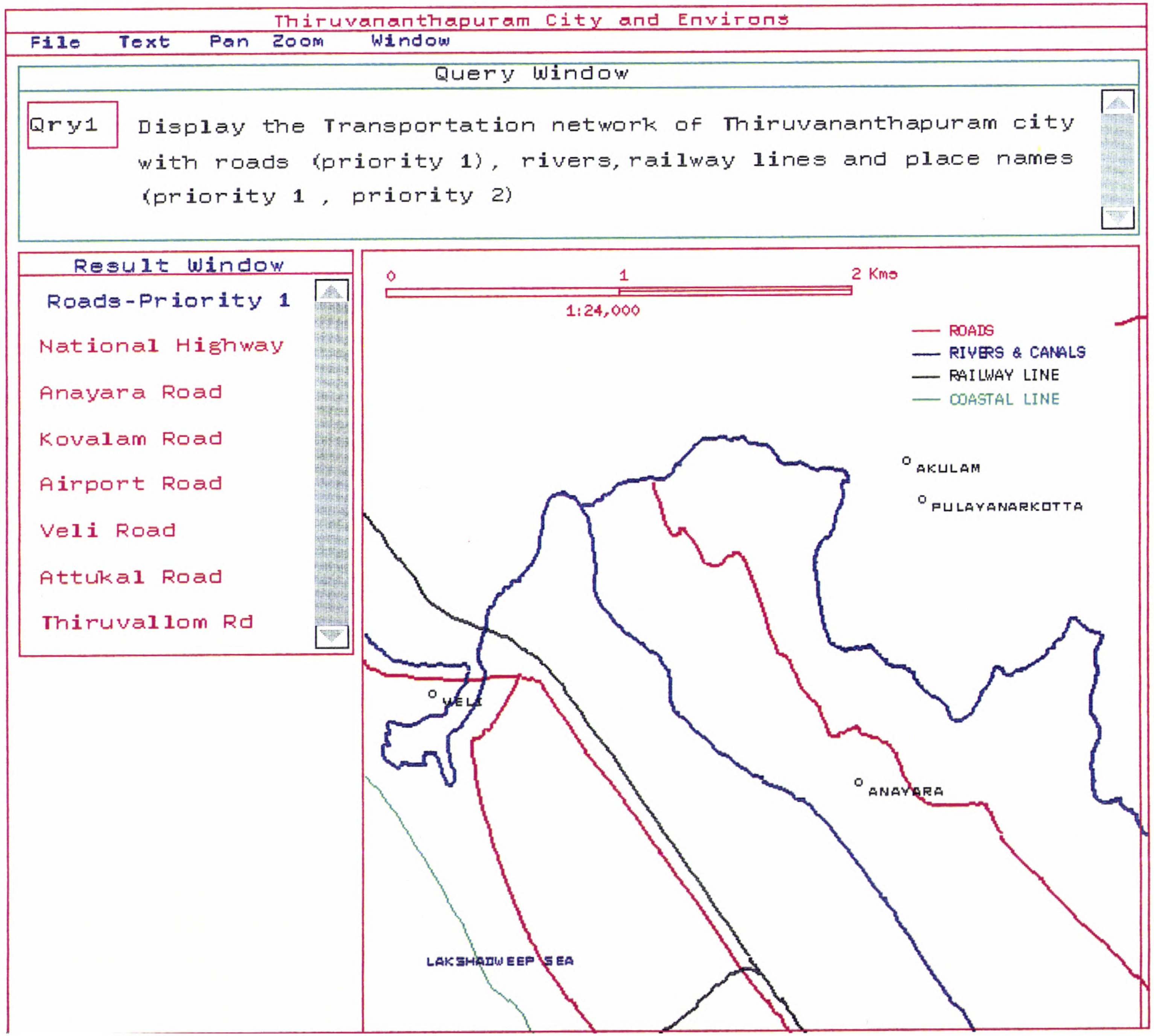
Veli Road

Attukal Road

Thiruvallom Rd

Click the item to isolate it on the Map





Thiruvananthapuram City and Environs
File Text Pan Zoom Window
Query Window
Qry1
Display the Transportation network of Thiruvananthapuram city with roads (priority 1), rivers, railway lines and place names (priority 1 , priority 2)
Result Window
Roads-Priority 1
National Highway
Anayara Road
Kovalam Road
Airport Road
Veli Road
Attukal Road
Thiruvallom Rd
0
1
2 Kms
1:24,000
ROADS
RIVERS & CANALS
RAILWAY LINE
COASTAL LINE
AKULAM
PULAYANARKOTTA
ANAYARA
VELI
LAKSHADWEEP SEA

7. 1. B. Plate 1

Query Objective: To display **railway line, rivers, canals**, and **selected roads** that pass through Thiruvananthapuram City, with important place names marked.

Query Options: Roads: **Priority 1;**
Placenames: **Priority 1 & 2**

Production Name: **find-roads-rivers-rail**

Production:

```
(p  find-roads-rivers-rail
(alrdrrl          ^rrlname          name)
        (road     ^rd-name          <rdname>
                  ^st-xcord         <st-xcord>
                  ^st-ycord         <st-ycord>
                  ^end-xcord        <e-xcord>
                  ^end-ycord        <e-ycord>
                  ^rd-cord          <rd-cord>
                  ^rd-no            <rdn>
                  ^pr-no            <1>)
        (river    ^rrname           <rrname>
                  ^rr-cord          <rr-cord>)
        (rail     ^rlname           <rlname>
                  ^rl-cord          <rl-cord>)
        (place    ^pname            <pname>
                  ^x-cord           <x-cord>
                  ^y-cord           <y-cord>
                  ^priority         <<1 2>>)
-->
        (call rdplotall             <rdname>      <rd-cord>      <rdn>
                                    <st-xcord>    <st-ycord>)
        (call rrplotall             <rrname>      <rr-cord>)
        (call rlplotall             <rlname>      <rl-cord>)
        (call plotpointp            <pname>       <x-cord>       <y-cord>))
```

Explanation:

Here, find-roads-rivers-rail is the name of the production. The element

```
(alrdrrl          ^rrlname          name)
```

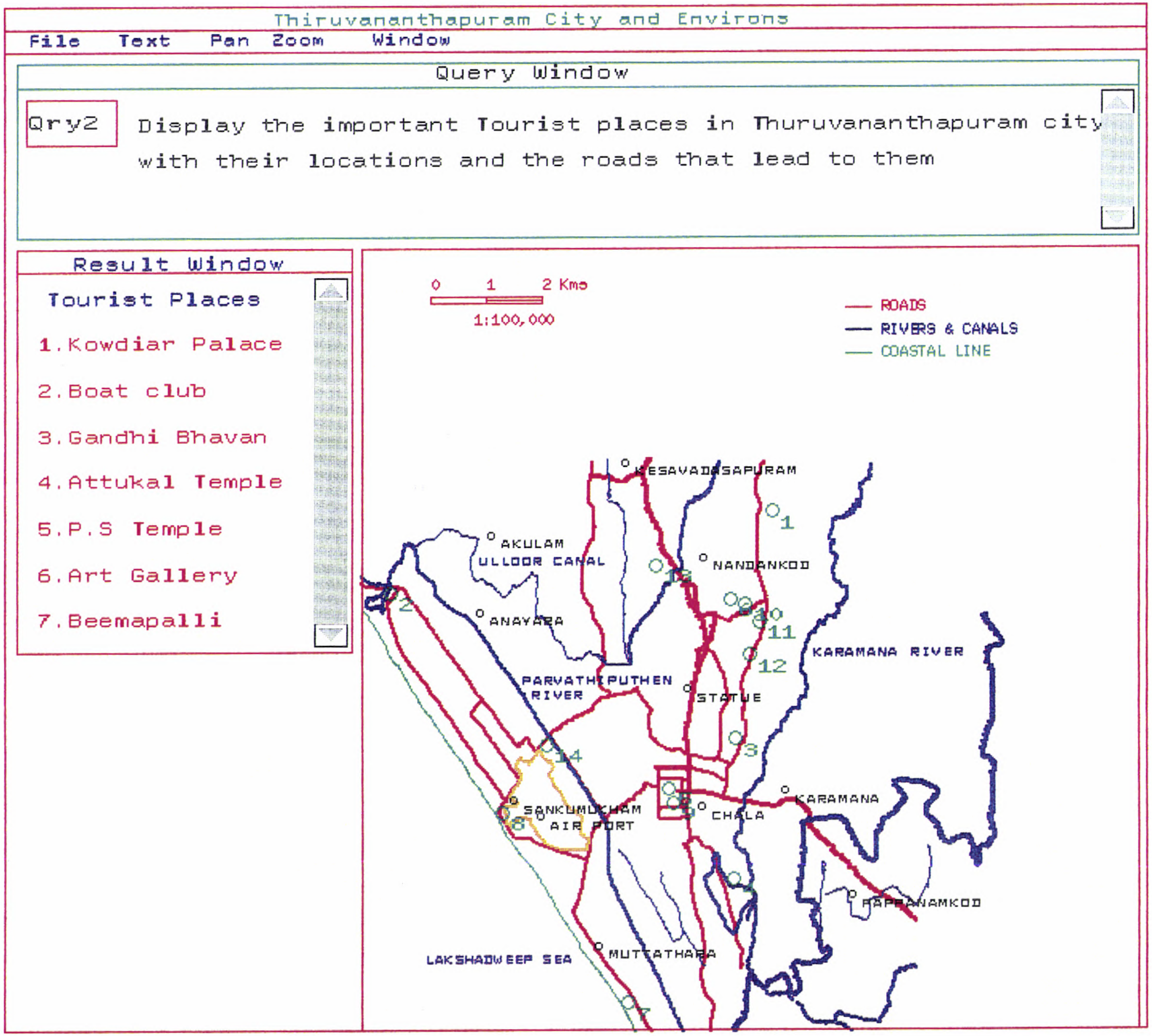
is used as the control element, (also called context element) for selective activation of the rules in the context. This element has been declared by the **literalize** declaration of the OPS5 programme.Since OPS5 treats all data as symbolic or integer atoms, the individual strings in the control element must be converted to symbolic atoms before it is finally placed in the working memory. The program statements that perform these operations are shown in Appendix D.1

The attribute ^pr-no (on the tenth line of the definition) indicates the priority number of the roads. Since the priority is <1>, roads with "priority value" of 1 will be selected. This will eliminate the clutter of the display by limiting the the number of roads to be displayed. Priority 1 refers to the most important object, 4 refers to the least important. If all the roads are to be seen, priority value "All" is to be selected.

In the action part of the production, various procedures, external to OPS5 and written in Fortran, are invoked. Since the procedure calls are made *from* OPS5 environment, *to* a different, external, language environment, the arguments, *i.e.*, the strings enclosed within angle brackets, used in the procedure calls must be converted from the symbolic atoms to the appropriate data types before they can be processed. The routine that handles this conversion is shown in Appendix D.2.

The strings rdplotall, rrplotall, rlplotall, and plotpointp, are all procedure names, and the strings enclosed within the angle-brackets are the arguments to the respective procedures.

Scale:	1:100,000 1:24,000 also shown upon zooming selected area
Response Time:	11 seconds


Thiruvananthapuram City and Environs
File Text Pan Zoom Window
Query Window
Qry2
Display the important Tourist places in Thuruvananthapuram city
with their locations and the roads that lead to them
Result Window
Tourist Places
1.Kowdiar Palace
2.Boat club
3.Gandhi Bhavan
4.Attukal Temple
5.P.S Temple
6.Art Gallery
7.Beemapalli
0 1 2 Kms
1:100,000
ROADS
RIVERS & CANALS
COASTAL LINE
KESAVADASAPURAM
AKULAM
ULLOOR CANAL
NANDANKOD
ANAYARA
KARAMANA RIVER
PARVATHIPUTHEN RIVER
STATUE
KARAMANA
SANKUMUKHAM
AIR PORT
CHALA
PAPPANAMKOD
MUTTATHARA
LAKSHADWEEP SEA

7. 2. QUERY 2

7. 2. A. Input Screen

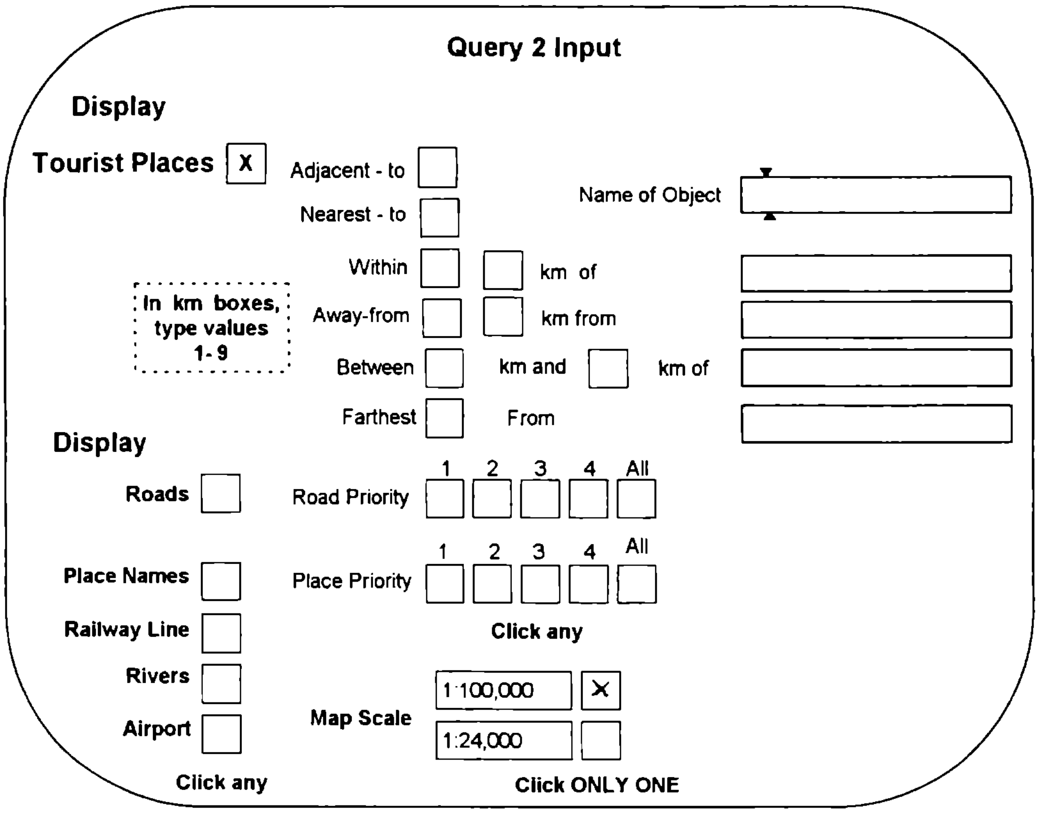


Figure 7. 2. Screen for Selecting the Optional Query Parameters for Query 2

7. 2. B. Plate 2

Query Objective: To display the **important tourist places** in Thiruvananthapuram City with their names and locations, and the **major roads that lead to them.**

Production Name: **find-tourist-places-routes**

Production:

```
p find-tourist-places-routes
(altpl          ^name               name)
        {<altloc>       (tourist-place   ^name   <name1>
                        ^location        <location>
                        ^x-cord          <x-cord1>
                        ^y-cord          <y-cord1>
                        ^rd-adj          <rdname>
                        ^tp-no           <tpno>)}
        (place          ^pname           <location>
                        ^x-cord          <x-cord2>
                        ^y-cord          <y-cord2>)
        (road           ^rd-name         <rdname>
                        ^rd-cord         <rd-cord>
                        ^st-xcord        <st-xcord>
                        ^st-ycord        <st-ycord>
                        ^rd-no           <rd-no>)
```

```
-->
        (call clgplotall   <name1>      <location>
                           <x-cord1>    <y-cord1>
                           <x-cord2>    <y-cord2>)
        (call rdplotall    <rdname>     <rd-cord>
                           <st-xcord>   <st-ycord>
                           <rd-no>))
```

Explanation:

In this query, find-tourist-places-routes is the identifier of the production that is to be used. The element

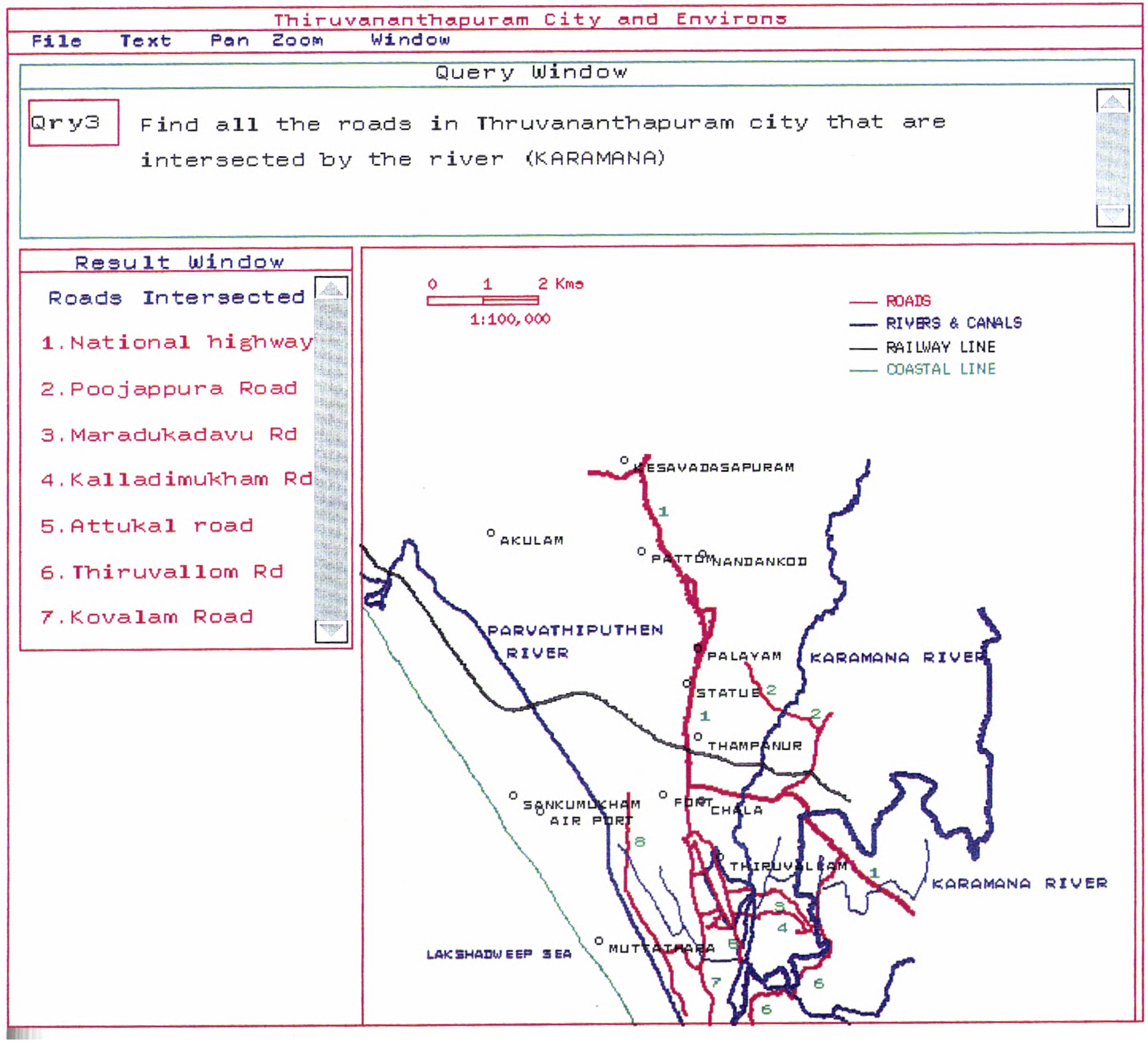
```
(altpl   ^name   name)
```

is the control-element or context-element used for activating the production. The symbol <altloc> is used as the element designator, which may be used to refer to that particular condition-element of the production at any stage in the program. The condition-element must be enclosed in braces when an element designator is associated with it. clgplotall and rdplotall are external procedure names.

The Disply menu with the option Tourist Places appears first. The user has to select this option. Now the important tourist places are displayed with the roads that lead to them. The user also has the option to display more roads, railway lines, rivers etc. and place names on the map by selecting the other two options. If the Roads option is selected, then the Road Priority menu appears. If the Place Names option is selected then Place Names Priority menu appears

In the resulting map, the tourist places are marked in green, with their names displayed in the result window The user can scroll through the display using the scroll buttons on the scroll bar

Response Time: 12 seconds



Thiruvananthapuram City and Environs
File Text Pan Zoom Window
Query Window
Qry3
Find all the roads in Thruvananthapuram city that are intersected by the river (KARAMANA)
Result Window
Roads Intersected
1.National highway
2.Poojappura Road
3.Maradukadavu Rd
4.Kalladimukham Rd
5.Attukal road
6.Thiruvallom Rd
7.Kovalam Road
0 1 2 Kms
1:100,000
ROADS
RIVERS & CANALS
RAILWAY LINE
COASTAL LINE
KESAVADASAPURAM
AKULAM
PATTOM
NANDANKOD
PARVATHIPUTHEN RIVER
PALAYAM
KARAMANA RIVER
STATUE
THAMPANUR
SANKUMUKHAM
AIR PORT
FORT
CHALA
THIRUVALLAM
KARAMANA RIVER
LAKSHADWEEP SEA
MUTTATHARA

7. 3. QUERY 3

7. 3. A. Input Screen

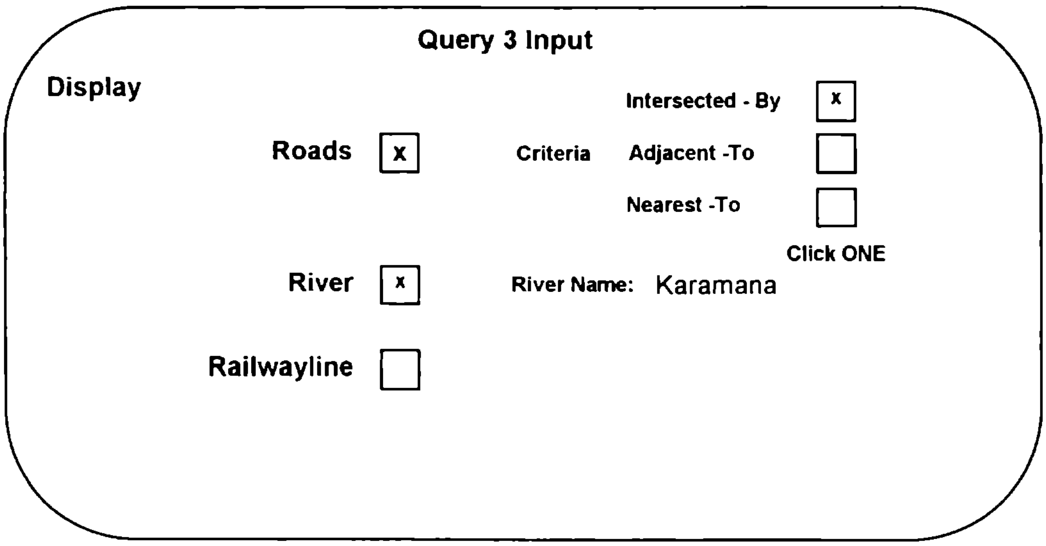


Figure 7. 2. Screen for Selecting Option Parameters for Query 3

7. 3. B. Plate 3

Query Objective: To find and display all the roads that cross a particular river

Query Option: **Karamana river**

Production Name **find-all-roads-cross-river**

Production:

```
(p find-all-roads-cross-river
(rdrcross         ^rname          name)
        (river    ^name           <namer>
                  ^rd-cross       yes
                  ^crplcrd        <crplc>
                  ^rr-cord        <rrcord>)
        (road     ^rd-name        <rdname>
                  ^rcross         yes
                  ^rname          <naner>
                  ^rd-cord        <rd-cord>
                  ^st-xcord       <st-xcord>
                  ^st-ycord       <st-ycord>
                  ^rd-no          <rd-no>)
        (place    ^pname          <crplc>
                  ^x-cord         <x-cord>
                  ^y-cord         <y-cord>)
-->
                  (call riverplot         <namer> <rrcord>)
                  (call roadplot          <rdname> <rd-cord>
                  (call plotpointp        <rd-no> <st-xcord> <st-ycord>))
```

Explanation:

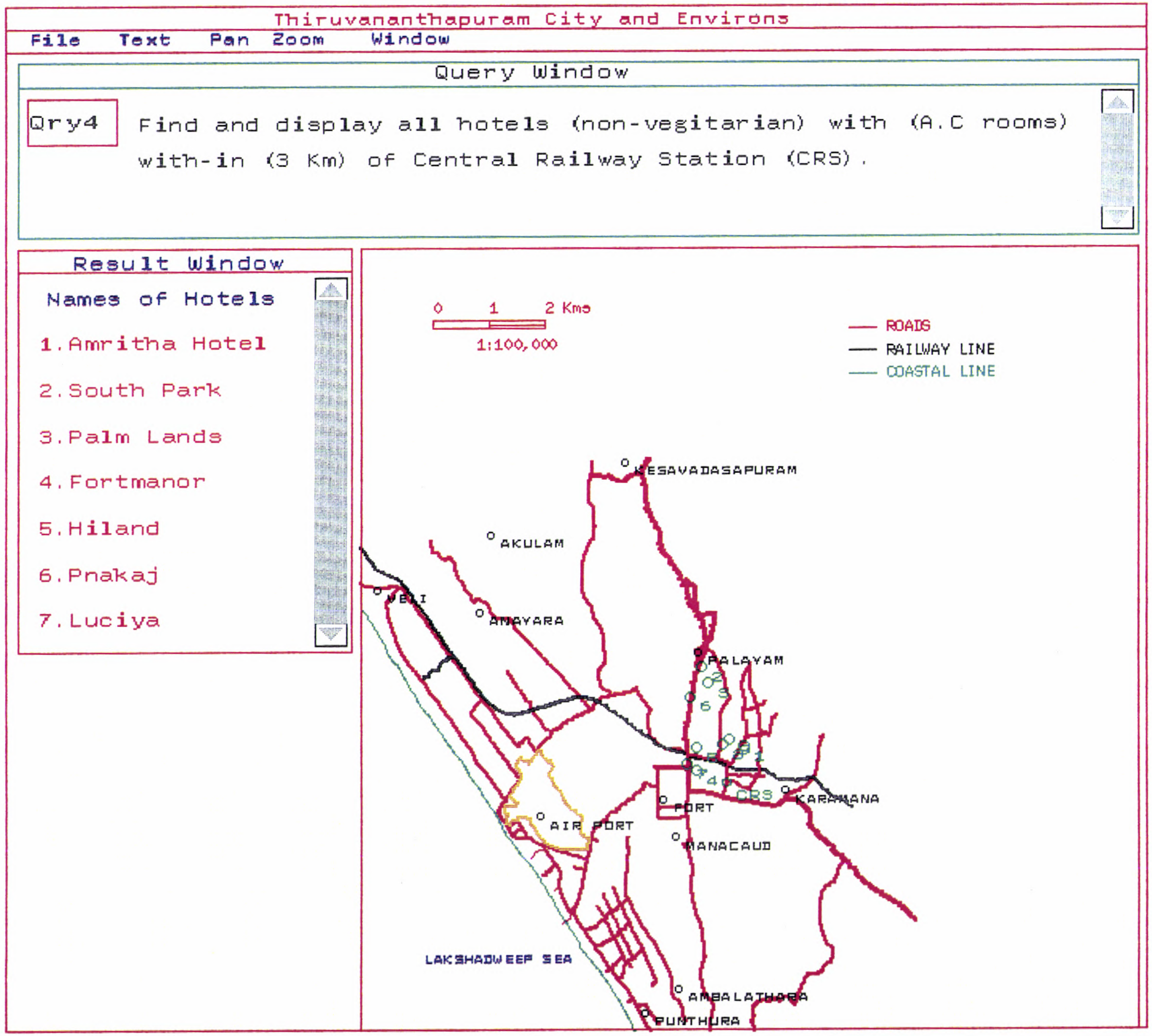
As in the explanation for the previous queries, the element

(rdrcross ^rname name)

is the control element used to activate the production.

In this query, the Display menu with the option Roads appears first. This is to be selected by the user. Now the the Criteria menu appears with three options. Here the user can select any one of the three options. By selecting the first option the user is prompted to choose any one of the two options River/Railway line. By choosing the River option, the user is prompted for entering the name of the river, Once the name of the river is entered, the resullting map is displayed with all the roads that are intersected by the river (KARAMANA) The names of the roads that are intersected by the river are displayed in the result window

Response Time: 9 seconds



Thiruvananthapuram City and Environs
File Text Pan Zoom Window
Query Window
Qry4
Find and display all hotels (non-vegitarian) with (A.C rooms) with-in (3 Km) of Central Railway Station (CRS).
Result Window
Names of Hotels
1.Amritha Hotel
2.South Park
3.Palm Lands
4.Fortmanor
5.Hiland
6.Pnakaj
7.Luciya
0 1 2 Kms
1:100,000
ROADS
RAILWAY LINE
COASTAL LINE
KESAVADASAPURAM
AKULAM
ANAYARA
PALAYAM
CRS
KARAMANA
FORT
AIR PORT
MANACAUD
LAKSHADWEEP SEA
AMBALATHARA
PUNTHURA

7. 4. PLATE 4

Query Objective: To find and display **all hotels** with **air-conditioned double rooms within 3 km** of **Central Railway Station** and where **non-vegetarian food** is available.

Query Option: **Hotels**
Air-conditioned
Double rooms
Non-vegetarian food
Within 3 km of
Central Railway Station

Production Name: **find-hotels-aircon-near-central-rlystn**

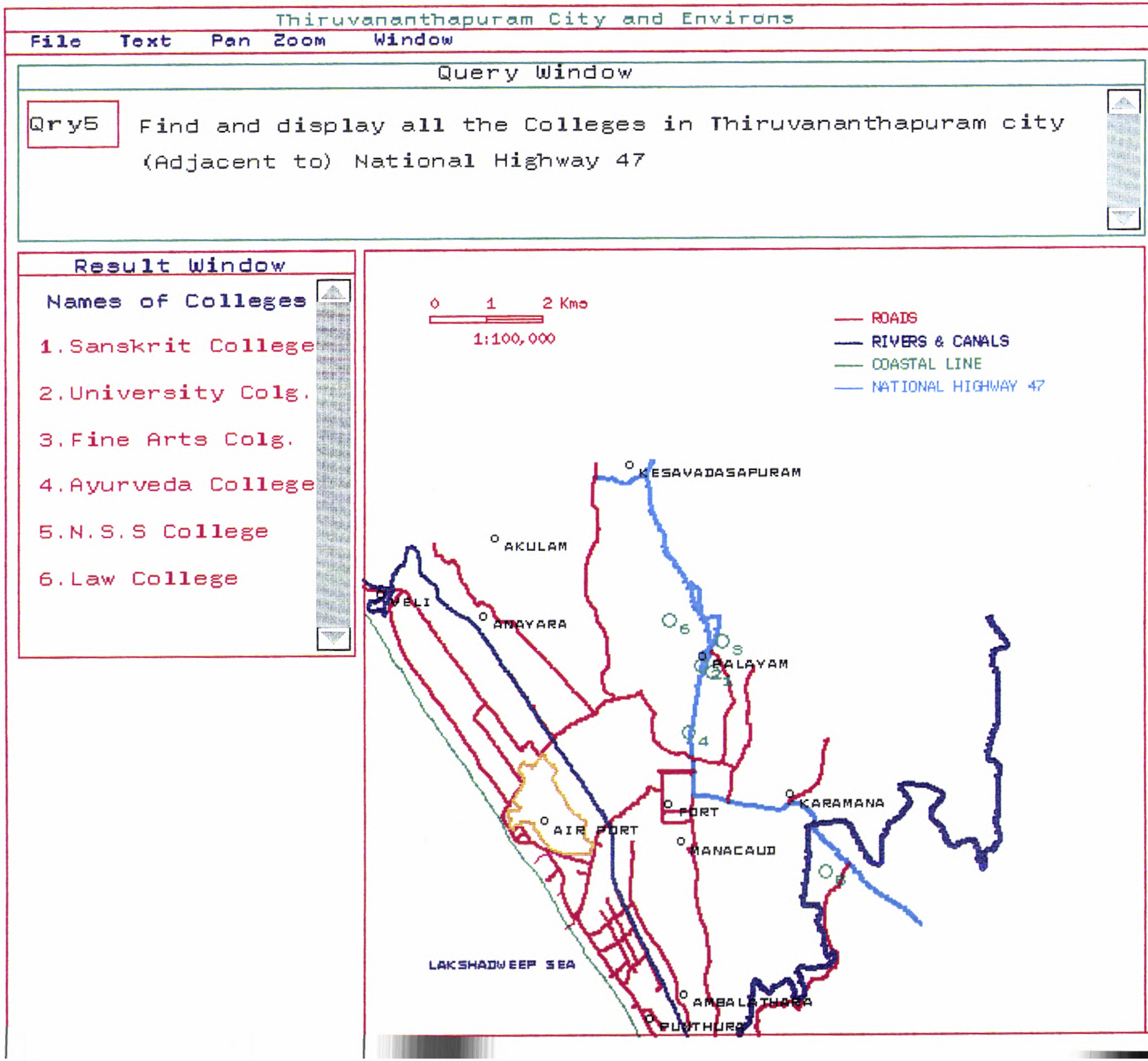
Production:

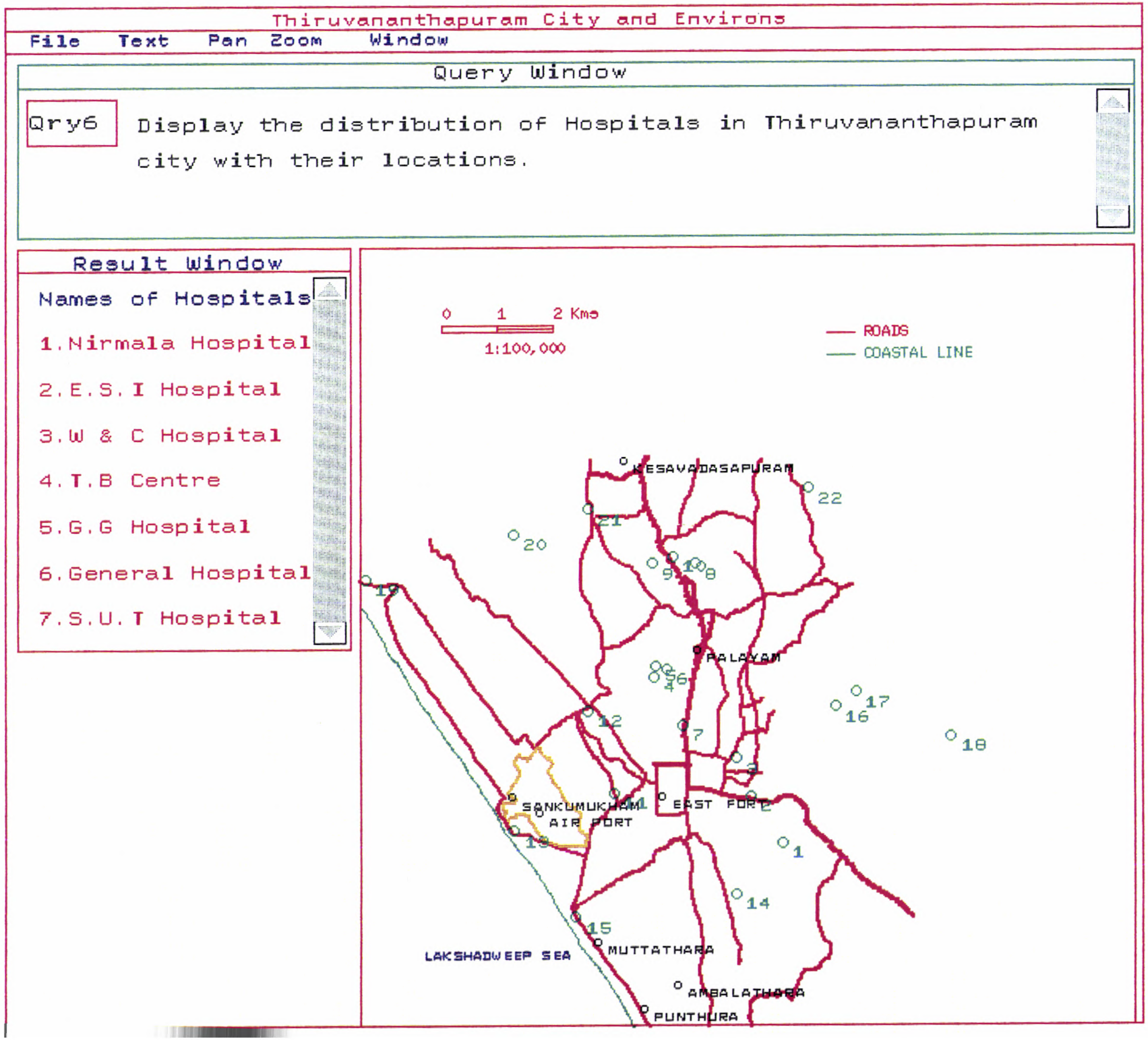
```
(p find-hotels-aircon-near-central-rlystn
(alhtlac        ^name          name1)
{<alhtlloc>     (hotel         ^name           <name>
                ^location      <location>
                ^x-cord1       <x-cord1>
                ^y-cord1       <y-cord1>
                ^htl-no        <htlno>
                ^m-type        non-veg
                ^r-type        ac-double
                ^criteria      (with-in <3>
                                     <central-railway-station>))}
-->
        (call clgplotall       <name>          <x-cord1>
                               <y-cord1>       <htlno>))
```

Explanation:

For this production, find-hotels-aircon-near-central-rlystn is the name of the production, and (alhtlac ^name name1) is the control element. The symbol <alhtlloc> is the element designator The condition-element is now enclosed in braces. The value for the attribute criteria is evaluated by the function within which takes two arguments, *viz.*, the distance in km and the object identifier

Response Time: 8 seconds


Thiruvananthapuram City and Environs
File Text Pan Zoom Window
Query Window
Qry5
Find and display all the Colleges in Thiruvananthapuram city
(Adjacent to) National Highway 47
Result Window
Names of Colleges
1. Sanskrit College
2. University Colg.
3. Fine Arts Colg.
4. Ayurveda College
5. N.S.S College
6. Law College
0 1 2 Kms
1:100,000
ROADS
RIVERS & CANALS
COASTAL LINE
NATIONAL HIGHWAY 47
KESAVADASAPURAM
AKULAM
ANAYARA
PALAYAM
KARAMANA
FORT
AIR PORT
MANACAUD
LAKSHADWEEP SEA
AMBALATHARA
PUTHURA


Thiruvananthapuram City and Environs
File Text Pan Zoom Window
Query Window
Qry6
Display the distribution of Hospitals in Thiruvananthapuram city with their locations.
Result Window
Names of Hospitals
1. Nirmala Hospital
2. E.S.I Hospital
3. W & C Hospital
4. T.B Centre
5. G.G Hospital
6. General Hospital
7. S.U.T Hospital
0 1 2 Kms
1:100,000
ROADS
COASTAL LINE
KESAVADASAPURAM
PALAYAM
EAST FORT
SANKUMUKHAM
AIR PORT
MUTTATHARA
AMBALATHARA
PUNTHURA
LAKSHADWEEP SEA

7. 5. PLATE 5

Query Objective: To find and display **all the colleges** in Thiruvananthapuram City that are **adjacent to National-Highway 47.**

Query Option: **Colleges**
Adjacent (≤ 0.5 km)
National-Highway-47

Production Name: **find-all-colleges-NH47**

Production: See Appendix D.5

Explanation:

In the production, detailed in Appendix D 1 (alclg ^name name1) is the control element, while <edn> and <clgs> are element designators. NH 47 is displayed in cyan colour, to be distinct from other streets. In the knowledge base, the expression adjacent to means a distance less than or equal to 0.5 km. The output colleges are marked in green colour

Response Time: 10 seconds

7. 6. PLATE 6

Query Objective: To find and display the **distribution of hospitals** in Thiruvananthapuram city.

Query Option: **hospitals**

Production Name: **find-all-hospital**

Production: See Appendix D.2

Response Time: 8 seconds

Thiruvananthapuram City and Environs

File Text Pan Zoom Window

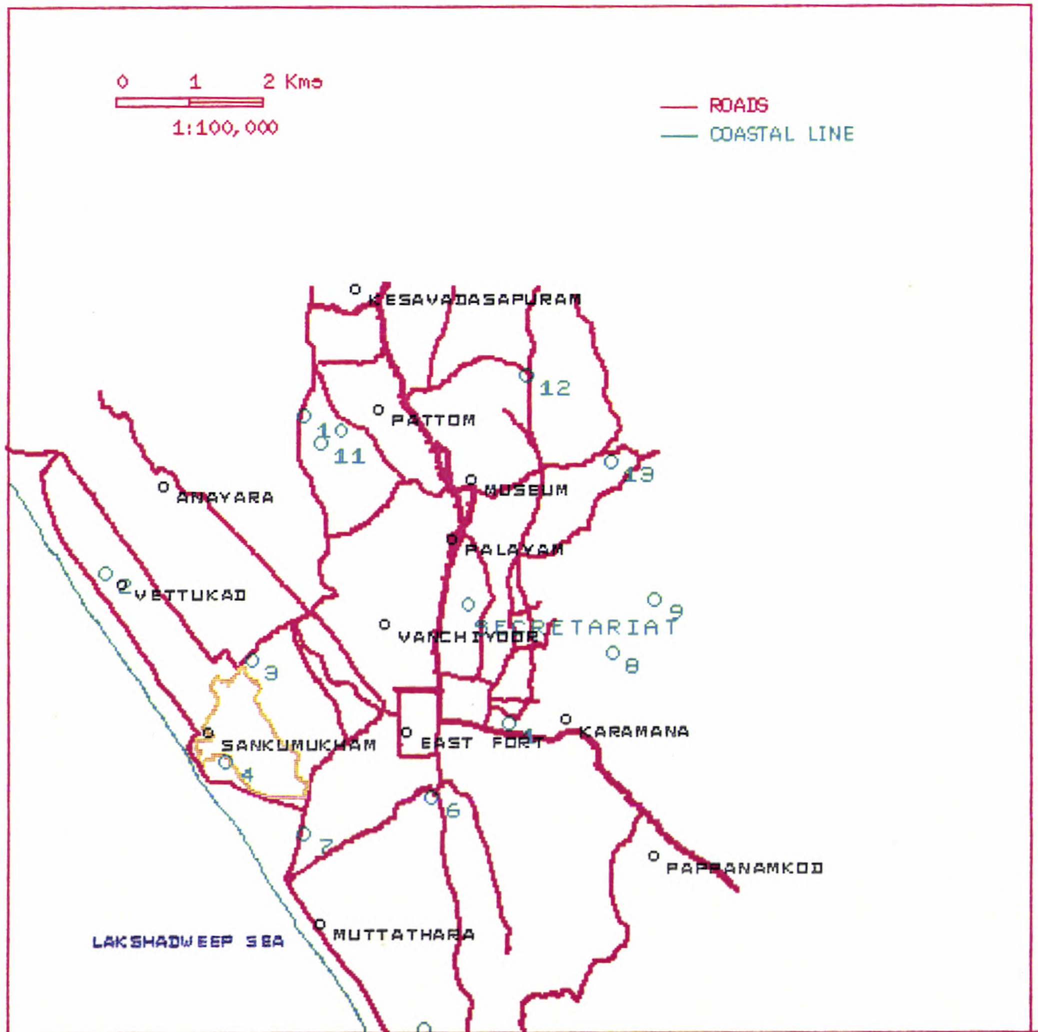
Query Window

Qry7 Find and display all schools located between (2 Km) and (6 Km) of the SECRETARIAT building.

Result Window

Names of Schools

1. Gvt. H. S Aryasala
2. St. Mary's H. S
3. N. S. S E. M. H. S
4. St. Rocha's H. S
5. Gvt. H. S Punthura
6. Gvt. H. S Manacaud
7. Med. College H. S



7. 7. PLATE 7

Query Objective: To find and display all **schools between 2 km** and **6 km of the Secretariat** building.

Query Options: **schools** located
between 2 km and **6** km
of Secretariat building

Production Name: **find-schools-around-secretariat**

Production: See Appendix D.3

Response Time: 16 seconds

7. 8. QUERY 8

7. 8. A. Input Screen

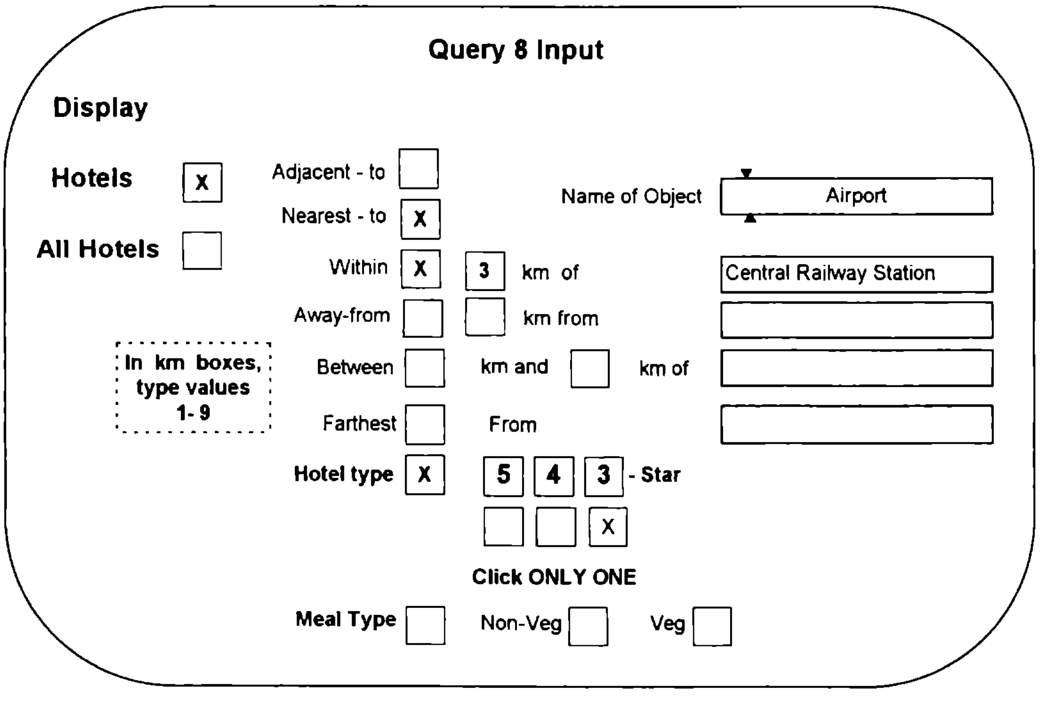
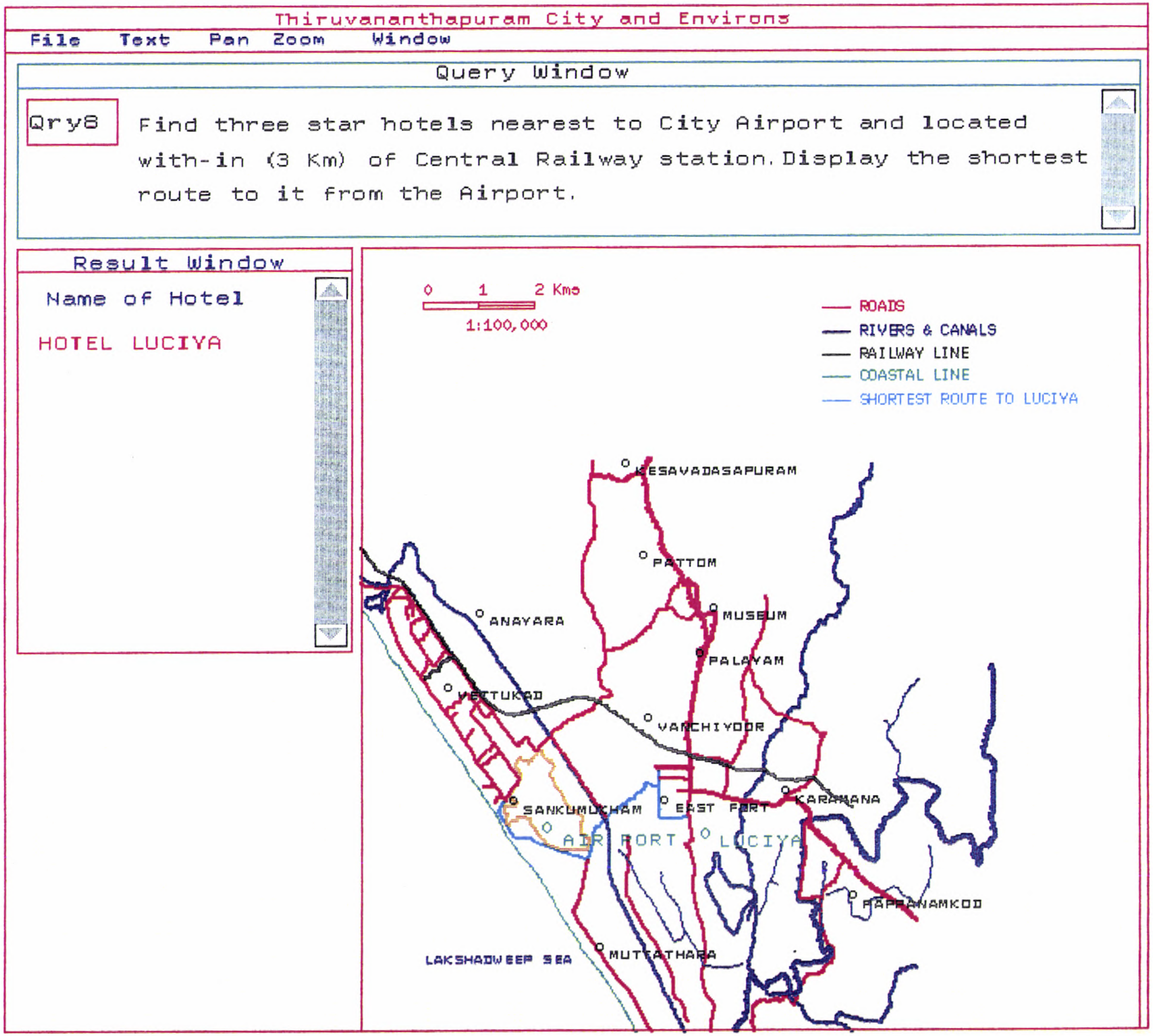


Figure 7. 8. Screen for Selecting the Optional Query Parameters for Query 8


Thiruvananthapuram City and Environs
File Text Pan Zoom Window
Query Window
Qry8
Find three star hotels nearest to City Airport and located with-in (3 Km) of Central Railway station. Display the shortest route to it from the Airport.
Result Window
Name of Hotel
HOTEL LUCIYA
0 1 2 Kms
1:100,000
ROADS
RIVERS & CANALS
RAILWAY LINE
COASTAL LINE
SHORTEST ROUTE TO LUCIYA
KESAVADASAPURAM
PATTOM
MUSEUM
ANAYARA
PALAYAM
VETTUKAD
VANCHIYOOR
SANKUMUKHAM
EAST FORT
KARAMANA
AIR PORT
LUCIYA
PAPPANAMKOD
LAKSHADWEEP SEA
MUTTATHARA

7. 8. B. PLATE 8

Query Objective: To display the **3-star hotel nearest to City Airport and located** within **3 km of Central Railway Station.** Display the **shortest route to it from the Airport.**

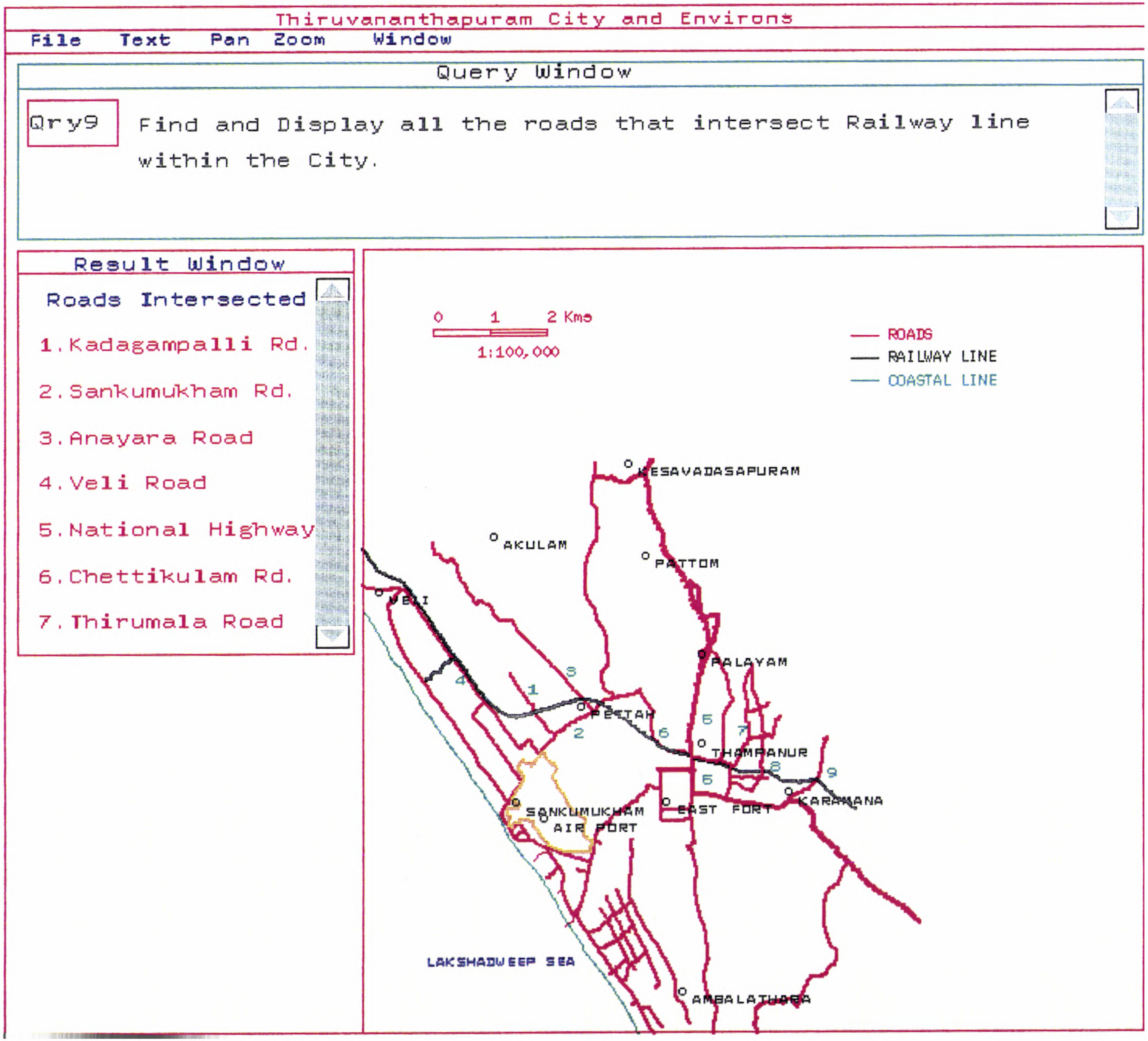
Query Options: **3-star hotel**
nearest to **City Airport**
within 3 km
of Central Railway Station

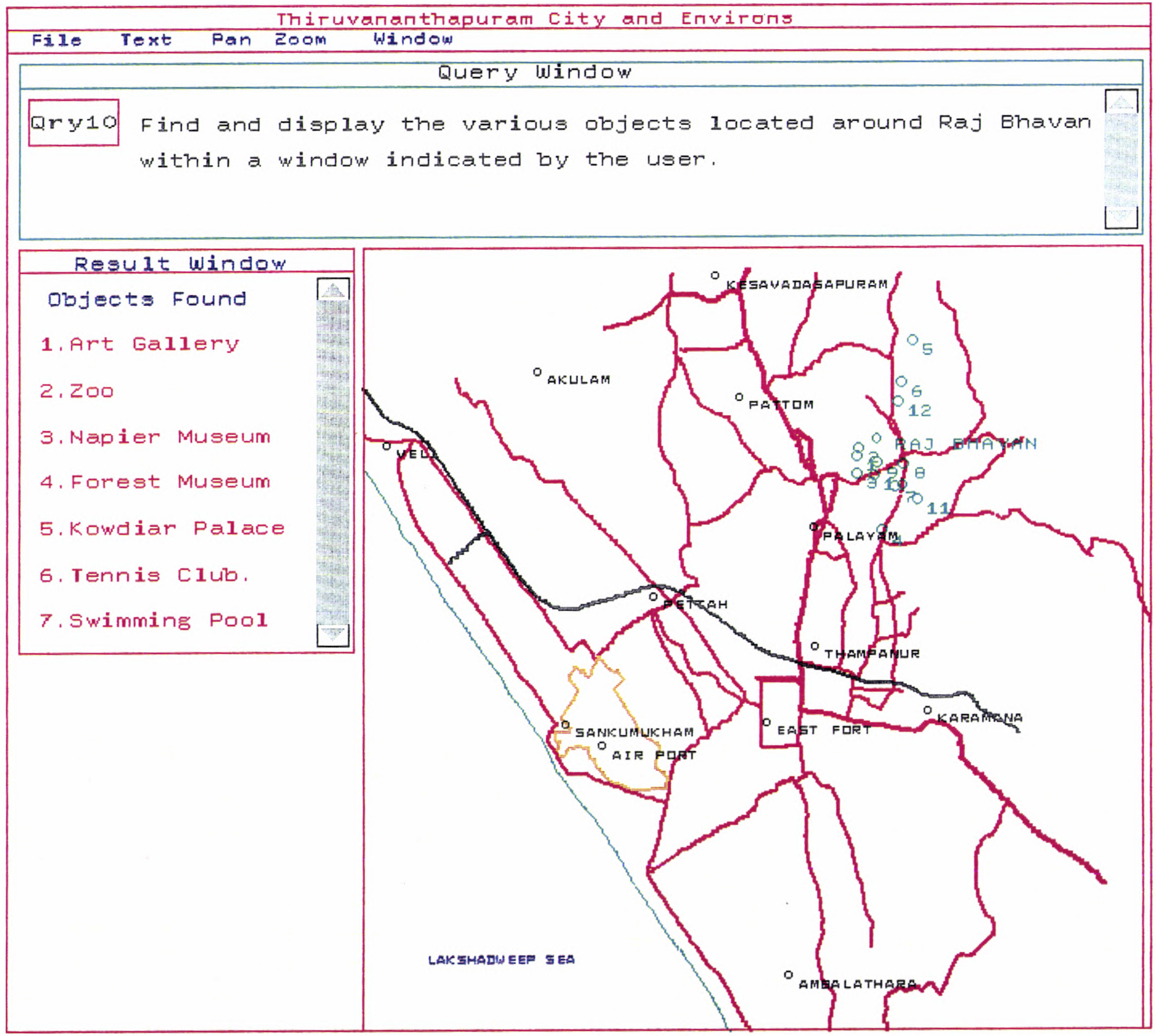
Production Name: **find-hotel- nearest-airport-and-route**

Production:

```
(p find-hotel-nearest-airport-and-route
(htl      ^name   <name1>)
          (public-utility   ^name          airport
                            ^x-cord        <x-cord1>
                            ^y-cord        <y-cord1>)
          {<hloc> (hotel    ^name          <htlname>
                            ^location      <location>
                            ^x-cord        <x-cord2>
                            ^y-cord        <y-cord2>
                            ^criteria  ({closest<x-cord1>          <y-cord1>
                                           within  <3> <central-railway-station>}))}
          (place            ^pname         <location>
                            ^x-cord        <x-cord3>
                            ^y-cord        <y-cord3>)
- ->
          (call htlplot                  <name>          <location>
                                         <x-cord2>       <y-cord2>)
          (call plotpoint                <x-cord1>       <y-cord1>)
          (call  plot-shortest-route     <x-cord1>       <y-cord1>
                                         <x-cord2>       <y-cord2>))
```

Response Time: 18 seconds


Thiruvananthapuram City and Environs
File Text Pan Zoom Window
Query Window
Qry9
Find and Display all the roads that intersect Railway line within the City.
Result Window
Roads Intersected
1. Kadagampalli Rd.
2. Sankumukham Rd.
3. Anayara Road
4. Veli Road
5. National Highway
6. Chettikulam Rd.
7. Thirumala Road
0 1 2 Kms
1:100,000
ROADS
RAILWAY LINE
COASTAL LINE
KESAVADASAPURAM
AKULAM
PATTOM
VELI
PALAYAM
PETTAH
THAMPANUR
EAST FORT
KARAMANA
SANKUMUKHAM
AIR PORT
LAKSHADWEEP SEA
AMBALATHARA


Thiruvananthapuram City and Environs
File Text Pan Zoom Window
Query Window
Qry10 Find and display the various objects located around Raj Bhavan within a window indicated by the user.
Result Window
Objects Found
1. Art Gallery
2. Zoo
3. Napier Museum
4. Forest Museum
5. Kowdiar Palace
6. Tennis Club.
7. Swimming Pool
KESAVADASAPURAM
AKULAM
PATTOM
RAJ BHAVAN
VELI
PALAYAM
PETTAH
THAMPANUR
KARAMANA
SANKUMUKHAM
AIR PORT
EAST FORT
LAKSHADWEEP SEA
AMBALATHARA

7. 9. PLATE 9

Query Objective: Display all the **roads** that **Intersect Railway line** within the City.

Query Options: **Roads**
Intersect railway line

Production Name: **find-all-roads-cross-rail.**

Production:

```
(P find-all-roads-cross-rail
        (rail    ^rl-cord  <rl-cord> )
        (road    ^rd-name        <rdname>
                 ^rl-cross       yes
                 ^crplcrl        <crplc>
                 ^rd-cord        <rd-cord>
                 ^st-xcord       <st-xcord>
                 ^st-ycord       <st-ycord>
                 ^rd-no          <rd-no>)
        (place   ^pname          <crplc>
                 ^x-cord          <x-cord>
                 ^y-cord          <y-cord>)
- ->
        (call  railplot    <rl-cord>)
        (call  roadplot    <rdname>       <rd-cord>)
        (call  plotpoint   <rd-no>        <st-xcord>       <st-ycord>))
```

Response Time: 12 seconds

7. 10. PLATE 10

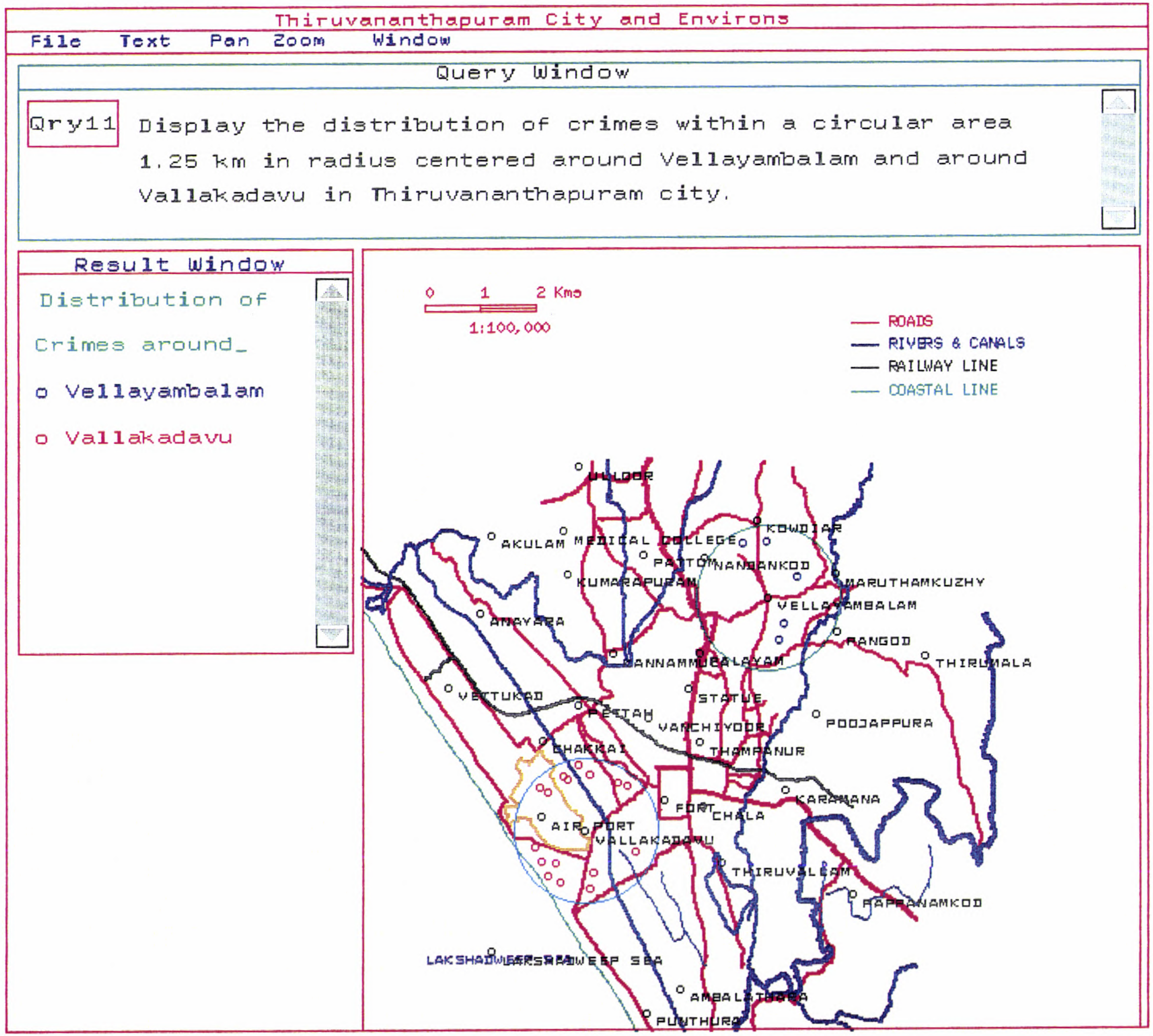
Query Objective: Display all the objects located around 'Raj Bhavan' (the State Governor's House) within the window indicated by the user.

Query Options: **All objects** located around '**Raj Bhavan**'
within the **window selected** at run-time

Production Name: **find-all-objects-in window**

Production: Not shown

Response time: 24 seconds; depends on the size and orientation of the window chosen


Thiruvananthapuram City and Environs
File Text Pan Zoom Window
Query Window
Qry11 Display the distribution of crimes within a circular area 1.25 km in radius centered around Vellayambalam and around Vallakadavu in Thiruvananthapuram city.
Result Window
Distribution of Crimes around_
o Vellayambalam
o Vallakadavu
0 1 2 Kms
1:100,000
ROADS
RIVERS & CANALS
RAILWAY LINE
COASTAL LINE
ULLOOR
AKULAM
MEDICAL COLLEGE
KOWDIAR
PATTOM
NANDANKOD
KUMARAPURAM
MARUTHAMKUZHY
VELLAYAMBALAM
ANAYARA
RANGOD
THIRUMALA
KANNAMMOOLA
PALAYAM
VETTUKAD
STATUE
PETTAH
VANCHIYOOR
POOJAPPURA
CHAKKAI
THAMPANUR
KARAMANA
FORT
CHALA
AIRPORT
VALLAKADAVU
THIRUVALLAM
PAPPANAMKOD
AMBALATHARA
PUNTHURA

7. 11. PLATE 11

Query Objective: To display the distribution of crime within a circular area of diameter 2. 5 km each centred around *Vellayambalam* and *Vallakadavu*

Query Options: **Crime** Incidence
Within a circle of radius **1.25 km Centred** around
Vellayambalam and also
Vallakadavu

Production Name: **find-crimes**

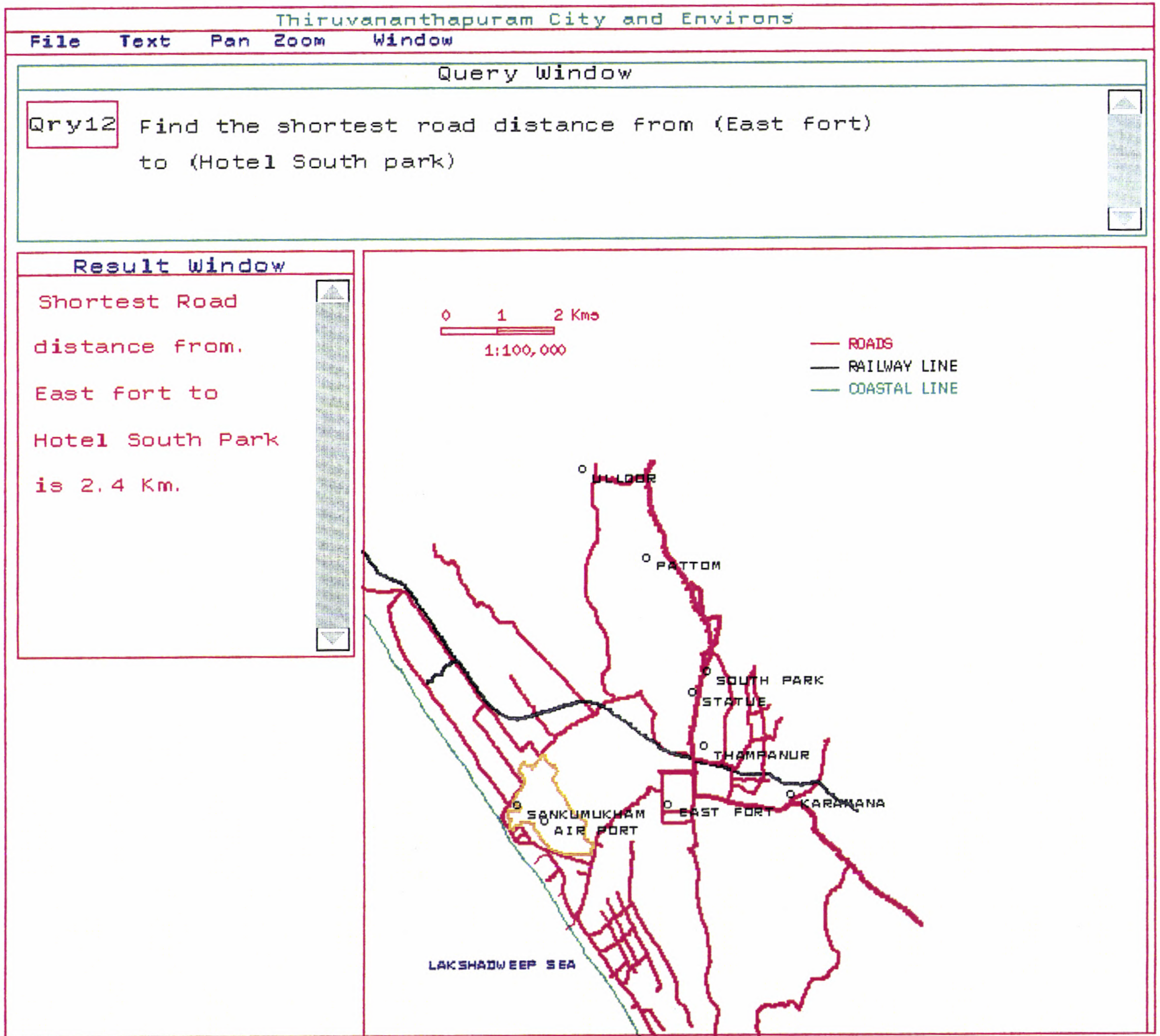
Production:

```
(p find-crimes
        (place   ^pname          <pname1>
                 ^x-cord         <x-cord1>
                 ^y-cord         <y-cord1> )
        (crime   ^location       <pname1>
                 ^x-cord  {> (compute x-cord1 - 1.25)
                                 < (compute x-cord1 + 1.25)}
                 ^y-cord  { > (compute y-cord1 - 1.25)
                                 < (compute y-cord1 + 1.25)})
- -->
        ( call clgplotall <pname1> <x-cord> <y-cord>))
```

Response Time: 11 seconds

Explanation:

The data pertaining to incidence of crime in the two localities, *Vellayambalam* and *Vallakadavu*, are **not** based on the police records; attempts made to obtain such data were not successful. The display shown are based on hypothetical data for two of the crime-prone areas of the city


Thiruvananthapuram City and Environs
File Text Pan Zoom Window
Query Window
Qry12
Find the shortest road distance from (East fort)
to (Hotel South park)
Result Window
Shortest Road
distance from.
East fort to
Hotel South Park
is 2.4 Km.
0 1 2 Kms
1:100,000
ROADS
RAILWAY LINE
COASTAL LINE
PATTOM
SOUTH PARK
STATUE
THAMPANUR
KARAMANA
EAST FORT
SANKUMUKHAM
AIR PORT
LAKSHADWEEP SEA

7. 12. PLATE 12

Query Objective: Find the shortest road distance from East Fort to Hotel South Park

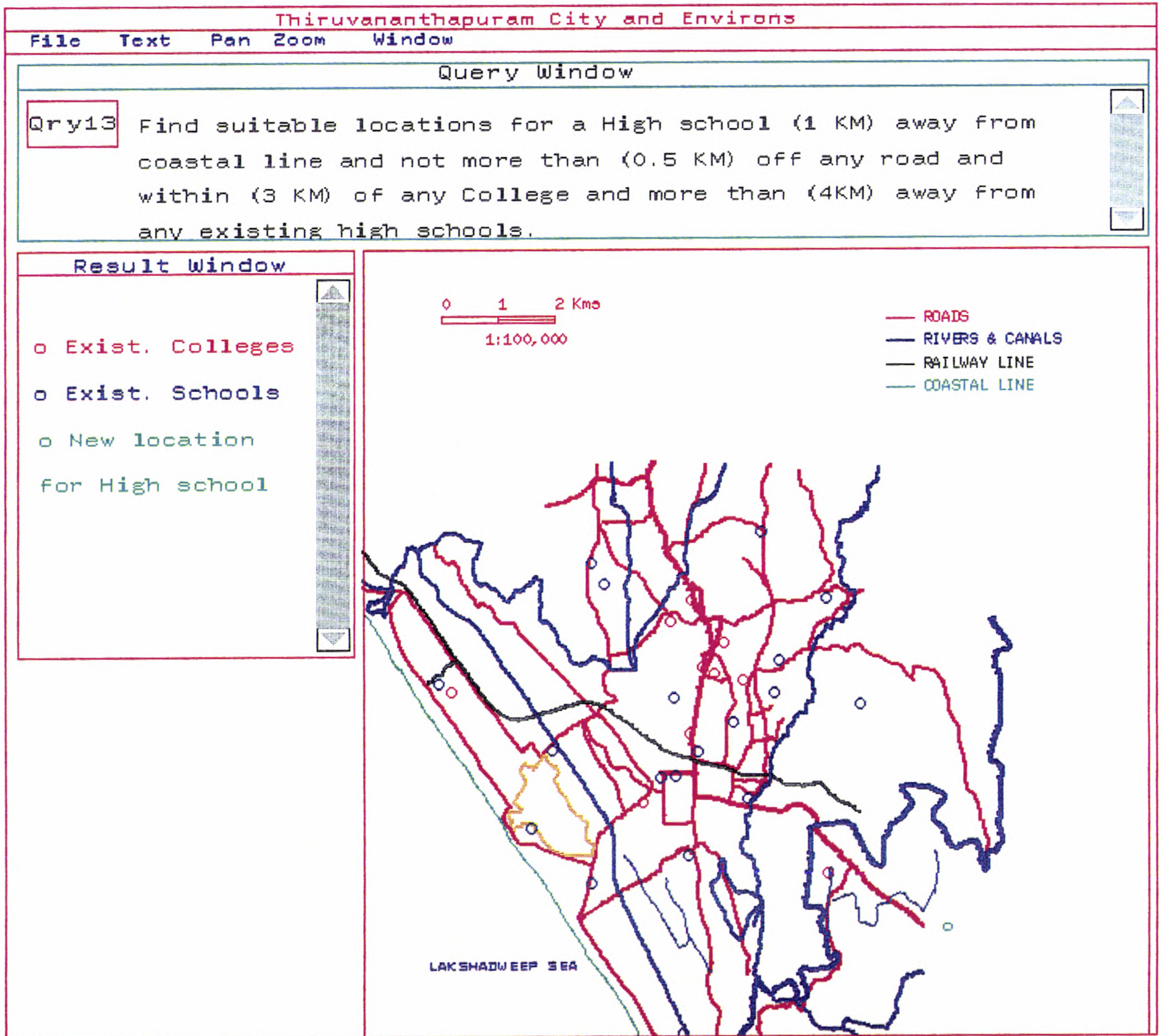
Query Options: **East Fort**
Hotel South Park

Production Name: **find-dist**

Production

```
(p find-dist
(f-dist   ^name    name)
          (place   ^pname          <pname1>
                   ^x-cord         <x-cord1>
                   ^y-cord         <y-cord1>)
          (hotel   ^name           <name1>
                   ^location       <loc1>
                   ^x-cord         <x-cord2>
                   ^y-cord         <y-cord2>)
          (road    ^rd-cord        <rd-cord>
                   ^rd-name        <rd-name>
                   ^criteria       {adjacent <pname1>      adjacent <loc1>})
- ->
          (call find-shortest-dist <x-cord1>       <y-cord1>
                                   <x-cord2>       <y-cord2 >
                                   <rd-name>       <rd-cord>)
          (call roadplot           <rd-name>       <rd-cord>)
          (call plotpoint          <x-cord2 >      <y-cord2 >)
          (call plotpoint          <x-cord1>       <y-cord1> ))
```

Response Time: 11 seconds


Thiruvananthapuram City and Environs
File Text Pan Zoom Window
Query Window
Qry13
Find suitable locations for a High school (1 KM) away from coastal line and not more than (0.5 KM) off any road and within (3 KM) of any College and more than (4KM) away from any existing high schools.
Result Window
o Exist. Colleges
o Exist. Schools
o New location for High school
0 1 2 Kms
1:100,000
ROADS
RIVERS & CANALS
RAILWAY LINE
COASTAL LINE
LAKSHADWEEP SEA

7. 13. PLATE 13

Query Objective: Find suitable locations for a High School 1 km away from coastal line and not more than 0.5 km off any road and within 3 km of any college and more than 4 km away from existing schools.

Query Options: **High school**
1 km away from coastal line
≤ 0.5 km from a road
≤ 3 km of a college
> 4 km away from existing schools

Production name: **find-locations-hs**

Production:

```
(p find-locations-hs
        (fschool            ^name          <name1>)
        (edn-research       ^type          high-school
                            ^name          <name1>
                            ^location      <location1>
                            ^x-cord        <x-cord1>
                            ^y-cord        <y-cord1>)
        (edn-research       ^type          college
                            ^name          <name2>
                            ^location      <location2>
                            ^x-cord        <x-cord2>
                            ^y-cord        <y-cord2>)
        (boundary           ^bound-cord    <bound-cord1>)
        (road               ^rd-name       <rdname>
                            ^rd-cord       <rd-cord>)
        (make edn-research  ^type          high-school
                            ^x-cord        <x-cord3>
                            ^y-cord        <y-cord3>
                            ^criteria  {away-from <bound-cord1> <1>
                                       <x-cord3> <y-cord3>
                                       with-in <3> <name2>
                                       away-from <name1> <4>
                                       <x-cord3> <y-cord3>
                                       with-in <0.5> <rd_cord> })
-->
        (call plotpoint     <x-cord3>      <y-cord3>))
```

Response Time: 26 seconds

7. 14. SUMMARY

THE OUTPUT some typical queries, in the form of screen print-outs are presented in this chapter. In certain cases, the input screen for the query is shown. A narration of the query, the options exercised by the user, the name of the production, and the code for the production, are followed by an explanation of the production and the related terms. In some other cases, the Input Screens are not shown.

Chapter 8

Conclusions and Future Directions

A GIS has been designed with limited functionalities; but with a novel approach in its design. The spatial data model adopted in the design of KBGIS is the unlinked vector model. Each map entity is encoded separately in vector form, without referencing any of its neighbouring entities. Spatial relations, in other words, are not encoded. This approach is adequate for routine analysis of geographic data represented on a planar map, and their display (Pages 105-106). Even though spatial relations are not encoded explicitly, they can be extracted through the specially designed queries.

This work was undertaken as an experiment to study the feasibility of developing a GIS using a knowledge base in place of a relational database. The source of input spatial data was accurate sheet maps that were manually digitised. Each identifiable geographic primitive was represented as a distinct object, with its spatial properties and attributes defined. Composite spatial objects, made up of primitive objects, were formulated, based on production rules defining such compositions. The facts and rules were then organised into a production system, using OPS5.

Presently, KBGIS supports three classes of objects:

Point, to represent places, hotels, institutions, buildings, *et cetra*;
Network chain, to represent roads, rivers & canals, railway lines, *et cetra*;

Node, to represent road intersections, railway crossings, traffic intersections, *et cetra*.

Each of these classes differ in their attributes, based on their geographic rôles (Pages 129-131). An important attribute of a point object is *criteria*, pertaining to the locational details of the point object relative to other objects, point as well as chains. These attributes are convenient in processing queries involving locations of objects.

Among the subsystems that support KBGIS (Data Handling, Editing, Query, Library, and Functions), the Query subsystem has a novel design, and plays a crucial rôle in the analysis of spatial data. In addition to the *ad hoc* menu-based input of queries, the *standard queries* module of the Query subsystem provides a set of pre-stored query statements pertaining to the domain city, with facility for inputting optional parameters (Page 125). This simplifies the tasks of the user while navigating the KBGIS or in getting the required information, through a "filling-the-form" mode of input. Inputs for the Query 2 (Page 148) and Query 8 (Page 154) are illustrative of this facility

KBGIS is intended to support the spatial information needs of those who constantly interact with a given city and its services. Typical of these users are

- Tourists, seeking information on hotels, restaurants, tourist centres, government offices, post office, banks, *et cetra*.
- Officials who are involved regularly in allocating resources to schools, colleges, or research institutions that come under their superintendence.
- Traffic department and the Police, who would like to monitor traffic accidents or incidence of crime, analyse them locality-wise, and take preventive measures appropriate to each locality

- Officials at the City Corporation/Town Planning department, who could browse KBGIS for the objects of interest, assess their relative locations or determine the distance between one object and the rest of others, while making strategic planning.

8. 1. MAJOR CONTRIBUTIONS OF THE WORK

THE MAIN contributions of the research work reported in this thesis can be described as follows:

1 Identification of the major issues and problems that surround the designing of GISs using relational, object-oriented, and hybrid, databases. These have been elaborated in Sections 2.3 and 2.4 (Pages 58-74).

2. Proposal for finding alternate schemes for designing a robust and user-friendly GIS. Reports from the casual or irregular users of GISs, point towards the fact that several commercially available GIS packages are hard to use [Traynor & Williams 1997].

3 Identification of OPS5 as a suitable tool for the design of a GIS. Section 3.2.7 (Page 104-106) have provided the rationale for the suitability of OPS5 in the design of GIS. The use of a sophisticated indexing scheme, adopted in the Rete algorithm for finding rules that match the current database makes OPS5 one of the tools that executes the fastest, as per the benchmark studies conducted on several expert system tools.

4. It has been demonstrated that the relationships between geographic entities can be represented in a natural way by treating each spatial information as an object; and an object can belong to one of the many classes. Classes can be created at different levels of spatial resolutions. A spatial tree structure can be generated through class-subclass relationship formalism.

5. It has been demonstrated that a spatial object knowledge-base of a reasonable size can be created easily with the built-in facilities of OPS5. This is borne out by the results and their discussions, covered in Chapter 7

6. The data model adopted in the design of KBGIS is primarily a vector representation strategy, because the geographic data entities are stored as a vector of coordinate points that define the geometry and topology of these entities precisely. Secondly, it is a logic-based structure because two geographic entities, polygon and line are represented as predicate structures describing certain relationship between a series of objects that are attributes of space. Thirdly, it is a relational structure because the polygon and line predicate structures can be interpreted as implementation of two relations.

7 In contrast to a traditional DBMS-based GIS, KBGIS allows for an incremental construction process. Every object is encoded, independent of all the rest; but once a new object is added to the corpus of facts, it takes part in the query process, along with the previous ones. No new links have to be established when a piece of data is added.

8. KBGIS has met most of the specifications (*vide* Section 4.2.1, Page 114), it was required to satisfy, with the exception of those enumerated under 8.2, below

8. 2. LIMITATIONS OF THE PRESENT WORK

1 The data encoded in this prototype, relates only to one city.

2. The roads have been treated in a uniform manner, not distinguishing them as one-way/two-way, single-lane/multi-lane, with-median/without-median, bituminised/gravel, *et cetra*. Nor have the widths of the roads from segment-to-segment been registered in the fact base.

3. The Parcel Nos. for land parcels or Plot Nos. for plots of land, assigned by the Land Registry, Building Nos. assigned to buildings and houses by the Corporation, have not been implemented. This makes KBGIS unable to locate a building or a house or a plot, on the basis of appropriate identifying numbers.

4. It has not been possible to verify the relative performance of KBGIS with respect to a commercially available GIS, due to the non-availability of the latter. There are only very few GIS workstations in India, and access to them are jealously restricted.

8. 3. DIFFICULTIES WHILE PURSUING THIS RESEARCH

1 The topic of this research itself was daunting. There are only few Geography departments in this country where research is conducted; and none of such research is related to spatial information or cartography. As a result, the author of this thesis had to spend a substantial amount of time familiarising the geographical concepts and terminology

2. The data capturing process involved several months of manually digitising, verifying, and encoding. Automatic digitisers are prohibitively priced.

3. The author of this thesis had to work under severe limitations imposed by the scarcity of books and journals dealing with the concerned topic, in this university as well as in others. It was only towards the beginning of 1998 that this university was connected into the Internet; but the connections were very poor, resulting in frequent loss of carrier

4. The hardware used in this work were a prestigious lot at the time it was installed; but became outdated by the time this work got underway Figures like "16 MB RAM", "159 MB harddisk", "180 dpi colour graphics printer" sounded primitive even before the work began.

8. 4. FUTURE DIRECTIONS

FUTURE research directions may be divided into two channels *viz.*, continuation of the present work, so as to take KBGIS from the status of a prototype into a working system; and seeking applications of GIS which will have far-reaching impact on the society's development.

8. 4. 1. Further Work on KBGIS

The programme for enhancing the utility of KBGIS will have the following three tasks:

A. Expanding the domain gradually so as to bring more cities and districts under its cover;
B. Prospective use of data acquired during the new land survey using GPSs.
C. Classify the roads into sub-classes, as indicated in Item #2 under 8.2.
D. Implement the labelling of buildings and houses, as indicated in Item #3 under 8.2.

8. 4. 1. A. Expanding the Domain

This will augment KBGIS with data relating to more districts, starting with the headquarters of each district. Those districts for which accurate maps are readily available, will be the first ones to be encoded.

8. 4. 1. B. Using Data Acquired with GPS

The Government of Kerala has launched a project for surveying the entire state, using Geographic Positioning Systems (GPS) (*vide* Page 21). Efforts will be made in persuading the Government to release the data so that it can be used with KBGIS. With such accurate data, it should be possible to label each square metre of the land individually. Integrating such data with the older data will certainly be a challenging task by itself.

The tasks shown as Items C and D are interrelated with that of B.

8. 4. 2. Strategic Applications of GIS

Today, after spending many years on this research topic, the author is concerned much more about the design and use of systems that address broad-ranging and strategic applications of GIS, than the technical advances of GIS. It is true that GIS has been a major success story, academically, technologically, and in business practices. However, the advances in GIS have so far contributed very little to understanding the problems concerned with the use of systems that address strategic planning issues. The concern with the capabilities and application of proprietary GIS software has overshadowed the more crucial, albeit non-technical, set of issues and applications. The latter issues ought to be addressed, if there is to be a continued development path in this area.

GIS applications are virtually non-existent in India; and wherever they exist, they are marginal technical adjunct to management. New applications have to be identified which are low-cost, and not requiring complex spatial analysis. Two such applications are briefly described below.

8. 4. 2. A. Education Management System

Of all the states in India, Kerala spends the largest percentage of the annual state budget on education. This has created a large government department for overseeing the appointment of teachers, allocating them to various government schools, and transferring them periodically from one government school to another, often the wrong ones for the wrong reasons. The purchaser-provider paradigm of marketing practice aptly applies to the education system, with the student and the teacher in the respective rôles of the purchaser and the provider, and the education department in the rôle of a facilitator.

The functions of the facilitator is to

Understand the current and future needs of the schools in a locality, in terms of teacher-skills and other factors.
Understand where teacher-skills are currently existing but under-utilised.
Evaluating the consequences of exchanging teachers from one school to another on the basis of skills, in terms of efficiency changes, as well as effectiveness.

8. 4. 2. B. Health Management System

The principles of Education Management System can be applied, with suitable changes, to the government hospitals that are invariably overcrowded due to higher cost of treatment in the private hospitals. The functions of the facilitator (Health Department) would be to provide appropriate specialists in a locality where certain diseases are prevalent.

Both these applications can certainly be managed through a database. But the addition of a visualising system with localities marked out, will enhance the cognitive process. The kind of accuracy expected from a proprietary GIS could be relaxed here, unless it is required to calculate the distance between the two points where a teacher (or a specialist doctor) is to be relieved and the point where she is to join, to fractions of a km, while calculating the cost of that transfer. Under-utilisation of teacher-skills (specialist doctor) at one place and unavailability of the very same skills (or specialist doctor) at another can be reduced considerably

8. 5. SUMMARY

THIS concluding chapter has given a wrap-up of the thesis, restating the objectives of the thesis, highlighting the achievements of the research, and pointing out the objectives that could not be realised. A brief agenda for future research have also been given.

Appendix A

Definitions of a GIS

Year of Reference	Author	Definition
1989	Arnoff	"any manual or computer-based set of procedures used to store and manipulate geographically referenced data."
1989	Carter	"an institutional entity, reflecting an organisational structure that integrates technology with a database, expertise, and continuing financial support over time.
1989	Koshkariov *et al.*	"a system with advanced geo-modelling capabilities.
1989	Parker	"an information technology which stores, analyses, and displays both spatial and non-spatial data."
1988	Cowen	"a decision support system involving the integration of spatially referenced data in a problem-solving environment.
1987	Smith *et al.*	"a database system in which most of the data are spatially indexed, and upon which a set of procedures operated in order to answer queries about spatial entities in the databases."
1987	US Department of Environment	"a system for capturing, storing, checking, manipulating, analysing, and displaying data which are spatially referenced to the Earth."
1986	Burrough	"a powerful set of tools for collecting, storing, retrieving, at will, transforming and displaying spatial data from the real world."
1986	Devine & Field	"a form of MIS [Management Information System] that allows map display of the general information.
1981	Ozemoy *et al*	"an automated set of functions that provides professionals with advanced capabilities for the storage, retrieval, manipulation, and display of geographically located data.
1979	Dueker	"a sspecial case of information systems where the database consists of observations on spatially distributed features, activities, or events, which are definable in space as oints, lines or areas. A GIS manipulates data about these points, lines and areas to retrieve data for *ad hoc* queries and analyses."

Source: Cited in [Mennecke 1997]

Appendix B

The VAX OPS5 Structure

VAX OPS5, a production system language, consists of a **declaration section** that describes the data objects of the programme, followed by a **production section** that contains the rules. During execution, the data operated on by the programme are kept in the **working memory.** The rules are maintained in the **production memory.** Working memory is usually initialized after the declarations and rules have been loaded. The declaration section contains the definitions of the data object types and all *user-defined* functions that can be referenced in the rules

The **inference engine** in VAX OPS5, consisting of the three algorithms for "match-rules," "select-rules," and "execute-rules," directly supports forward chaining, even though backward chaining can be implemented [Digital 1988].

B. 1. DATA TYPES

VAX OPS5 uses two primitive data types, also known as *scalar types:* **numbers** and **symbolic atoms**. Symbolic atom is any sequence of characters that is not a number and which can be treated as a single unit. The compound data structure type definable in a VAX OPS5 programme is **element class**. The components of an element class are called **attributes**. An element class declaration is similar to a structure declaration in Cobol or a record declaration in Pascal.

VAX OPS5 is characterized by:

A global database.
Condition-action (or if-then) rules programmed in the form of productions, which operate on the global database.
Productions that are executed in an unspecified order.
Computations with symbolic expressions and numbers.
Simple syntax.

As mentioned earlier (Appendix B, Para 1), VAX OPS5 has two key components: **working memory (wm)** and **production memory** Working memory is a global database of information pertaining to the problem domain. This information is stored in elements which are grouped into classes. Elements storing similar information can be grouped in the same class. WM can contain several classes of elements, and each class can have more than one element. WM is dynamic. While an OPS5 programme is executed, elements are added, deleted, or modified, continually

A working-memory element (**wme**) is a sequence of atoms that represents an object or a concept. Each atom is stored in a field that can be labelled with an attribute name. A wme can be represented using a combination of the following:

A class-name;
A list of scalar attributes and their values;
A vector attribute and its value.

The class-name identifies an element's class, and the attributes and their values describe the element's characteristics. The value of each scalar attribute is an atom. The value of a vector attribute can be one or more atoms. The format for specifying a wme is as follows:

[class-name] [{scalar-attribute value}...] [vector-attribute value]

A class-name is a symbol that identifies a group of similar elements. Elements that have the same class-name have the same attributes; but the values of the attributes are different.

An attribute consists of the attribute designator (^), followed by the name of the attribute which descibes the element's characteristics.

The value of a scalar attribute is an atom, *e.g.*, ^number 102

The value of a vector attribute is a list of one or more atoms, *e.g.*, ^date 2 Jan 1988

B. 2. INTERNAL REPRESENTATION OF WORKING-MEMORY ELEMENTS

Working memory elements are marked with a **time tag** or **recency attribute**, which is used as a part of conflict resolution. Larger values of the time tag indicate more recent elements. When an element in working-memory gets modified the element gets a new time tag. When the OPS5 "examine working memory" command (wm) is issued, each working memory element is displayed preceded by an integer, which is the time tag for the element.

The first field of the structure that holds a wme is reserved for the class-name of the element. If an element does not have a class-name, the run-time places **NIL** in the first field. The compiler assigns fields to the names of the scalar attributes when the attribute names are declared. An attribute's field stores that attribute's value. The field assigned to each attribute name is global meaning that, the attribute name refers to the same field for each element-class in which the attribute name appears.

Declarations: Attribute names and external routines must be declared using the **literalize**, **literal**, **vector-attribute**, or **external** declaration, and they must appear before other kinds of statements of the programme. Literalize declaration does the following:

associates a class with a list of attribute names;
signals to the compiler to assign unique fields to the specified attribute names.

```
(literalize road
        name
        start-point
        end-point
        junctions)
```

The above declaration associates the class-name "road" with the attribute names, "name," "start-point," "end-point," and "junctions." This declaration is also an indication to the compiler to assign appropriate fields to the attribute names.

B. 3. PRODUCTIONS

Each production is of the form

```
(p production-name
        (condition-element-1)
        (condition-element-2)

        (condition-element-n)
-->
        (action-1)
        (action-2)

        (action-m))
```

The p that follows the opening parenthesis signifies that the code following it is a production. The "production-name" distiguishes the production from other productions in a programme. The LHS of the production is separated from the RHS by an arrow The name of each production within a programme must be unique. A production can be named with any string except "NIL."

B. 3. 1. The Left-Hand Side

Condition-elements on the LHS of the production can be positive or negative. The first condition-element must be positive. A LHS can have a maximum of 64 positive condition-elements and and any number of negative condition-elements. The values that can be specified in the components of a condition-element can be constant atoms, variables or function calls. They can be preceded by a predicate indicating the comparison operation (equal to, greater than, *et cetra*) to be performed.

In OPS5, **constants** can be symbols, integers, or real numbers. A **variable** is a symbol enclosed in angle brackets (< >).

Conjunctions: A conjunction is a pattern of conditional tests (similar to logical AND), all of which must be true of an atom in a working memory element. A conjunction can be specified by enclosing the list of conditional tests in braces ({}).

Disjunctions: A disjunction is a pattern containing a list of constant atoms. Only the atoms specified in the list can match the pattern. A disjunction is similar to logical inclusive OR. A disjunction can be specified by enclosing the list of constant atoms between double angle brackets (<< >>).

B. 3. 2. The Right-Hand Side

The RHS of a production consists of one or more actions. Actions perform the following operations:

Modify working-memory
Save and restore the state of working-memory and the conflict set.
Stop programme execution
Bind variables
Manipulate files
Write output
Control loops
Add productions to executing programmes.
Call external sub-programmes

An action includes an action-name and its attributes enclosed within parentheses.

B. 4. CONFLICT RESOLUTION STRATEGIES

Instantiations are the objects in the conflict set. An instantiation is an ordered pair consisting of a production name and a list of working-memory elements satisfying the production's LHS. A conflict set is a collection of ordered pairs of the form:

< Production-name, List of elements matched by production's LHS >.

During the conflict resolution step, the OPS5 interpreter examines the conflict set to find an instantiation that dominates all the others under a set of ordering rules called a "conflict resolution strategy" OPS5 provides two such strategies, LEX and MEA [Brownston *et al.* 1985].

When more than one rule is created in the conflict set, conflict resolution is achieved using one of the two strategies: LEX and MEA. LEX is the simpler of the two. Both the strategies use the following concepts:

Refraction. Selects an instantiation only once. Refraction prevents programmes from looping indefinitely on the same data by removing instantiations from the conflict set after they have been selected.

Recency. Selects the instantiations that refers to the most recent data in working-memory, as indicated by the time tag attached to each wme (*vide* Section B.2). Therefore, the system must select the instantiation that contains the highest time tag.

Specificity. Selects an instantiation of a production whose LHS is the most specific. Specificity of each production in a programme is calculated by counting the number of condition tests on the LHS. The production with the maximum number of conditional tests is the most specific.

Both strategies, LEX and MEA, apply the rules in the following order: refraction, recency, specificity However, MEA strategy includes an extra step after refraction, which helps to organize large programmes. This step orders the instantiation according to the recency of the wme, matching the first condition-element in each production. The default strategy for VAX OPS5 is LEX. This can be changed to MEA, if required, through the **strategy** command with MEA as the keyword.

LEX Strategy. LEX is suitable for programmes that do not depend on the order in which the productions are executed—the normal case for production systems. LEX uses the following rules in sequence to order the instantiations in the conflict set.

> Apply refraction by removing from the conflict set the instantiations the run-time system has selected during the previous cycle.

Order the rest of the instantiations according to their recency, and select the instantiation with the highest level of recency.
If more than one instantiation has the highest level of recency, order those instantiations according to their specificity, and select the instantiation with the highest level of specificity.
If more than one instantiation has the highest level of specificity, select an instantiation arbitrarily.

MEA Strategy. MEA places the highest priority on the production whose first condition-element is matched by the most recent wme. This is most appropriate when the most important condition-element is placed first on the LHS of each production. MEA uses the following rules in sequence to order the instantiations in the conflict set.

Apply refraction by removing from the conflict set the instantiations the run-time system has selected during the previous cycle.
Compare the first time tag of each instantiation still remaining in the conflict set and select the instantiation with the highest level of recency.
If more than one instantiation has the highest level of recency for the first time tag, order those instantiations according to their recency (using all time tags) and select the instantiation with the highest level of recency.

After the run-time system has selected an instantiation, the recognize-act cycle enters the "execute" phase. During this phase, variables are bound to values, and the actions on the RHS of the productions to which the selected instantiations refer are executed.

B. 5. ORGANIZATION AND CONTROL IN OPS5

By exploiting the built-in capabilities of OPS5, a production system designer can exert the desired degree of control over the execution of the system. In OPS5, the built-in structured data types are element-classes and vector attributes. The **substr** function can be used to implement data structures such as arrays, strings, stacks, queues, and dequeues, depending on the set of operations permitted on the vector

Because production systems access data by *value*, rather than by address, it is easier to create linked data structuresin OPS5. Links are traced by the process of matching rather than dereferencing, and this makes the linked data structures easier to manage. Linking allows implementation of graphs such as goal trees, semantic networks, frames and other complex structures.

The OPS5 interpreter does not simply test the expressions to determine whether they are true; rather it engages in a search to determine if it can bind variables in such a way as to make the expression true. It is this characteristic of pattern matching, along with dynamic size of working-memory that gives production systems their distictive capabilities.

B. 6. LEARNING IN OPS5

Learning is any change in behaviour of a system that contributes to the enhancement of the system's performance. There are different approaches to machine learning. Early programmes affected learning through self modification of stored parameters. In the production system model, incremental changes in performance are effected by adding new production rules to an existing rule base. Researchers have also investigated a form of learning that results from changes in working-memory and reorganisation of existing rules [Brownston *et al.* 1985].

Learning can be accomplished in two ways, either by remembering the results of previous search or by learning the definition of an object more precisely so that the search space may be pruned rapidly Learning, therefore, may be classified as either rôte learning or inductive learning.

B. 6. 1. Rôte Learning

Rôte learning allows the system to memorise the examples of an object for which it has already searched so that, when it is asked to search for the same object again, it retrieves the previous examples instead of searching again. It stores only pre-defined high level objects. Known examples are stored in a database that is separate from the spatial object KB. The decision on whether an example should be stored or not depends on factors such as complexity of the object and the frequency with which they are sought.

B. 6. 2. Inductive Learning

Inductive learning is used to provide a new definition for an object from a given set of examples so that the search for an object can proceed more efficiently To learn a new definition of an object, the user can give example definitions, by specifying the appropriate values, and the system generates the definitions using these properties.

B. 6. 3. Strengthening and Weakening

One way of implementing learning in a production system is to tune a set of parameters so that they converge on a set of optimal values. A number can be associated with each rule to indicate the desirability of its firing. The probability that this particular rule will fire can be raised by increasing this number. Such a number is often called the rule's strength. Naturally enough, incrementing the strength is called *strengthening* and decrementing the strength is called *weakening*.

B. 6. 4. Generalization

A system that learns by generalization, starts with a set of overly specific rules and generates new rules that are more general and abstract. These inferred rules may improve performance even if they are incorrect, although they are not guaranteed to do so. If the system overgeneralizes, it may be necessary to backtrack or to specialize some of the rules that are too general.

B. 7. SUMMARY

Many people associate knowledge-based systems with rule-based systems. We can have knowledge-intensive problem solving without using rule-based representation and inferencing, although rules or implications are indeed one of the most common ways of expressing knowledge. Rule-based systems are a kind of production system that utilizes rules of deductive reasoning to reach logical conclusions. Production systems are used most frequently by interpreting these rules (1) as condition-action rules in forward-chaining control or (2) as sets of logical implications, from which deductions are drawn, in backward-chaining control.

OPS5 is a forward-chaining production system development language. The use of sophisticated indexing scheme, known as the Rete algorithm, for finding rules that match the current rule-base makes OPS5 one of the tools that executes the fastest.

When more than one rule become qualified to be fired, two OPS5 conflict-resolution strategies are available: LEX and MEA. LEX is the simpler of the two. MEA (means end analysis) places additional emphasis on the recency of a wme that matches the first rule condition.

Appendix C

C1. The START Menu

```
      SUBROUTINE  startmenu()
C     Subroutine for displaying the START Menu
      implicit  integer (a-z)
      include  'ops$library:opsdef.for'
      include  'sys$library:uisentry'
      include  'sys$library:uisusrdef'
      integer*4  colindx,keybuf
      character*(*)  title
      common  /cb1/vd_id1,wd_id1,vd_id2,wd_id2,vd_id5,wd_id5,keybuf
      common  /cd/vd_idm,wd_idm
      external create_window1,kerala1,hilights,hilight2,delwidm
      parameter  (title =  'Start Menu')

      call create_window1 (vd_idm,wd_idm,-1.0,-1.0,4.0,2.0,5.0,3.0,title,-1.0,-1.0,
     2    4.0,2.0,5.0,3.0,0.0,0.0)

      call  uis$set_color (vd_idm,0,0.8,0.9,0.0)
      call  uis$set_color(vd_idm,1,1.0,0.0,0.0)
      call  uis$set_writing_index(vd_idm,0,9,1)
      call  uis$text(vd_idm,0, '1.Kerala',0.0,1.5)
      call  uis$text(vd_idm,0, '2.Exit',0.0,0.50)

      call  uis$set_pointer_ast(vd_idm,wd_idm,hilights,,-1.0,1.0,4.0,1.50,
     2  hilight2,%ref(%loc(vd_idm)))
      call  uis$set_button_ast(vd_idm,wd_idm,keral1,,keybuf,-1.0,1.0,4.0,1.50)
      call  uis$set_pointer_ast(vd_idm,wd_idm,hilights,,-1.0,0.0,4.0,0.50,
     2    hilight2,%ref(%loc(vd_idm)))
      call  uis$set_button_ast(vd_idm,wd_idm,delwidm,,keybuf,-1.0,0.0,4.0,0.50)
      return
      end
```

C2. Selecting theTrivandrum District

```
      SUBROUTINE kerala()
C     Displays the map of Kerala with all districts from which the Trivandrum District is selected
      implicit integer (a-z)
      include 'sys$library:uisentry'
      include 'sys$library:uisusrdef'
      common /cb1/vd_id1,wd_id1,vd_id2,wd_id2,vd_id5,wd_id5,keybuf
      common /cb4/keycode,vd_id6,wd_id6,vd_ide,wd_ide
      common /cd/vd_idm,wd_idm
      character*20 rname,rdname
      real x1,y1,x2,y2
      character*(*) title
      external plotlinek,plotpointk,create_window1,hmenu,tvm
      parameter (title = 'KERALA STATE')

      call create_window1(vd_id6,wd_id6,0.0,0.060.0,60.0,25.0,25.0,title,
 2    0.0,0.0,60.0,60.0,25.0,25.0,8.0,15.0)
      call hcuis$read_display(vd_id6,'kerala.uis')
      rd_name = 'dstnames.dat'
      call plotpointk(rdname,vd_id6)
      call uis$text(vd_id6,0,'Move the pointer inside the district boundary ',1.0,60.0)
      call uis$text(vd_id6,0,'Click "left mouse button" to select',1.0,59.0)
      call uis$text(vd_id6,0,'Click "Right mouse button" to Delete Map',1.0,58.0)
      call uis$set_button_ast(vd_id6,wd_id6,tvm,,keybuf,41.48,0.78,47.85,6.36)
      return
      end

      SUBROUTINE tvm()
C     Invoking the Trivandrum District Menu
      implicit integer (a-z)
      include 'sys$library:uisentry'
      include 'sys$library:uisusrdef'
      common /cb1/vd_id1,wd_id1,vd_id2,wd_id2,vd_id5,wd_id5,keybuf
      common /cb4/keycode,vd_id6,wd_id6,vd_ide,wd_ide
      common /cbc/vd_id7,wd_id7
      external kerala
      if (keybuf .eq. uis$c_pointer_button_3) then
              call uis$delete_window(wd_id6)
              call startmenu
      else if (keybuf .eq. uis$c_pointer_button_1) then
              call uis$delete_window(wd_id6)
              call trivandrum
      end if
      return
      end

      SUBROUTINE trivandrum()
C     Displays the map of Trivandrum District
      implicit integer (a-z)
      include 'sys$library:uisentry'
```

```
      include 'sys$library:uisusrdef'
      integer*(*) title
      common /cb1/vd_id1,wd_id1,vd_id2,wd_id2,vd_id5,wd_id5,keybuf
      common /cb4/keycode,vd_id6,wd_id6,vd_ide,wd_ide
      common /cbc/vd_id7,wd_id7
      external plotlinek,plotpointk,create_window1,tvmcity
      data vcm_size/16/
      parameter (title = 'Trivandrum District' )
      lwidth = 1.0
      call create_window1(vd_id7,wd_id7,0.0,0.0,60.0,60.0,25.0,25.0,title,
2       40.0,0.0,50.0,10.0,25.0,25.0,8.0,3.0)
      rname = 'tvm.dat'
      call plotlinek (rname,vd_id7)
      call uis$set_color (vd_id7,0,0.8,0.8,0.0)
      call uis$set_color (vd_id7,1,1.0,0.0,0.0)
      call uis$set_writing_index (vd_id7,0,8,1)
      call uis$set_line_width(vd_id7,8,9,lwidth,wdpl$c_width_world)
      call uis$text(vd_id7,8, 'TRIVANDRUM  DISTRICT',45.0,4.8)
      call uis$text(vd_id7,0, 'Click "left mouse button" to select',40.0,10.50)
      call uis$text(vd_id7,0, 'Click "Right mouse button" to Delete Map',40.0,10.0)
      call uis$set_button_ast(vd_id7,wd_id7,tvmcity,,keybuf,40.0,0.0,50.0,10.0 )
      return
      end
```

C3. Code for Trivandrum City Menu

```
      SUBROUTINE tvmcity()
C     Invokes the Trivandrum city menu
      implicit integer (a-z)
      include 'sys$library:uisentry'
      include 'sys$library:uisusrdef'
      integer*(*) title
      common /cb/vd_idt,wd_idt
      common /cb1/vd_id1,wd_id1,vd_id2,wd_id2,vd_id5,wd_id5,keybuf
      common /cbc/vd_id7,wd_id7
      external tvmmenu , kerala
      parameter (title = ' Thiruvananthapuram City' )

      if (keybuf .eq. uis$c_pointer_button_3) then
            call uis$delete_window(wd_id7)
            call kerala
            else if (keybuf .eq. uis$c_pointer_button_1) then
                  call uis$delete_window(wd_id7)
                  call tvmmenu
            end if
      return
      end

      SUBROUTINE tvmmenu()
C     Displays menu for Trivandrum City
      implicit integer (a-z)
      include 'sys$library:uisentry'
      include 'sys$library:uisusrdef'
      character*(*) title
      integer ovlswitch
      common /kb/kb_id,vcm_id1
      common /cb/vd_idt,wd_idt
      common /cb1/vd_id1,wd_id1,vd_id2,wd_id2,vd_id5,wd_id5,keybuf
      common /cbc/vd_id7,wd_id7
      common /os/ ovlswitch
      external test,test1,test3,hilightc.hilight2, util
      data vcm_size/16/
      parameter (title = ' Thiruvananthapuram City' )
      ovlswitch = 0
      call create_window1 (vd_idt,wd_idt,0.0,0.0,8.0,3.0,8.0,3.0,title,
    2  0.0,0.0,8.0,3.0,8.0,3.0,0.0,0.0 )
      call uis$set_color (vd_idt,0,1.0,1.0,0.0)
      call uis$set_color (vd_idt,1,1.0,0.0,0.0)
      call uis$set_writing_index (vd_idt,0,9,1)
      call uis$text(vd_idt,0, '1. Transportation System ',1.0,2.5 )
      call uis$text(vd_idt,0, '2. Display Informatin ',1.0,1.5 )
      call uis$text(vd_idt,0, '3. Exit this Menu ',0.0,1.0 )
      call uis$set_pointer_ast(vd_idt,wd_idt, hilightc,, 0.0,2.0,8.0,2.5,hilight2,%ref(%loc(vd_idt)))

      call uis$set_button_ast(vd_idt,wd_idt, test1,, keybuf,0.0,2.0,8.0,2.5)
```

```
call uis$set_pointer_ast(vd_idt,wd_idt, hilightc,, 0.0,1.0,8.0,1.5,hilight2,%ref(%loc(vd_idt)))
call uis$set_button_ast(vd_idt,wd_idt, test3,, keybuf,0.0,1.0,8.0,1.5)
call uis$set_pointer_ast(vd_idt,wd_idt, hilightc,, 7.0,0.0,7.5,0.5,hilight2,%ref(%loc(vd_idt)))
call uis$set_button_ast(vd_idt,wd_idt, test,, keybuf,7.0,0.0,7.5,0.5 )
call util
return
end
```

C4. Code for Zooming Selected Area of the Map

```
      SUBROUTINE gozoom( )
C     Zooming a selected area of the map
      implicit integer (a-z)
      include 'sys$library:uisentry'
      include 'sys$library:uisusrdef'
      integer*4 keycode,button
      integer box_start_flag,switch
      character*15 hcopy
      real*4 mfact
      real*4 xmin,ymin,xmax,ymax,stx,sty,x1,y1,x2,y2
      common /cb1/vd_id1,wd_id1,vd_id2,wd_id2,vd_id5,wd_id5,keybuf
      common /cb2/button,keylist
      common /blk1/box_start_flag
      common /blk2/xmin,ymin,xmax,ymax,stx,sty,x1,y1,x2,y2
      common /sw/switch
      common /mul/mfact
      common /zmw/vd_idz,wd_idz
      common /sws/fswitch,tswitch,pswitch,zswitch,mswitch
      common /hc/hcopy
      external zoomsize,create_window2

      if (button .eq. uis$c_pointer_button_1) then
        if (zswitch .lt. 1) then
        zswitch = 1
        call create_window2(vd_idz,wd_idz,0.0,0.0,12.0,1.0,
     2  12.0,1.0,0.0,0.0,12.0,1.0,12.0,1.0,19.0,24.7)

      call uis$set_color(vd_idz,0,0.8,0.0,0.0)
      call uis$set_color(vd_idz,1,1.0,1.0,0.0)
      call uis$set_writing_index(vd_idz,0,1,1)
      call uis$plot(vd_idz,1,0.1,0.9,0.1,0.1,11.9,0.1,11.9,0.9,0.1,0.9)
      call uis$plot(vd_idz,1,1.5,0.1,1.5,0.9)
      call uis$plot(vd_idz,1,3.0,0.1,3.0,0.9)
      call uis$plot(vd_idz,1,4.5,0.1,4.5,0.9)
      call uis$plot(vd_idz,1,6.0,0.1,6.0,0.9)
      call uis$plot(vd_idz,1,7.5,0.1,7.5,0.9)
      call uis$plot(vd_idz,1,9.0,0.1,9.0,0.9)
      call uis$plot(vd_idz,1,10.5,0.1,10.5,0.9)
      call uis$plot(vd_idz,1,12.0,0.1,12.0,0.9)
      call uis$set_color(vd_idz,2,1.0,1.0,1.0)
      call uis$set_writing_index(vd_idz,2,3,2)
      call uis$set_char_size(vd_idz,3,4,,0.4,0.4)
      call uis$text(vd_idz,4,'10x',0.2,0.8)
      call uis$text(vd_idz,4,'20x',1.7,0.8)
      call uis$text(vd_idz,4,'40x',3.2,0.8)
      call uis$text(vd_idz,4,'50x',4.7,0.8)
      call uis$text(vd_idz,4,'75x',6.2,0.8)
      call uis$text(vd_idz,4,'100x',7.55,0.8)
      call uis$text(vd_idz,4,'150x',9.05,0.8)
```

```
call uis$text(vd_idz,4,'200x',10.6,0.8)
call  uis$get_window_size(vd_id1,wd_id1,xmin,ymin,xmax,ymax)
call uis$set_button_ast(vd_idz,wd_idz,zoomsize,,keybuf,0.0,0.0,1.5,1.0)
call uis$set_button_ast(vd_idz,wd_idz,zoomsize,,keybuf,1.5,0.0,3.0,1.0)
call uis$set_button_ast(vd_idz,wd_idz,zoomsize,,keybuf,3.0,0.0,4.5,1.0)
call uis$set_button_ast(vd_idz,wd_idz,zoomsize,,keybuf,4.5,0.0,6.0,1.0)
call uis$set_button_ast(vd_idz,wd_idz,zoomsize,,keybuf,6.0,0.0,7.5,1.0)
call uis$set_button_ast(vd_idz,wd_idz,zoomsize,,keybuf,7.5,0.0,9.0,1.0)
call uis$set_button_ast(vd_idz,wd_idz,zoomsize,,keybuf,9.0,0.0,10.5,1.0)
call uis$set_button_ast(vd_idz,wd_idz,zoomsize,,keybuf,10.5,0.0,12.0,1.0)
end if
else if (button .eq.  uis$c_pointer_button_3 ) then
if  (zswitch .gt.  0) then
call uis$delete_window(wd_idz)
zswitch = 0
end if
end if
return
end

 SUBROUTINE  zoomsize( )
C       Selects the zooming ratio
 implicit  integer (a-z)
 include  'sys$library:uisentry'
 include  'sys$library:uisusrdef'
 real*4 mfact,x,y
 real*4 xmin,ymin,xmax,ymax,stx,sty,x1,y1,x2,y2
common  /cb1/vd_id1,wd_id1,vd_id2,wd_id2,vd_id5,wd_id5,keybuf
 common  /cb2/button,keylist
 common  /blk1/box_start_flag
 common  /blk2/xmin,ymin,xmax,ymax,stx,sty,x1,y1,x2,y2
 common  /sw/switch
common  /mul/mfact
 common  /zmw/vd_idz,wd_idz
 common  /sws/fswitch,tswitch,pswitch,zswitch,mswitch
 external  update

 if  (keybuf  .lt.  0) then
call uis$get_pointer_position(vd_idz,wd_idz,x,y)
call uis$delete_window(wd_idz)
zswitch  = 0
box_start_flag = 0
if (( y .gt. 0.0)  .and.  (y  .lt.  1.0) )  then
if (( x .gt. 0.0)  .and.  (x  .lt.  1.5) )  then
mfact = 0.10
else if (( x .gt.  1.5)  .and.  (x  .lt.  3.0) )  then
mfact = 0.20
else if (( x .gt. 3.0)  .and.  (x  .lt.  4.5) )  then
mfact = 0.40
else if (( x .gt. 4.5)  .and.  (x  .lt.  6.0) )  then
mfact = 0.50
else if (( x .gt. 6.0)  .and.  (x  .lt.  7.5) )  then
mfact = 0.75
else if (( x .gt. 7.5)  .and.  (x  .lt.  9.0) )  then
```

```
      mfact = 1.0
      else if (( x .gt. 9.0) .and. (x .lt. 10.5) ) then
      mfact = 1.5
      else if (( x .gt. 10.5) .and. (x .lt. 12.0) ) then
       mfact = 2.0
      end if
      call uis$set_button_ast(vd_id1.wd_id1.update..button.xmin.ymin.xmax.ymax)
      end if
      end if
      return
      end
      SUBROUTINE update( )
C     Allows user to select an area of the map
      implicit integer (a-z)
      include 'sys$library:uisentry'
      include 'sys$library:uisusrdef'
      integer box_start_flag
      real*4 xmin.ymin.xmax.ymax.stx.sty.x1.y1.x2.y2
      common /cb1/vd_id1.wd_id1.vd_id2.wd_id2.vd_id5.wd_id5.keybuf
      common /cb2/button.keylist
      common /blk1/box_start_flag
      common /blk2/xmin.ymin.xmax.ymax.stx.sty.x1.y1.x2.y2
      common /mul/mfact
      external rubberband.gdum
       if (button .eq. uis$c_pointer_button_1) then
      call uis$get_pointer_position(vd_id1.wd_idz1.stx.sty)
      call uis$delete_window(wd_idz)
      x1 = stx
      x2 = =stx
      y1 = =sty
      y2 = sty

C     Draw first "degenerate" box
      call uis$plot(vd_id1.0.x1.y1.x2.y1.x2.y2.x1.y2.x1.y1)
      call uis$set_button_ast(vd_id1.wd_id1....xmin.ymin.xmax.ymax)
C     Set up pointer ast for rubberband
      call uis$pointer_ast(vd_id1.wd_id1.rubberband..xmin.ymin.xmax.ymax..)
      call uis$set_button_ast(vd_id1.wd_id1.gdum..keybuf.xmin.ymin.xmax.ymax)
      end if
      return
      end

      SUBROUTINE gdum( )
      implicit integer (a-z)
      include 'sys$library:uisentry'
      include 'sys$library:uisusrdef'
      integer box_start_flag
      real*4 xmin.ymin.xmax.ymax.stx.sty.x1.y1.x2.y2
      common /cb1/vd_id1.wd_id1.vd_id2.wd_id2.vd_id5.wd_id5.keybuf
      common /cb2/button.keylist
      common /blk1/box_start_flag
      common /blk2/xmin.ymin.xmax.ymax.stx.sty.x1.y1.x2.y2
      common /mul/mfact
      common /hc/hcopy
```

```
external   rubberband.gozoom2.zoomstop1

if (keybuf .eq. uis$c_pointer_button_3) then
box_start_flag = 1
call uis$set_writing_index(vd_id1,0,7,0)
call uis$plot(vd_id1,7,x1,y1,x2,y1,x2,y2,x1,y2,x1,y1)
call uis$set_writing_index(vd_id1,0,8,1)
call uis$plot(vd_id1,8,x1,y1,x2,y1,x2,y2,x1,y2,x1,y1)
call uis$set_pointer_ast(vd_id1,wd_id1,0.,0.0,0.0,0.0,0.0,0.0,.)
call uis$set_button_ast(vd_id1,wd_id1,,,,xmin,ymin,xmax,ymax)
call hcuis$read_display(vd_id1,'tvm.uis')
call gozoom2
call uis$set_button_ast(vd_id1,wd_id1,zoomstop1,,keycode,0.0,0.0,100.0,80.0)
end if
return
end


 SUBROUTINE  rubberband( )
 implicit  integer (a-z)
 include  'sys$library:uisentry
 include  'sys$library:uisusrdef'
 integer  box_start_flag,button,keybuf
 real*4 xmin,ymin,xmax,ymax,stx,sty,x1,y1,x2,y2
common  /cb1/vd_id1,wd_id1,vd_id2,wd_id2,vd_id5,wd_id5,keybuf
 common  /cb2/button,keylist
 common  /blk1/box_start_flag
 common  /blk2/xmin,ymin,xmax,ymax,stx,sty,x1,y1,x2,y2
 external   gdum,gozoom2

 if (box_start_flag  .eq.  1) then
call uis$set_pointer_ast(vd_id1,wd_id1,0.,0.0,0.0,0.0,0.0,.)
else
call uis$get_pointer_position(vd_id1,wd_id1,stx,sty)
call uis$set_writing_index(vd_id1,0,7,0)
call uis$plot(vd_id1,7,x1,y1,x2,y1,x2,y2,x1,y2,x1,y1)

x2 = stx
y2 = sty
call uis$set_writing_index(vd_id1,0,8,1)
call uis$plot(vd_id1,8,x1,y1,x2,y1,x2,y2,x1,y2,x1,y1)
end if
return
end

 SUBROUTINE  gozoom2( )
 implicit  integer (a-z)
 include  'sys$library:uisentry'
 include  'sys$library:uisusrdef'
 integer *4  keycode
 real  x1,y1,zx11,zy11,zx21,zy21
common  /cb1/vd_id1,wd_id1,vd_id2,wd_id2,vd_id5,wd_id5,keybuf
 common  /cb2/button,keycode
 common  /cb3/vd_id3,wd_id3,vd_id9,wd_id9,vd_id10,wd_id10
```

```
common /cb5/zx1,zx2,zy1,zy2
common /cb6/wd_id4
external zoomstop1
structure/placelist/
integer*4           code_1
real*4              abs_pos_x
integert*4          code_2
real*4              abs_pos_y
integer*4           end_of_list
end structure
record/placelist/placelist2

placelist2.code_1 = wdpl$c_abs_pos_x
placelist2.abs_pos_x = 16.0
placelist2.code_2 = wdpl$c_abs_pos_y
placelist2.abs_pos_y = 10.0
placelist2.end_of_list = wdpl$c_end_of_list

if (keycode .eq. uis$c_pointer_button_1) then
status = uis$get_pointer_position(vd_id1,wd_id3),x1,y1)
zx11 = x1 - 2.0
zy11 = y1 - 2.0
zx21 = x1 + 2.0
zy21 = y1 + 2.0
if (zx11 .lt. 0.0) then
zx11 = 0.0
end if
if (zy11 .lt. 0.0) then
zy11 = 0.0
end if
if (zx21 .gt. zx2) then
zx21 = zx2
end if
if (zy21 .gt. zy2) then
zy21 = zy2
end if
wd_id4 = uis$create_window(vd_id1,'sys$workstation' 'zoomed View',zx11,zy11,
2   zx21,zy21,10.0,10.0,placelist2)
call uis$pop_view_port(wd_id3)
call uis$set_button_ast(vd_id1,wd_id3,zoomstop1,,keycode,zx1,zy1,zx2,zy2)
end if
return
end

SUBROUTINE zoomstop1( )
implicit integer (a-z)
include 'sys$library:uisentry
include 'sys$library:uisusrdef
common /cb2/button,keycode
common /cb3/vd_id3,wd_id3,vd_id9,wd_id9,vd_id10,wd_id10
common /cb5/zx1,zx2,zy1,zy2
common /cb6/wd_id4
if (keycode .eq. uis$c_pointer_button_2) then
call uis$delete_window(wd_id4)
```

```
end if
return
end
```

C5. Code for Important Roads and Rivers Menu

```
      SUBROUTINE rdmenu( )
C     Program fragments that displays the distribution of major roads
      implicit integer (a-z)
      include 'sys$library:uisentry'
      include 'sys$library:uisusrdef'
      character*(*) title
      common /cb1/vd_id1,wd_id1,vd_id2,wd_id2,vd_id5,wd_id5,keybuf
      common /cb2/button,keycode
      common /cb3/vd_id3,wd_id3,vd_id9,wd_id9,vd_id10,wd_id10
      common /wd2/vd_id20,wd_id20,vd_id21,wd_id21
      external roaddbtn,roadpart,riverdbtn,riverpart
      parameter (title = 'Roads')
      if(keybuf .lt. 0) then
      call uis$delete_window(wd_id9)
      call create_window1(vd_id20,wd_id20,-1.0,-1.0,9.0,3.0,10.0,4.0,
    2 title,-1.0,-1.0,9.0,3.0,10.0,4.0,0.0,0.0)

      call uis$set_color(vd_id20,0,1.0,1.0,0.0)
      call uis$set_color(vd_id20,1,1.0,0.0,0.0)
      call uis$set_writing_index(vd_id20,0,9,1)
      call uis$text(vd_id20,9,'1. Layout of Major Roads',0.0,2.50)
      call uis$text(vd_id20,9,'2. Particular Road',0.0,1.50)
      call uis$text(vd_id20,9,'3. Layout of Rivers & canals',0.0,0.50)
      call uis$text(vd_id20,9,'4. Particular River',0.0,-0.50)
      call uis$set_button_ast(vd_id20,wd_id20,roaddbtn,,button,0.0,2.0,9.0,2.50)
      call uis$set_button_ast(vd_id20,wd_id20,roadpart,,button,0.0,1.0,9.0,1.50)
      call uis$set_button_ast(vd_id20,wd_id20,riverdbtn,,button,0.0,0.0,9.0,0.50)
      call uis$set_button_ast(vd_id20,wd_id20,riverpart,,button,0.0,-1.0,9.0,-0.50)

      end if
      return
      end

      SUBROUTINE roaddbtn( )
      implicit integer (a-z)
      include 'sys$library:uisentry'
      include 'sys$library:uisusrdef'
      common /cb1/vd_id1,wd_id1,vd_id2,wd_id2,vd_id5,wd_id5,keybuf
      common /cb2/button,keycode
      common /cb3/vd_id3,wd_id3,vd_id9,wd_id9,vd_id10,wd_id10
      common /wd2/vd_id20,wd_id20,vd_id21,wd_id21
      common /os/ovlswitch
      external window,priorityrd,util

      if(button .lt. 0) then
      call uis$delete_window(wd_id20)
      if (ovlswitch .ne. 1) then
      call window
      call util
```

```
      call priorityrd
      end if
      return
      end

       SUBROUTINE priorityrd( )
C      Display the priority menu
       implicit integer (a-z)
       include 'sys$library:uisentry'
       include 'sys$library:uisusrdef'
       integer dswitch,prn,cnt
       character*(*) title
      common /cb1/vd_id1,wd_id1,vd_id2,wd_id2,vd_id5,wd_id5,keybuf
      common /cy/x1,y1,x2,y2
      common /txtm/vd_ids,wd_ids,vd_idp,wd_idp
      common /ds/dswitch,cnt,prn
      common /sws/fswitch,tswitch,pswitch,zswitch,mswitch
      external priorityrd1,priorityrd2,priorityrd3,priorityrd4,hilightc,hilight2,stptxt
      data vcm_size/16/
      parameter (title = 'PRIORITY MENU')

      call create_window1(vd_idp,wd_idp,0.0,0.0,5.0,4.0,5.0,4.0,title,
     2 0.0,0.0,5.0,4.0,5.0,4.0,0.0,0.0)

      call uis$set_color(vd_idp,0,1.0,1.0,0.0)
      call uis$set_color(vd_idp,1,1.0,0.0,0.0)
      call uis$set_writing_index(vd_idp,0,9,1)
      call uis$text(vd_idp,0,'1. Priority 1',0.2,3.5)
      call uis$text(vd_idp,0,'2. Priority 2',0.2,2.50)
      call uis$text(vd_idp,0,'3. Priority 3',0.2,1.50)
      call uis$text(vd_idp,0,'4. Priority 4',0.2,0.50)
      call uis$plot(vd_idp,0,4.0,0.5,4.5,0.5,4.25,0.0,4.0,0.5)
      call uis$set_button_ast(vd_idp,wd_idp,priorityrd1,,keybuf,0.0,3.0,5.0,4.0)
      call uis$set_button_ast(vd_idp,wd_idp,priorityrd2,,keybuf,0.0,2.0,5.0,3.0)
      call uis$set_button_ast(vd_idp,wd_idp,priorityrd3,,keybuf,0.0,1.0,5.0,2.0)
      call uis$set_button_ast(vd_idp,wd_idp,priorityrd4,,keybuf,0.0,0.0,5.0,1.0)
      call uis$set_button_ast(vd_idp,wd_idp,stptxt,%ref(%loc(wd_idp)),
     2 keybuf,4.0,0.0,4.5,0.5)
      return
      end

       SUBROUTINE priorityrd1( )
C      Road with priority 1
       implicit integer (a-z)
       include 'sys$library:uisentry'
       include 'sys$library:uisusrdef'
       integer dswitch,prn,cnt
      common /cb1/vd_id1,wd_id1,vd_id2,wd_id2,vd_id5,wd_id5,keybuf
      external entdbtnrd

      if (keybuf .eq. uis$c_pointer_button_1) then
      prn = 1
      call entdbtnrd
      end if
```

```
      return
      end

      SUBROUTINE entdbtnrd()
C     Sort out priorities and necessary parameters for transferring controls
C     to knowledge base  for searching
      implicit  integer (a-z)
      include  'sys$library:uisentry'
      include  'sys$library:uisusrdef'
      include ops$library:opsdef.for'
      integer dswitch.prn.cnt
      character*10 name1.name2.cname1
      common  /cb1/vd_id1.wd_id1.vd_id2.wd_id2.vd_id5.wd_id5.keybuf
      external  opsdbtn

       if (prn .eq.  1)  then
       cname1  =  'ALROAD'
       else if  (prn  .eq.  2 )  then
       cname1  =  'ALROADM'
       else if  (prn  .eq.  3 )  then
       cname1  = 'ALROADN'
       else if  (prn  .eq.  4)  then
       cname1  =  'ALROADO'
       end if
       name1  =  '^RD_NAME'
       name2  =  'NAME'
       call  opsdbtn(cname1.name1.name2)
       return
       end


C     Program fragment that activates the OPS5 production system after making necessary
C     data type conversion suitable for OPS5.

       SUBROUTINE  opsdbtn(cname1.name1.name2)
       implicit  integer (a-z)
       include  'sys$library:uisentry'
       include  'sys$library:uisusrdef'
       include ops$library:opsdef.for'
       integer  wmc_cname.wme_name1.wme_name2.length.ln.i
       integer narayn(30)
       character*10 came1.name1.name2
       character*40  laray(30)
       character*2  naray(30)
       logical*1 keybuf1(4)
       logical  completed
       common  /cb1/vd_id1.wd_id1.vd_id2.wd_id2.vd_id5.wd_id5.keybuf
       common  /wd2/vd_20.wd_id20.vd_id21.wd_id21
       common  /kb/kd_id.vcm_id1
       common  /os/ovlswitch
       common  /lc/count
       common  /la/laray.naray.narayn
       common  /shm/vd_idd.wd_idd.cindex
       external  overlay.legend
```

```
cindex = cindex + 1
count = 1
do 1 = 1,20
laray (i) = '
naray(i) =
narayn(i) = 0
end do
call ops$initilize()
call ops$clear
ln = length(cname1)
call ops$reset()
me_cname = ops$intern(%ref(cname1), %val(ln))
call ops$tab (%val(ops$cvna(%val (1))))
call ops$value (%val(wme_cname))
ln = length(name1)
wme_name1 = ops$intern(%ref(name1), %val(ln))
call ops$tab (%val(ops$cvna(%val(2))))
call ops$value (%val(wme_name1))
ln = length(name2)
wme_name2 = ops$intern(%ref(name2), %val(ln))
call ops$tab (%val(ops$cvna(%val(2))))
call ops$value (%val(wme_name2))
call ops$assert()
call ops$startup
call ops$run
call legend
call overlay
return
end
```

C. 6. Creating A Control Element in the Working Memory

The control element in the production **find-roads-rivers-rail,** used for the query shown in 7 1 Plate 1, may be constructed as explained below. The control element in this production is (alrdrr1 ^rrlname name). Let the three strings be assigned to three string variables as in the following

```
cname1 = 'alrdrr1'
name1  = '^rrlname'
name2  = 'name'
call ops$reset( )
```

This action clears the buffer called the *result element* which is used by VAX OPS5 run-time system to place the atoms before it is finally transferred to the working memory

```
ln = length (cname1)
wme-cname = ops$intern(%ref(cname1), %val(ln))
call ops$tab(%val(OPS$cvna(%val(1))))
call ops$value(%val(wme-cname))
ln = length(name1)
wme-name1 = ops$intern(%ref(name1),%val(ln))
call ops$tab(%val(ops$cvna(%val(2))))
call ops$value(%val(wme-name1))
ln = length (name2)
wme-name2 = ops$intern(%ref(name2),%val(ln))
call ops$tab(%val(ops$cvna(%val(3))))
call ops$value(%val(wme-name2))
call ops$assert ( )
```

In the above, wme-cname, wme-name1, and wme-name2 are integer variables. Another integer variable ln holds the length of a string returned by the function **length**. The support routine **ops$intern** translates the character strings such as cname1,name1,and name2 into symbols. The arguments to the function is the address of the string and the length of the string.

Another support routine, **ops$cvna** converts an integer to an integer atom. The arguments for this routine such as integer values 1, 2, 3, *et cetra,* indicate the corresponding field number in the result element into which the atom is to be placed.

The routine **ops$tab** specifies the field in the result element in which the next entry is to be placed.

The role of **ops$value** routine is to place an atom in the result element.

Finally, **ops$assert** copies the contents of the result element into the working memory

C. 7. Data Type Conversion Procedure While Invoking Within the OPS5 Environment, an External Procedure Written in a Language Different From OPS5

Consider the procedure call in the production **find-tourist-places-routes**, shown in 7.2. Plate 2.

```
(call clgplotall  <name1>     <location>
                  <x-cord1>   <y-cord1>
                  <x-cord2>   <y-cord2>)
```

The arguments in the procedure call *viz.*, name1, location, x-cord1, y-cord1, x-cord2, and y-cord2, are all symbolic atoms.

In the procedure clgplotall following type declararations have bneen made.

ename, and lname	*character* type variables of length 40
ename1, lname1, ex-cord1, lx-cord1, ey-cord1, and ly-cord1	*integer* type variables
ex-cord, lx-cord, ey-cord, and ly-cord	*real* type variables

Now we have

```
c = ops$parametercount( )
ename1 = ops$parameter (%val(1))
call ops$pname (%val(ename1), %ref(ename), %val(40))
lname1 = ops$parameter (%val(2))
call ops$pname (%val(lname1), %ref(lname), %val(40))
ex-cord1 = ops$parameter( %val(3))
if (ops$integer (%val(ex-cord1)))
        then ex-cord = ops$cvan(%val(ex-cord1))
else if (ops$floating (%val(ex-cord1)))
                then    ex-cord = ops$cvaf (%val(ex-cord1))
                        ey-cord1 = ops$parameter (%val(4))
```

```
if (ops$integer (%val(ey-cord1)))
        then ey-cord = ops$cvan(%val(ey-cord1))
else if (ops$floating (%val(ey-cord1)))
                then ey-cord = ops$cvaf (%val(ey-cord1))
```

This conversion process should be applied to other arguments also.

The **ops$parametercount** routine returns an integer that indicates the number of argument values stored in the result element. And the **ops$parameter** routine retrieves an argument value stored in the result element. This routine is to be specified with an integer that indicates the field from which the argument value is to be retrieved.

The **ops$pname** now translates the symbolic atom stored in the variable ename1 into a character string and copies it to the buffer ename. This routine is to be specified with three arguments, *viz.*, the symbolic atoms to be translated, the address of the buffer to which the string is to be copied, and the integer that represents the number of characters in the string.

The **ops$integer** routine tests whether or not an atom is an integer, and returns a boolean result. Similarly, **ops$floating** test whether or not an atom is a floating point number and returns a boolean result.

The routine **ops$cvan** converts an integer atom into an integer and returns the result. Similarly, **ops$cvaf** takes a floating point argument and converts it into a floating point number

Appendix D

D. 1. Definition of the Production **find-all-colleges-NH47**

(*vide* 7 5. Plate 5)

```
(p find-all-colleges-NH47
        {<edn> (alclg   ^name  name1)}
               (road    ^rd-name        |national-highway-47|
                        ^rd-cord        <rd-cord>)
        {<clgs> (edn-research
                               ^type            college
                               ^name            <name>
                               ^location        <location>
                               ^x-cord          <x-cord1>
                               ^y-cord          <y-cord1>
                               ^clg-no          <clgno>
                               ^criteria  (adjacent  <rd-cord>))}
-->
        (call clgplotall       <name>           <x-cord1>
                                                <y-cord1>  <clgno>))
```

D. 2. Definition of the Production **find-all-hospital**

(*vide* 7.6. Plate 6)

```
(p find-all-hospital
        {<alhosp>      (ahosp       ^name          <name1>)}
        (hospital      ^name        <name>
                       ^location    <location>
                       ^x-cord      <x-cord1>
                       ^y-cord      <y-cord1>
                       ^h-no        <hno>)
        (place         ^pname       <location>
                       ^x-cord      <x-cord2>
                       ^y-cord      <y-cord2>)
-->
        (call clgplotall <name>     <x-cord1>
                                    <y-cord1>      <hno>))
```

D. 3. Definition of the Production **find-schools-around-secretariat** (vide 7.7 Plate 7)

```
(p find-schools-around -secretariat
        (schlsec              ^name          <name1>)
                (office-building       ^type          government
                                       ^name          secretariat
                                       ^x-cord        <x-cord1>
                                       ^y-cord        <y-cord1>)
                (edn-research          ^type          school
                                       ^name          <name>
                                       ^location      <location>
                                       ^x-cord        <x-cord2>
                                       ^y-cord        <y-cord2>
                                       ^schl-no       <schlno>
                                       ^criteria  (between <2>  <6>
                                                      <secretariat>))
                (place                 ^pname         <location>
                                       ^x-cord        <x-cord3>
                                       ^y-cord        <y-cord3>)
-->
                (call  clgplot         <name>         <x-cord2>
                                       <y-cord2>      <schlno>))
```

References

Short Citation	Cited on (Page No.)	Long Citation
[Abdelmoty *et al.* 1993]	56	**Abdelmoty, A.I., M.H. Williams, and N.W. Paton,** "Deduction and Deductive Databases for Geographic Data Handling," in *Advances in Spatial Databases*. Lecture Notes in Computer Science No. 692. Eds. D. Abel and B.C. Ooi. Berlin, Germany: Springer-Verlag, pp. 443-464.
[Abel 1989]	62	**Abel, D.J.,** "SIRO-DBMS: A Database Toolkit for Geographical Information Systems," *Int. J. Geographical Info. Sys.* vol. 3(2), pp. 103-116.
[Adam *et al.* 1997]	27,28	**Adam, N., B. Awerbuch, J. Slonim, P. Wegner, and Y. Yesha,** "Globalizing Business, Education, Culture Through the Internet," *ACM Communications*, vol. 40(2), pp. 115-121.
[Amidon 1964]	7,8	**Amidon, E. L. ,** "A Computer Oriented System for Assembling and Displaying Land Management Information," *U.S. Forest Service Research Paper, PSW-17,* Berkeley, CA, USA. U. S. Forest & Range Experiment Station.
[Anthony 1991]	12	**Anthony Gar-on Yeh,** "The Development and Applications of Geographic Information Systems for Urban and Regional Planning in the Developing Countries," *Int. J. Geographical Info. Sys.*, vol. 5, pp. 5-27
[Barnsley *et al.* 1993]	12	**Barnsley, M. J., S. L. Barr, A. Hamid, D. A. L. Muller, G. J. Sadler, and J. W. Shepherd,** "Analytical Tools to Monitor Urban Areas," in *Geographical Information Handling — Research and Applications,* Ed. P M. Mather, Chichester, UK: John Wiley, pp. 147-184.
[Benbasat *et al.* 1986]	27,28	**Benbasat, I., A.S. Dexter, and P. Todd,** "An Experimental Program Investigating Color Enhancement and Graphical Information Presentation: An Integration of the Findings," *ACM Communications,* vol. 29(11), pp. 1094-1105.
[Bentley 1975]	52	**Bentley, J. L.,** "Multidimensional Binary Search Trees Used for Associative Searching," *ACM Communications,* vol. 18(9), pp. 509-517
[Berry 1987]	11	**Berry, J. K.,** "Fundamental Operations in Computer-Assisted Map Analysis," *Int. J. Geographical Info. Sys.*, vol. 1, pp. 119-136.
[Brotchic *et al.* 1980]	9	**Brotchic, J. F., J.W. Dicky, and R. Sharpe,** "*TOPAZ*," *Lecture Notes in Economics and Mathematical Systems,* vol. 160, Berlin, Germany: Springer-Verlag.

Short Citation	Cited on (Page No.)	Long Citation
[Brownston *et al.* 1985]	100,101	**Brownston, L., R. Farrel, E. Kant, and N. Martin,** *Programming Expert Systems in OPS5 - An Introduction to Rule-Based Programming,* , Reading, MA, USA: Addison-Wesley.
[Burrough 1986]	35	**Burrough, P.A.,** *Principles of Geographical Information System for Land Resources Assessment,* Oxford, UK: Clarendon Press.
[Butterworth *et al.* 1991]	68	**Butterworth, P., A. Otis, and J. Stein,** "The GemStone Database Management System," *ACM Communications,* vol. 34(10), pp. 64-77.
[Cattell 1991]	66	**Cattell, R.G.G.,** "Next-Generation Database Systems: Introduction to Special Section," *ACM Communications,* vol. 34(10), pp. 30-33.
[Chou 1993]	24	**Chou, Y.,** "Critical Issues in the Evaluation of Spatial Autocorrelation," in *Spatial Information Theory: A Theoretical Basis for GIS,* Lecture Notes in Computer Science No. 716, Eds. A.U. Frank & I. Campari, Berlin, Germany: Springer-Verlag, pp. 421-433.
[Chrisman 1997]	2, 39	**Chrisman, N.,** *Exploring Geographic Information Systems,* New York, USA. John Wiley.
[Clementini & Felice 1995]	56	**Clementini, E., P.D. Felice,** "A Comparison of Methods for Representing Topological Relationships," *Information Sciences,* vol. 3(3), pp. 149-178.
[Clementini *et al.* 1995]	56	**Clementini, E., P.D. Felice, and G. Califano,** "Composite Regions in Topological Queries," *Information Systems,* vol. 20(7), pp. 579-594.
[Cobb *et al.* 1998]	82	**Cobb, M.A., H. Foley III, R. Wilson,** and 2 others, "An OO database Migrates to the Web," *IEEE Software,* vol. 15(3), pp. 22-30.
[Coetzee & Bishop 1998]	81	**Coetzee, S. and J. Bishop,** "A New Way to Query GISs on the Web," *IEEE Software,* vol. 15(3), pp. 31-40.
[Coppock & Rhind 1991]	14	**Coppock, J.T., and D.W. Rhind,** "The History of GIS," in *Geographical Information Systems: Principles and Applications,* vol. 1, Eds. D. J. Maguire, M. F Goodchild, and D. W. Rhind, London, UK:Longman Scientific and Technical, pp. 21-43.
[Cowen & Shirley 1991]	74	**Cowen, D. J. and W. L. Shirley,** "Integrated Planning Information System," in *Geographical Information Systems: Principles and Applications,* Eds. D. J. Maguire, M. F. Goodchild, and D. W. Rhind, London, UK: Longman Scientific and Technical, pp. 297-310.
[Cowen & Ehler 1994]	74	**Cowen, D. J., and G. B. Ehler,** "Incorporating Multiple Sources of Knowledge into Spatial Decision Support System," in *Advances in GIS Research,* vol.1, Eds. T. C. Waugh and R. G. Healey, London, UK: Taylor & Francis, pp. 60-72.

Short Citation	Cited on (Page No.)	Long Citation
[Crossland *et al.* 1993]	28	**Crossland, M.D., J.N. Scudder, R.L. Herschel, and B.E. Wynne,** "Measuring the Relationships of Tasks and Cognitive Style Factors and Their Effects on Individual Decision-Making Effectiveness Using a Geographic Information System," *Proc. Hawaii Int. Conf. on System Sciences,* vol.IV, Los Alamitos, CA, USA: IEEE Computer Society Press, pp. 575-584.
[Cui *et al.* 1993]	56	**Cui, Z., A.G. Cohn, and D.A. Randell,** "Qualitative and Topological Relationship in Spatial Databases," in *Advances in Spatial Databases.* Lecture Notes in Computer Science, No. 692, Eds. D.J. Abel and B.C. Ooi, Berlin: Springer-Verlag, pp. 296-315.
[David *et al.* 1993]	70	**David, B., L. Raynal, G. Schorter, and V. Mansart,** "GeO_2: Why Objects in Geographical DBMS?" in *Advances in Spatial Databases.* Lecture Notes in Computer Science No. 692, Eds. D. Abel and B.C. Ooi. Berlin, Germany: Springer-Verlag, pp. 264-276.
[Densham 1991]	17	**Densham, P.J.,** "Spatial Decision Support Systems," in *Geographical Information Systems: Principles and Applications,* vol. 1, Eds. D. J. Maguire, M. F Goodchild, and D. W Rhind, London, UK:Longman Scientific and Technical, pp. 403-412.
[Deux *et al.* 1990]	68	**Altaïr Consortium,** "The Story of O_2, *IEEE Trans. Knowledge and Data Engg.* vol. 2 (1), pp. 91-108.
[Deux *et al.* 1991]	68	**O_2 Technology,** "The O_2 System," *ACM Communications* vol. 34 (10), pp. 35-48.
[Digital 1988]	102	**Digital Equipment Corporation,** *VAX OPS5 Reference Manual,* Version 2.2 (1988), Maynard, MA, USA:Digital Equipment Corporation.
[Dodson & Haines-Young 1993]	21	**Dodson, A.H., and R.H. Haines-Young,** "Datum Transformations and Data Integration in a GIS Environment," in *Geographical Information Handling — Research and Applications.* Ed. P.M. Mather, Chichester, UK: John Wiley, pp. 79-90.
[Douglas & Peucker 1973]	39	**Douglas, D.H., and T.K. Peucker,** "Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature," *Canadian Cartographer* vol. 10(4), pp. 110-122.
[Egenhofer *et al.* 1989]	55	**Egenhofer, M.J., A. Frank, and J. Jackson,** "A Topological Data Model for Spatial Databases," in *Design and Implementation of Large Spatial Databases.* Lecture Notes in Computer Science, No. 409, Eds. A. Buchmann, O. Günther, T.R. Smith, and Y.F. Wang, Berlin: Springer-Verlag, pp. 271-286.
[Egenhofer 1993]	56	**Egenhofer, M.J.,** "A Model for Detailed Binary Topological Relationships," *Geomatica,* vol. 47(3-4), pp. 261-273.

Short Citation	Cited on (Page No.)	Long Citation
[Ervin 1993]	22	**Ervin, S.M.,** "Landscape Visualization with Emaps," *IEEE Computer Graphics & Applications,*vol. 13(2), pp. 28-33.
[ESRI 1994]	60	**Environmental Systems Research Institute,** *ARC/INFO Data Management: Concepts, data models, database.* Redlands, CA, USA: Environmental Systems Research Institute, Inc.
[Feigenbaum 1977]	85	**Feigenbaum, E.A.,** "The Art of Artificial Intelligence: Themes and Case Studies in Knowledge Engineering," *Proc. IJCAI,* Menlo Park, 1977, pp. 1014 -1029.
[Finkel & Bentley 1974]	51	**Finkel, R.A., and J.L. Bentley,** "Quad trees: A Data Structure for Retrieval on Composite Keys," *Acta Informatica,* vol. 4 (1), pp. 1-9.
[Floriani *et al.* 1993]	55	**Floriani, L.D., P. Marzano, and E. Puppo,** "Spatial Queries and Data Models," in *Spatial Information Theory: A Theoretical Basis for GIS.* Lecture Notes in Computer Science No. 716, Eds. A.U. Frank and I. Campari. Berlin, Germany: Springer-Verlag, pp. 113-138.
[Forgy 1982]	103	**Forgy, C.L.,** "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem," *Artificial Intelligence,* vol. 19, pp. 17-37
[Fraser 1994]	31	"Cartographic Visualisation and and Spatial Data Handling." in *Advances in GIS Research,* vol.1, Eds. T. C. Waugh and R. G. Healey, London, UK: Taylor & Francis, pp. 16-28.
[Gardels 1994]	111	**Gardels, K.,** "Sequoia 2000 and Geographic Information: The Guernewood Geoprocessor," in *Advances in GIS Research* vol. 2, Eds.T.C. Waugh and R.G. Healey, London, UK. Taylor & Francis, pp. 1072-1085.
[Gentner & Grudin 1996]	27	**Gentner, D.R., and J. Grudin,** "Design Models for Computer-Human Interfaces," *IEEE Computer,* vol. 29(6), pp. 28-35.
[Gevarter 1987]	106	**Gevarter, W.B.,** "The Nature and Evaluation of Commercial Expert System Building Tools," *IEEE Computer,* vol. 20(5), pp. 24-41.
[Goodchild 1992]	14	**Goodchild, M.F.,** "Geographical Information Science," *Int. J. Geographical Info. Sys.,* vol. 6(1), pp. 31-45.
[Goodchild *et al.* 1992]	24	**Goodchild, M.F., R. Haining, S. Wise,** and 12 others, "Integrating GIS and Spatial Data Analysis: Problems and Possibilities," *Int. J. Geographical Info. Sys.,* vol. 6(5), pp. 407- 423.
[Green 1964]	7	**Green, R.,** "The Storage and Retrieval of Data for Water Quality Control," *Public Health Service Publication No. 1236,* Washington, DC: U. S. Department of Health, Education and Welfare.
[Grosky 1997]	27,79	**Grosky, W.I.,** "Managing Multimedia," *ACM Communications,* vol. 40(12), pp. 73-80.

Short Citation	Cited on (Page No.)	Long Citation
[Guida & Mauri 1993]	23	**Guida, G. and G. Mauri,** "Evaluating Performance and Quality of Knowledge-Based Systems: Foundation and Methodology," *IEEE Trans. Knowledge and Data Engg.*, vol. 5(2), pp. 204-224.
[Günther & Lamberts 1994]	71	**Günther, O. and J. Lamberts,** "Object-Oriented Techniques for the Management of Geographic and Environmental Data," *The Computer Journal,* vol. 37 (1), pp. 16-25.
[Guptill 1994]	10,20	**Guptill, S.C.,** "Synchronization of Discrete Geospatial Data Bases," in *Advances in GIS Research* vol. 2, Eds.T.C. Waugh and R.G. Healey, London, UK: Taylor & Francis, pp. 945-956.
[Haas *et al.* 1990]	68	**Haas, L.M., W. Chang,** and 8 others, "Starburst Mid-Flight: As the Dust Clears," *IEEE Trans. Knowledge and Data Engg.*, vol. 2 (1), pp. 143-160.
[Hamre 1994]	22	**Hamre, T.,** "An Object-Oriented Conceptual Model for Measured and Derived Data Varying in 3D Space and Time, in *Advances in GIS Research* vol. 2, Eds.T.C. Waugh and R.G. Healey, London, UK: Taylor & Francis, pp. 868-881.
[Harmon 1995]	32	**Harmon, P.,** "Object-Oriented AI: A Commercial Perspective," *ACM Communications,* vol. 38 (11), pp. 80-86.
[Hebbar 1991]	30	**Hebbar, K.J.,** "Multi-Spectral Classification Using Artificial Neural Networks and Knowledge-Based Geographic Information System," in *Artificial Intelligence and Expert System Technologies in The Indian Context,* Eds. V V.S. Sharma, N. Viswanadham, B. Yegnanarayana, and B.L. Deekshatalu, New Delhi, India: Tata McGraw-Hill, pp. 115-120.
[IEEE Computer Society 1996]	82	**IEEE Computer Society,** "New Standard Could Put GIS on the Map," *IEEE Computer,* vol. 29(9), pp. 20-21.
[Isakowitz *et al.* 1995]	78	**Isakowitz, T., E.A. Stohr, and P. Balasubramanian,** "RMM: A Methodology for Structured Hypermedia Design," *ACM Communications,* vol. 38(8), pp. 34-44.
[Jayaprakash & Menon 1995]	33, 117	**Jayaprakash, P. and A. K. Menon,** "Development of a Geographic Information System for Cities," in *Proc. Seventh Kerala Sci. Congress,* Palghat, India, pp. 259-261.
[Johnson 1997]	79	**Johnson, R.E.,** "Frameworks = (Components + Patterns)," *ACM Communications,* vol. 40(10), pp. 39-42.
[Jones & Luo 1994]	22	**Jones, C.B., and L.Q. Luo,** "Hierarchies and Objects in Deductive Spatial Database," in *Advances in GIS Research* vol. 1, Eds.T.C. Waugh and R.G. Healey, London, UK: Taylor & Francis, pp. 588-603.

Short Citation	Cited on (Page No.)	Long Citation
[Kemp 1995]	4,26	**Kemp, Z.,** "Multimedia and Spatial Information Systems," *IEEE Multimedia*, vol. 2 (4), pp. 68-76.
[Krack 1996]	1,26, 78	**Krack, M. J.,** "Integrating Multimedia in Geographical Information Systems," *IEEE Multimedia*, vol. 3 (2), pp. 59-65.
[Knuth 1975]	51	**Knuth, D.E.,** *The Art of Computer Programming*, vol. 1, Fundamental Algorithms, 2nd Ed. Reading, MA, USA: Addison-Wesley.
[Kochevar & Wanger 1995]	109	**Kochevar, P.D. and L.R. Wanger,** "Tecate: A Software Platform for Browsing and Visualizing Data from Networked Data Sources," *Digital Technical Journal*, vol. 7(3), pp. 66-83.
[Ku *et al.* 1994]	22	**Ku, C.S., H.D. Kim, and L.S. Henschen,** "An Efficient Indefiniteness Inference Scheme in Indefinite Deductive Databases," *IEEE Trans. Knowledge and Data Engg.*, vol. 6(5), pp. 713-722.
[Lamb *et al.* 1991]	68	**Lamb, C., J. Landis, J. Orenstein, and D. Weinreb,** "The ObjectStore Database System," *ACM Communications*, vol. 34 (10), pp. 50-63.
[Lapalme *et al.* 1992]	16	**Lapalme, G., J. Rousseau, S. Chapleau,** and 3 others, "GeoRoute: A Geographic Information System for Transportation Applications," *ACM Communications*, vol. 35(1), pp. 80-94.
[Larson *et al.* 1995]	109	**Larson, R.R., C. Plaunt, A.G. Woodruff, and M. Hearst,** "The Sequoia 2000 Electronic Repository," *Digital Technical Journal*, vol. 7(3), pp. 50-65.
[Larue *et al.* 1993]	60	**Larue, T., D. Pastre, and Y. Viémont,** "Strong Integration of Spatial Domains and Operators in a Relational Database System," in *Advances in Spatial Databases*. Lecture Notes in Computer Science No. 692. Eds. D. Abel and B.C. Ooi. Berlin, Germany: Springer-Verlag, pp. 53-72.
[Laurini & Thompson 1994]	7,20,35, 70,80	**Laurini, R. and D. Thompson,** *Fundamentals of Spatial Information System*, London, UK: Academic Press.
[Little & Ghafoor 1991]	78	**Little, T.D.C., and A. Ghafoor,** "Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks," *IEEE Computer*, vol. 24(10), pp. 42-50.
[Lohman *et al.* 1991]	19,63, 68	**Lohman, G., B. Lindsay, H. Pirahesh, and K.B. Schiefer,** "Extensions to Starburst: Objects, Types, Functions, and Rules," *ACM Communications*, vol. 34 (10), pp. 94-109.
[Luger & Stubblefield 1989]	93	**Luger, G.F., and W.A. Stubblefield,** *Artificial Intelligence and the Design of Expert Systems*, Redwood City, CA, USA, Benjamin/Cummings.

Short Citation	Cited on (Page No.)	Long Citation
[Pasquale *et al.* 1995]	109	**Pasquale, J., E.W. Anderson, K. Fall, and J.S. Kay,** "High-Performance I/O and Networking Software in Sequoia 2000," *Digital Technical Journal,* vol. 7(3), pp. 84-96.
[Pequet & Wentz 1994]	25	**Pequet, D., and E. Wentz,** "An Approach for Time-Based Analysis of Spatiotemporal Data," in in *Advances in GIS Research,* vol.1, Eds. T C. Waugh and R. G. Healey, London, UK: Taylor & Francis, pp. 489-504.
[Petrakis & Orphanoudakis 1993]	27	**Petrakis, E.G.M., and S.C. Orphanoudakis,** "Methodology for the Representation, Indexing, and Retrieval of Images by Content," *Images and Vision Computing,* vol. 11, pp. 504-520.
[Posnak *et al.* 1997]	79	**Posnak, E.J., R.G. Lavender, and H.M. Vin,** "An Adaptive Framework for Developing Multimedia Software Components," *ACM Communications,* vol. 40(10), pp. 43-47
[Rafaat *et al.* 1991]	25	**Rafaat, H.M., Q. Xiao, and D.A. Gauthier,** "An Extended Relational Database for Remotely Sensed Image Data Management Within GIS," *IEEE Trans. Geoscience and Remote Sensing,* vol. 29(4), pp. 651-655.
[Ramachandran *et al.* 1994]	22,25	**Ramachandran, B., F. MacLeod, and S. Dowers,** "Modelling Temporal Changes in a GIS Using an Object-Oriented Approach," in *Advances in GIS Research* vol. 1, Eds.T.C. Waugh and R.G. Healey, London, UK: Taylor & Francis, pp. 518-537
[Raper & Bundock 1993]	108	**Raper, J. and M. Bundock,** "Development of a Generic Spatial Language Interface for GIS," in *Geographical Information Handling —Research and Applications.* Ed. P. M. Mather, Chichester, UK. John Wiley, pp. 113-143.
[Rector 1993]	15	**Rector, J.M.,** "Utilities," in *Profiting from a Geographic Information System,* Ed. G.H. Castle, Fort Collins, CO, USA. GIS World, Inc., pp. 193-208.
[Roman 1990]	23	**Roman, G.,** "Formal Specification of Geographic Data Processing Requirements," *IEEE Trans. Knowledge and Data Engg.,* vol. 2(4), pp. 370-380.
[Samet 1984]	51	**Samet, H.,** "The Quadtree and Related Hierarchical Data Structures," *ACM Computing Surveys,* vol. 16 (2) pp. 187-260.
[Schalkoff 1990]	94	**Schalkoff, R. J.,** *Artificial Intelligence: An Engineering Approach,* Singapore: McGraw-Hill.
[Schek & Wolf 1993]	15	**Schek, H. and A. Wolf,** "From Extensible Databases to Interoperability between Multiple Databases and GIS Applications," in *Advances in Spatial Databases,* Lecture Notes in Computer Science No. 692Eds. D. Abel and B.C. Ooi, , Berlin, Germany: Springer-Verlag. pp. 207-238.

Short Citation	Cited on (Page No.)	Long Citation
[Sellis *et al.* 1993]	76	**Sellis, T., C.-C. Lin, L. Raschid,** "Coupling Production Systems and Database Systems: A Homogeneous Approach," *IEEE Trans. Knowledge and Data Engg.*, vol. 5(2), pp. 240-256.
[Sendler 1996]	22	**Sendler, U.,** *CAD & Office Integration: OLE for Design and Modelling—A New Technology for CA Software,* Berlin, Germany: Springer-Verlag.
[Shaw *et al.* 1996]	22	**Shaw, K., M. Cobb, M. Chung, and D. Arthur,** "Managing the US Navy's First OO Digital Mapping Project," *IEEE Computer*, vol. 29(9), pp. 69-74.
[Shekhar *et al.* 1997]	22	**Shekhar, S., M. Coyle, B. Goyal, D. Liu, and S. Sarkar,** "Data Models in Geographic Information Systems," *ACM Communications,* vol. 40(4), pp. 103-111.
[Shepherd 1994]	27	**Shepherd, I.D.H.,** "Multi-Sensory GIS: Mapping Out the Research Frontier," in *Advances in GIS Research,* vol.1, Eds. T C. Waugh and R. G. Healey, London, UK: Taylor & Francis, pp. 356-390.
[Smith *et al.* 1987]	22,106, 115	**Smith, T. R., S. Menon, J. L. Star, and J.E. Estes,** "Requirements and Principles for the Implementation and Construction of Large Scale Geographic Information Systems," *Int. J. Geographical Info. Sys.* vol. 1 (1), pp. 13-31.
[Stonebraker 1989]	72,77	**Stonebraker, M.,** "Future Trends in Database Systems," *IEEE Trans. Knowledge and Data Engg.*, vol.1 (1), pp. 33-44.
[Stonebraker et al. 1990]	20,63, 68	**Stonebraker, M., L.A. Rowe, and M. Hirohama,** "The Implementation of POSTGRES," *IEEE Trans. Knowledge and Data Engg.*, vol. 2 (1), pp. 125-142.
[Stonebraker & Kemnitz 1991]	58,68	**Stonebraker, M., and G. Kemnitz,** "The POSTGRES Next-Generation Database Management System," *ACM Communications,* vol. 34 (10), pp. 78-92.
[Stonebraker 1995]	20,63, 76,109	"An Overview of the Sequoia 2000 Project," *Digital Technical Journal*, vol. 7 (3), pp. 39-49.
[Traynor & Williams 1997]	28, 162	**Traynor, C., and M.G. Williams,** "A Study of End-User Programming for Geographic Information Systems," in *Empirical Studies of Programmers: Seventh Workshop*, Alexandria, VA. New York, NY,USA: ACM Press, pp. 1-17.
[Turk 1993]	27,28	**Turk, A.,** "The Relevance of Human Factors to Geographical Information Systems," in *Human Factors in Geographical Information Systems.* Eds. D. Medyckyj-Scott and H.M. Hearnshaw, London, UK: Belhaven Press, pp. 15-31.

Short Citation	Cited on (Page No.)	Long Citation
[Vetter *et al.* 1995]	27	**Vetter, R., C. Ward, and S. Shapiro,** "Using Color and Text in Multimedia Projections," *IEEE Multimedia,* vol. 2(4), pp. 46-54.
[Wadge *et al.* 1993]	13	**Wadge, G., A. Wislocki, and J. Pearson,** "Mapping Natural Hazards With Spatial Modelling Systems," in *Geographical Information Handling — Research and Applications.* Ed. P M. Mather, Chichester, UK: John Wiley, pp. 239-250
[Waugh & Healey 1987]	62	**Waugh, T.C., and R.G. Healey,** "The GEOVIEW Design. A Relational Data Base Approach to Geographical Data Handling," *Int. J. Geographical Info. Sys.* vol. 1(2), pp. 101-118.
[White 1991]	15	**White, M.,** "Car Navigation Systems," in *Geographical Information Systems: Principles and Applications,* vol. 2, Eds. D. J. Maguire, M. F. Goodchild, and D. W Rhind, London, UK:Longman Scientific and Technical, pp. 115-125.
[Widom 1996]	20,63, 68	**Widom, J.,** "The Starburst Active Database Rule System," *IEEE Trans. Knowledge and Data Engg.,* vol. 8(4), pp. 583-595.
[Wiil 1995]	78	**Wiil, U.K.,** "Hyperform: Rapid Prototyping of Hypermedia Services," *ACM Communications,* vol. 38(8), pp. 109-111.
[Worboys *et al.* 1990]	22	**Worboys, M.F., H.M. Hearnshaw, and D.J. Maguire,** "Object-Oriented Modelling for Spatial Databases," *Int. J. Geographical Info. Sys.,* vol. 4(4), pp. 369-383.
[Worboys 1992]	55	**Worboys, M.F.,** "A Generic Model for Planar Geographical Objects," *Int. J. Geographical Info. Sys.,* vol. 6(5), 353-372.
[Worboys *et al.* 1993]	22	**Worboys, M.F., K.T. Mason, and B.R.P. Dawson,** "The Object-Based Paradigm for a Geographical Database System: Modelling, Design, and Implementation Issues," in *Geographical Information Handling —Research and Applications.* Ed. P M. Mather, Chichester, UK: John Wiley, pp. 91-102.
[Worboys 1994]	25,56	**Worboys, M.F.,** "Unifying the Spatial and Temporal Components of Geographical Information," in *Advances in GIS Research,* vol.1, Eds. T. C. Waugh nd R. G. Healey, London, UK: Taylor & Francis, pp. 505-517
[Wu 1997]	27	**Wu, J.K.,** "Content-Based Indexing of Multimedia Databases," *IEEE Trans. Knowledge and Data Engg.,* vol. 9(6), pp. 978-989.
[Yeo & Yeung 1997]	27	**Yeo, B., and M.M. Yeung,** "Retrieving and Visualizing Video," *ACM Communications,* vol. 40(12), pp. 43-52.

Short Citation	Cited on (Page No.)	Long Citation
[Yoshitaka *et al.* 1994]	27	**Yoshitaka, A., S. Kishida, M. Hirakawa, and T. Ichikawa,** "Knowledge-Assisted Content-Based Retrieval for Multimedia Databases," *IEEE Multimedia*, vol. 1(4), pp. 12-21.
[Zhang *et al.* 1994]	24	**Zhang, A., H. Yu, and S. Huang,** "Bringing Spatial Analysis Techniques Closer to GIS Users: A User-Friendly Integrated Environment for Statistical Analysis of Spatial Data, in *Advances in GIS Research*, vol.1, Eds. T. C. Waugh and R. G. Healey. London, UK. Taylor & Francis, pp. 297-313.

List of Publications by the Author

1. P Jayaprakash and A. K. Menon "Development of a Geographic Information System for Cities," Proceedings Seventh Kerala Science Congress, Palghat, pp. 259-261 (1995).

2. P Jayaprakash and A. K. Menon "Design of a Geographic Information System Using the Production System OPS5," Digest of Technical Papers, Southern Regional Convention, Cochin, Computer Society of India, pp. 121-131 (1995).

3 P Jayaprakash and A. K. Menon "Design of a Geographic Information System Using OPS5," Proceedings IEEE International Geoscience and Remote Sensing Symposium, Lincoln, Nebraska, pp. 408-410 (1996).