NETWORK ANALYST FOR GIS

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Engineering

By

Neha Rathore
(04MCE017

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY
Ahmedabad 382481
May 2006

This is to certify that the Dissertation entitled

Network Analyst for GIS

Presented by

Neha Rathore

has been accepted toward fulfillment of the requirement
for the degree of
Master of technology in Computer Science & Engineering

Professor In Charge **Head of The Department**

Date

ABSTRACT

The Shortest Path Problem is to find the shortest distance (least cost) from a source node to all other nodes or to a subset of nodes on the Network. Dijkstra's algorithm is best suited for finding shortest path on real road networks as it solves the single source shortest path problem on a weighted directed graph for the case in which all the edge weights are non-negative.

Network analysis is one of the cornerstones of GIS functionality. In Geographic Information System, for applying shortest path algorithm on a network knowledge of shape files is required. A network and its segments are normally evaluated based on the distance or time required to travel from one node into another. The best path is, then, defined such that the total time or distance between the two nodes is at a minimum .Applying shortest path algorithm on a network is helpful in solving many problems like , it finds the best path between origin and destination over a given network of bus routes, emergency route in case of disaster, minimum distance to reach a hospital so that decisions about the path to be followed can be taken in case of emergency treatment, Best path and minimum distance to reach an education department, Community Hall ,Bank ,Police Station etc., to determine the shortest path between a defined source and destination.

Keywords: Shortest Path Algorithm, Dijkstra's Algorithm, Geographic Information System.

ACKNOWLEDGEMENT

It gives me great pleasure in expressing my thanks and profound gratitude to **Mr. Shashikant Sharma,** Senior Scientist, **BISAG** (Gandhinagar) for his valuable guidance and continual encouragement throughout the Major project. I am heartily thankful to him for his time to time suggestion and clarifying the concepts of the topic that helped me a lot during this study.

I would like to give my special thanks to **Mr. T.P Singh**, Director, **BISAG** (Gandhinagar) for his continual kind words of encouragement and motivation throughout the Major Project.

I am thankful to **Professor S.N Pradhan** Institute of Technology, **Nirma University**, Ahmedabad, for his suggestions towards the project work..

The friends, who motivated me throughout this course, I am thankful to all of them.

Neha Rathore

Roll No.04MCE017
M-Tech SEM IV

INDEX

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

- Motivation
- Literature Survey
- Existing Systems
- Client and Targeted Users

1.1 MOTIVATION

Network analysis is one of the most powerful tools of GIS. Selecting optimum paths for inter-cities traveling, transportation vehicles of domestic, directing the bus fleet of tourism corporations and international transportation companies are some examples of network analysis applications. So different parts of the routes network are evaluated considering their distance and required time to pass each part to analyze the optimum route to take.

Bhaskaracharya Institute of Space Application & Geo-Informatics (BISAG) is an organization which identifies & satisfies the needs of Gujarat state by facilitating the use of Remote Sensing (RS), Geographic Information Systems (GIS) and Communication Technology. In the near future it will extend its work with Global Positioning System (GPS).

It has a singular purpose to identify & satisfy the needs of Gujarat state by using remote sensing, GIS technology & satellite communication. It has mandate to apply state of the art technology to create the GIS based decision support system for various infrastructure & resource utilities in Gujarat & promote, train & help users to establish their own GIS facility using satellite communication. To do all such activities, BISAG is armed with Scientists & Engineer.

Since 1998 BISAG(Gandhinagar) has been working on the task of creating GIS applications ,which can be distributed independently throughout the various Governmental Departments of the Gujarat State.They all require GIS software for simplifying their work.There are various GIS softwares available in the market like Arc/Info,ArcView,MapInfo etc,but these softwares are very costly.So BISAG decided to develop its own GIS software (Pragati GIS) that can meet all the needs of government.Arc view includes the facility of Network Analysis which has not been included in Pragati GIS and since Network Analysis helps in solving many real world problems I decided to work on it.

1.2 LITERATURE SURVEY

1.2.1 GEOGRAPHIC INFORMATION SYSTEM

Introduction

Geography is the science of spatial relationships. Maps from a major constitute of geography, as they are a means of representing very large spatial relationships in a physically handle-able size. The need for environmentally being and socially accountable development has put a heavy demand on the capability of planners. Therefore planning and execution now require more accurate ,reliable and timely information and better tools for the management of such information. This require not only a variety of maps but a large amount of spatial information, commonly known as statistics, a means to handle this tools to selectively extract information relevant to the planning task. In sort, an information system for geographical data is needed.

Any information system has four major components. There is an input module which accepts data, a data base module which organized and stores the data, an analysis module which selectively retrieves and manipulation the data and an output module which presents the analyzed information.

A Geographic Information System (GIS) is different in the sense that it handles both spatial and non-spatial data, consequently the corresponding module becomes more complex.

Information system incorporates apart from database management function, a set of analysis modules which selectively retrieve and manipulate the data and output module which represents analyzed information in the context of given planning needs.

Input to an Information System is data, which is raw and does not convey any meaning unless analyzed and converted into meaningful information. Output from any Information System is the information, which conveys a meaning and prompts a set of action or decisions.

GIS (Geographic Information System) plays major role by providing linkage between the information domain and the technologies available for management

and development planning. Geographical Information System is a particular form of information System is a set of process, executed on a raw data, to product information, which will useful in decision making. Therefore an information system must have a full
range of function to achieve its purpose, including observation, measurement, description, explanation, forecasting and decision-making.

Functionality of GIS

GIS, basically refers to science and technology dealing with the character and the structure of spatial information, its method of capture, organization, classification, analysis, measurement, display and dissemination as will as the infrastructure necessary for the optimal use of the information. With the increase in volume and dimensionality of data it becomes essential to use automated GIS. Use of an automated system has become necessary as the data are maintain in physically compact format (i.e. magnetic media), data can be retrieved with the greater speed, various computerized tools allow a variety of manipulation. Hence, GIS is defining as:

"AN AUTOMATED TOOL USEFUL TO CAPTURE, STOREGE, RETRIEVAL AND MANIPULATION, DISPLAY AND QUERYING BOTH SPATIAL AND NON-SPATIAL, DATA TO GENERATE VARIOUS PLANNING SCENARIOUS FOR DECISION MAKING."

Alternatively, we could describe "Geographic Information System (GIS) as a system which provide a computerized mechanism for integrating various geo-referenced datasets analyzing them in order to generate information to planning needs in a given context.

Thus GIS is a tool able to answer the location, condition, trends and modeling. For example, the location of feature is answered with the help of geographic reference Latitude and longitude. The second question the converse of the first one and requires spatial analysis to answer. Instead of identifying what exists at a given location, one wants to find location where certain conditions are satisfied. The third one involves both
of the above to answer and seeks to find the difference with an area through time. The fourth one determines what types of spatial pattern exists when the last one

would be "what if…..." question is paused to determine what happens. To answer this integrated to require combining spatial and non-spatial information.

A Geographical Information System (GIS) is a particular form of information system applied to geographical data. It is powerful tool to input, store, and manipulate analysis, model, and output spatial and attribute information. We could describe Geographical Information System as various geo referenced data sets and analyzing in order to generate information relevant to planning needs in a given context. The ultimate goal of a GIS is to support decision-making.

Like any other information System, GIS is also primarily an input output system and includes Database Management System (DBMS) function. However, it differs from conventional DBMS and information systems in the sense that every pieces of data element in GIS has to be directly as co-ordinates with respect to predefined Co-ordinate system.

The data handling in GIS is not limited to only data with explicit coordinate reference. Coordinates could be expressed indirectly, e.g. demography associated with the village and village being expressed in form of polygon or point coordinate. In general GIS facilitates handling of variety of data both spatial as well as non-spatial. By organizing spatial information into from, GIS allows us to manipulate and display geographical knowledge in new and exciting ways.

Requirements of GIS

The environment in which a GIS operates is defined by hardware (the machinery including a hot computer), a digitizer or a scanner for converting the input data, a plotter for presentation processed outputs and video display unit for commanding the system by a user, the software (programs that tell the computer what to do) and the data. In this context GIS can be seen as system of hardware, software, and procedures design to support the capture and management of data for solving, analyzing, modeling and displaying spatially-referenced data for solving complex planning and management problems. Although many other computer programs can use spatial data (e.g. Auto CAD and statistics packages), GIS include the additional ability to perform spatial operations.

Major Component of GIS

The major components of GIS are:

1. The end use or management
2. Data Acquisition
3. Data Input
4. Data Storage and retrieval
5. Analysis
6. Information presentation

There are two components of Geographic data i.e. spatial data and attribute data. On the maps, symbols and text convey descriptive information. The map then becomes a powerful tool for referencing geographic information. The same concept applied to spatial data model. Therefore the powerful capability of GIS lies in the link between the spatial and the tabular data, which illustrate the relationship between and attribute data.

Types of GIS

GIS may be categorize into three basic type:

- Map Based
- Integrated
- Linkage Based

Information Storage

Because graphical information is very different from tabular information, GIS uses to database to graphical and tabular information that must manage: a spatial and an attribute database.

Spatial Database

A spatial database is used to manage graphic information for several reasons. A spatial database makes searches for graphic extremely fast because the spatial database has been optimize to manage graphic information rather than tabular information.

A spatial database can store and manipulate complex entities as discrete objects. For ex., a polygon can be selected by referring a point anywhere on or within the polygon.

A spatial database allow the use of double precision numbers consequently, a spatial database can store very large coordinates values and is able to manage very large amount of graphic information without significant decline in performance.

A spatial database allows entities to be retrieved and manipulates independently of the graphic engine and attribute database.

The spatial database stores two forms of graphic information

Object Geometric

Object relationships (Topology)

Objects

All graphical entities are store in the spatial data objects. Any entity for which you wish to maintain, store and retrieve the geographical location may be considered an object. In GIS, first creating a graphic representation of the object in CAD and then loading this representation of the object into the spatial database create objects.

Object Type

The GIS spatial data is able to handle four types of objects. This object can be categorized as simple objects and complex objects. Simple objects include point and line objects are defined know number of coordinate pairs. Complex objects are defined by undetermined of coordinates pair and include line , string and polygon objects.

- Point Objects
- Line Objects
- String Objects
- Polygon Objects

Connecting Features and Attributes

The importance of GIS lies in its link between the graphic (spatial) and the

tabular (non-spatial or a spatial) data. Basically there are three characteristics of this connection.

They are

- There is one to one relationship between features on digital map and the records in the feature attribute table. The link between the feature and its records is maintained through a unique numerical identifier assigned to each feature (label points).
- The unique identifier is physically in two places i.e. in the file that content X, Y coordinates and with the corresponding records in the feature attribute table. So, once this connection is established one can based on the attributes stored in the feature attribute table.

Data Manipulation

GIS allow a variety of manipulation such as map measurement, map overlay analysis transformation from one coordinate projection to another coordinate projection, graphic design and manipulated simultaneously in related manner.

Apart from this system must also facilities other typical related DBMS function like interactive query language, basic file creation / update file management , basis search retrieval and report generation, database content and high level language interface.

Spatial Database Management in GIS

The scope database in GIS is quite varied as compared conventional DBMS. In the conventional DBMS on the deals with one-dimensional data (non- -spatial) in form of hierarchical, network or relation model. In GIS the data is handled in the form of Geo-Relational model, which ha two components viz., Entity location data in form topological structures handled in conventional file mode and attributes in form of a RDBMS, which is embedded within GIS.

The first component concerned with the graphic data domain, where in the location specific data for various geographic entities are stored, is system specific and a user is not given access for making any modifications in the file structure in

the file structures. In the second domain pertaining to the non-location attributes of the spatial entities, users can attempt variety of modification by adding, removing or changing the specifications of specific set of attribute for the spatial entities. Graphics and non-graphics domains are link through the relation tables. Such an organization gives sample scope for building up an integrated date base involving spatial and non-spatial data elements. This opens up many exciting possibilities with regards to the end use of the database. Looking at the ends of such an integrated database in a perspective, the Geo-relational nature of the data organizations makes it possible to:

- Visualize the GIS database as integrated database providing a common platform for spatial as well as non-spatial data from variety of sources.
- Make integrated queries on the underlying database looking at any combination of data elements together.
- General planning views based on multiple analyses.
- General planning views based on analyses of multiple parameters.
- And simulate "what – if…… Scenarios in an interactive and iterative manner providing flexible decision supports in spatial context approaching towards paperless planning.

Relation between Spatial and non-Spatial Data

With spatial data if proper LINKAGE BETWEEN SPATIAL AND NON-SPATIAL data Non-spatial data are an important part of GIS database. User can use these data along is defined. There linkages and inter relationship are an import element of the GIS database organizations as they define the user relations or user views that can be created.

Spatial and non-Spatial Data Linkage

All the spatial data sets have as associated attribute table where the detail attributes table of each feature is recorded. There are two major linkage aspects involved:

- For all the spatial data sets other than administrative maps, the linkage is achieved through the data dictionary feature code.
- For administrative maps, villages and Taluka maps, the linkage is archived on a one-to one relation based to the on a unique code for each village or the Taluka. This link code also is related to the census village number on a one-to –one basis. Thus the internal organization of the spatial village, Taluka
- boundaries if flexible to relate to the non-spatial data set son a one-to-one basis further, because of the co-relation to census village numbers it is also possible to abstract village data to Taluka data.

There are varying approaches for design of data models and structure as described below.

Two basic types of data models have evolved for storing the spatial data in digital form.

1. **Vector**
2. **Tessellation (grid) models.**

In the vector type of data model, the basic logical unit in a map is line or vector. A series of x-y point's location along the line are recorded as the components of a single data record. The points can be represented as lines of zero length (i.e. one x-y location).

In the grid model the basic logical unit for storage is a single cell or unit of space in the mesh. This class of data models encompasses much more than the approaches based on a rectangular of square grid. These may include May identify repeatable pattern of a regular polygon (two dimensional) or polyhedron (three dimensional) vector data mode also has its variation in the way they describe the entity interrelationships.

Whereas the two classes of spatial data models are in common practice today, there also exists a third type of model –the hybrid type. This class of the data model is a recent development with an attempt to possess the characteristics of both the vector and models.

Since a GIS attempts to model the real world as a set of digital number and features, these models should be amenable to structuring, querying and processing the problems. The way a spatial data processed creates the problems.

The way a spatial data processing algorithm works, is not necessary the way a human operator will solve a problem in the real world. The data structure must therefore account for the quirks of the algorithms. This requirement leads to the adoption of a functional structure based on primitives. The data structure decides the physical and logical storage, retrieval, query and analysis capabilities. It allows for the linking of spatial entitles and spatial attributes. It also impacts the efficiency of storage (space) and access (time). A spatial data is organizing their inter–relationships. In a GIS this is essential. This relationship is the topology.

The oldest and most popular model has been the vector data model that deals which an ordered set of x and y locations representing the points, line and polygons. The ordering takes into account the spatial inter-relationship.

With the advent of remote sensing as well as automatic scanners another structure, which has gained popularly, is the raster structure (one amongst the family of grid models). Other structure have evolved form the need to handle both vector and raster data. One such example is the quid-tree data structure.

Vector data model and associated structures

The vector data model is based on representation of objects by an array of coordinates. As mentioned above, the vector mode of data representation has its variations.

The spaghetti structure

The most simplest and elementary vector data model is a direct line-for-line translation of the paper map. This is called sapphire mode. Associated data structure for this model is very simple and easy to understand. Each geographic entity becomes one logical record in the digital file, and is defined as strings of X-Y coordinates.

This model has severe limitations since there are no spatial relationships explicitly defined amongst various entitles. Moreover, for adjacent polygon data, the model results in recording of X-Y coordinate of the shared boundary segment twice.

The model is very inefficient for most types of spatial data analysis, since any relationship required has to be derived through computation. However the lack of

explicitly stored relationships, which are extraneous to the plotting process, makes this model efficient for reproducing the original graphic input. The model is thus used primarily for applications that are limited to map inputs and outputs.

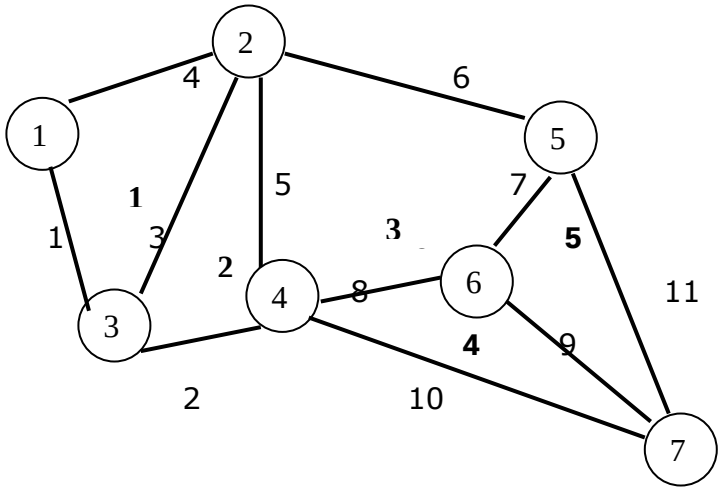

FIGURE:1.1 THE TOPOLOGICAL STRUCTURE

TABLE 1.1 TOPOLOGICALLY CODED NETWORK AND POLYGON FILE

Link	Right Polygon	Left Polygon	Node 1	Node 2
1	1	0	3	1
2	2	0	4	3
3	2	1	3	2
4	1	0	1	2
5	3	2	4	2
6	3	0	2	5
7	3	3	5	6
8	4	3	6	4
9	5	4	7	6
10	4	0	7	4
11	5	0	5	7

Node	X Coordinate	Y Coordinate
1	23	8
2	17	17
3	29	15
4	26	21
5	8	26
6	22	30
7	24	38

The second vector model is the topological model where in the spatial relationships amongst the entities are explicitly recorded. The basic logical entities in this structure are points or nodes, arcs which are lines joining nodes and polygons which are areas completely enclosed by arcs. The attributes or descriptors are lined to each of these entitles.

Such an organization has an advantage over Spaghetti Model for spatial analysis as it minimizes geometric computation overheads for inferring the relationships amongst the spatial object. Moreover, it records the coordinates of the common polygon boundaries only once thereby reducing the redundancy and facilities consistency and integrity of data storage.

GIS-Core of the Database

The Geographic Information System (GIS) package is the core of the data sets, as both spatial and non-spatial databases have to be handled. The GIS

package offers efficient utilities for handling both of these data sets and also allows for the spatial database organization; non-spatial data sets organization-mainly as attributes of the spatial elements; analysis and transformation for obtaining the required information; obtaining information in specific format (cartographic quality outputs and reports), organization of a user friendly query-system. In the present study ARC/INFO GIS package has been used as the core of the spatial database. ARC/INFO is a modular, vector based package and is for making cartographic quality out puts in the form of maps and generation of model while the non-spatial data is organized using topological data model while non-spatial attribute data is stored using a database management package.

Three main aspects of any modern technology are: the instrumentation, the software and the manpower. Even though GIS does not demand any special Instrumentation, with the rapidly expanding use of computer technology in India has been a great boon for GIS. Most public sector organizations today are increasingly using GIS technology for the effective utilization of their resources, and its use in private sector is also catching up. In GIS-related software development, even though the contribution of Indian GIS professionals is significant, a lot more needs to be done in developing indigenous software systems for global market. In manpower development also, many academic and professional organizations in India are today offering comprehensive-training courses related to GIS. However, introduction of GIS as a major field of study in Indian engineering and technological institutes and Universities is still a dream, and more efforts are required to be directed in this direction. The GIS-based publications, with their high quality technical contents, are also contributing significantly in spreading GIS awareness.

GIS is very useful to the following persons:

- Urban planner – might like to find out about the urban fringe growth in her/his city, and quantify the population growth that some suburbs are witnessing.
- Mining engineer could be interested in determining which prospect copper mines are best fit for future exploration, taking into account parameters such as extent, depth and quality of the one body, amongst others.

- Natural hazard analyst might like to identify the high-risk areas of annual monsoon-related flooding by looking at rainfall patterns and terrain characteristics.
- Land surveyor working on land development projects for urban expansion.
- Cadastral engineer working on the provision of up-to-date information.
- Regarding real estate and land property.
- Food security officer - working on monitoring systems of crop production.
- Land use suitability analyst working on assessments regarding production of certain crops in semi-arid areas.
- Wildlife manager - working to ensure that wildlife (e.g., elephant) has freedom of movement without damaging human settlements.

An important distinction between GIS application is whether the geographic phenomena studied are man-made or natural. There are many different uses of GIS, as may have become clear from our list of professionals as shown above who deal with geo information. An important distinction between GIS for urban planning purposes involves a study of man-made things: the roads, sidewalks, and at larger scale, suburbs and transportation routes are man-made. These entities are assumed to have clear-cut boundaries.

On the other hand, geomorphologies, ecologists and soil scientists often have natural phenomena as their study objects. They may be looking at rock formations, plate tectonics, and distribution of natural vegetation or soil units.

About Maps

The best-known models of the real world are maps. Maps have been used for thousands of years to represent information about the real world. Their conception and design has developed into a science with a high degree of sophistication. Maps have proven to be extremely useful for many applications in various domains.

A disadvantage of maps is that they are restricted to two-dimensional static representations, and that they always are displayed in a given scale. The map scale determines the spatial resolution of the graphic feature representation. The smaller the scale, the less detail a map can show. The accuracy of the base data, on the

other hand puts limits to the scale in which a map can be sensibly drawn. The selection of a proper map scale is one of the first and most important steps in map design.

Cartography as the science and art of map making functions as an interpreter of translation real world phenomena into correct, clear and understandable representations for our user.

1.2.2 Shape file

This document defines the shape file (.shp) spatial data format and describes why shape files are important. It lists the tools available in Environmental Systems Research Institute, Inc. (ESRI), software for creating shape files directly or converting data into shape files from the formats. This document also provides all the technical information necessary for writing a computer program to create shape files without the use of ESRI ® software for organizations that want to write their own data translators.

Why Shape files?

A shape file stores no topological geometry and attributes information for the spatial features in a data set. The geometry for a feature is stored as a comprising a set of vector coordinates.

Because shape files do not have the processing overhead of a topological data structure, they have advantages over other data sources such as faster drawing speed and edit ability. Shape files handle single features that overlap or that is noncontiguous. They also typically require less disk space and are easier to read and write.

Shape files can support point, line, and area features. Area features are represented as closed loop, double-digitized polygons. Attributes are held in a dBase ® format file. Each attribute record has a one-to-one relationship with the associated shape record.

How Shape files can be created

Shape files can be created with the following four general methods:

- Export -- Shape files can be created by exporting any data source to a shape file using ARC/INFO ®, PC ARC/INFO ®, Spatial Database Engine ™ (SDE ™), Arc View ® GIS, or Business MAP ™ software.
- Digitize -- Shape files can be created directly by digitizing shapes using ARCVIEWGIS feature creation tools.
- Programming -- Using Avenue ™ (Arc View GIS), MapObjects ™, ARC Macro Language (AML ™) (ARC/INFO), or Simple Macro Language (SML ™)(PC ARC/INFO) software, you can create shape files within your programs.
- Write directly to the shape file specifications by creating a program.

SDE, ARC/INFO, PC ARC/INFO, Data Automation Kit (DAK ™), and Arc CAD ® software provide shape-to-coverage data translators, and ARC/INFO also provides a coverage-to-shape translator. For exchange with other data formats, the shape file specifications are published in this paper. Other data streams, such as those from global positioning system (GPS) receivers, can also be stored as shape files or X, Y event tables.

Shape File Technical Description

Computer programs can be created to read or write shape files using the technical specification in this section.An ESRI shape file consists of a main file, an index file, and a dBASE table. The main file is a direct access, variable-record-length file in which each record describes a shape with a list of its vertices. In the index file, each record contains the offset of the corresponding main file record from the beginning of the main file. The dBASE table contains feature attributes with one record per feature. The one-to-one relationship between geometry and attributes is based on record number. Attribute records in the dBASE file must be in the same order as records in the main file.

Naming Conventions

All file names adhere to the 8.3 naming convention. The main file, the index file, and the dBASE file have the same prefix. The prefix must start with an alphanumeric character (a–Z, 0–9), followed by zero or up to seven characters (a–Z, 0–9, _, -). The suffix for the main file is .shp. The suffix for the index file is .shx.

The suffix for the dBASE table is .dbf. All letters in a file name are in lower case on operating systems with case-sensitive file names

Examples

- Main file: countries.shp
- Index file: countries.shx
- DBase table: countries.dbf

Numeric Types

A shape file stores integer and double-precision numbers. The remainder of this document will refer to the following types:

- Integer: Signed 32-bit integer (4 bytes)
- Double: Signed 64-bit IEEE double-precision floating point number (8 bytes)
- Floating point numbers must be numeric values. Positive infinity, negative infinity, and Not-a-Number (NaN) values are not allowed in shape files. Nevertheless, shape files support the concept of "no data" values, but they are currently used only for measures. Any floating point number smaller than –10 38 is considered by a shape file reader to represent a "no data" value.

  The first section below describes the general structure and organization of the shape file. The second section describes the record contents for each type of shape supported in the shape file.

Organization of the Main File

The main file (.shp) contains a fixed-length file header followed by variable-length records. Each variable-length record is made up of a fixed-length record header followed by variable-length record contents. Figure 3 illustrates the main file organization.

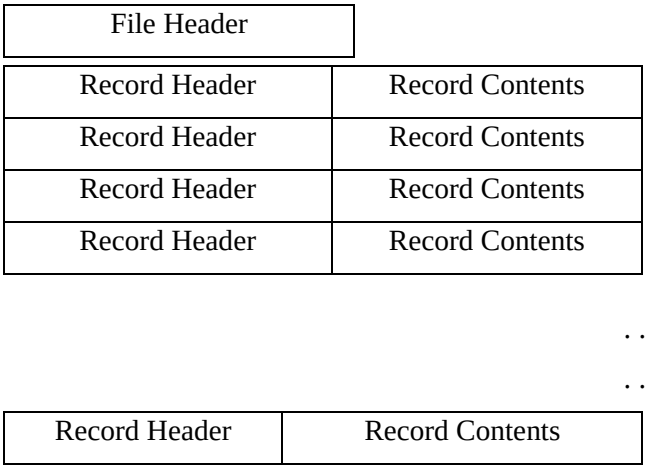

FIGURE:1.2 ORGANIZATION OF MAIN FILE

Byte Order

All the contents in a shape file can be divided into two categories:

Data related

· Main file record contents

· Main file header's data description fields (Shape Type, Bounding Box, etc.)

File management related

· File and record lengths

· Record offsets, and so on

The integers and double-precision integers that make up the data description fields in the file header (identified below) and record contents in the main file are in little endian (PC or Intel ®) byte order. The integers and double-precision floating point numbers that make up the rest of the file and file management are in big endian (Sun ® or Motorola ®) byte order. The Main File Header The main file header is 100 bytes long.

Table 1.2 shows the fields in the file header with their byte position, value, type, and byte order. In the table, position is with respect to the start of the file.

TABLE 1.2 DESCRIPTION OF THE MAIN FILE HEADER

Position	Field	Value	Type	Byte Order
Byte 0	File Code	9994	Integer	Big
Byte 4	Unused	0	Integer	Big
Byte 8	Unused	0	Integer	Big
Byte 12	Unused	0	Integer	Big
Byte 16	Unused	0	Integer	Big
Byte 20	Unused	0	Integer	Big
Byte 24	File Length	File Length	Integer	Big
Byte 28	Version	1000	Integer	Little
Byte 32	Shape Type	Shape Type	Integer	Little
Byte 36	Bounding Box	Xmin	Double	Little
Byte 44	Bounding Box	Ymin	Double	Little
Byte 52	Bounding Box	Xmax	Double	Little
Byte 60	Bounding Box	Ymax	Double	Little
Byte 68*	Bounding Box	Zmin	Double	Little
Byte 76*	Bounding Box	Zmax	Double	Little
Byte 84*	Bounding Box	Mmin	Double	Little
Byte 92*	Bounding Box	Mmax	Double	Little

* Unused, with value 0.0, if not Measured or Z type

The value for file length is the total length of the file in 16-bit words (including the fifty 16-bit words that make up the header).

All the non-Null shapes in a shape file are required to be of the same shape type. The values for shape type are as follows:

Value	Shape Type
0	Null Shapes
1	Point
3	PolyLine
5	Polygons
8	Multipoints
11	PointZ
13	PolyLineZ
15	PolygonZ
18	MultiPointZ
21	PointM
23	PolyLineM
25	PolygonM
28	MultiPointM
31	MultiPatch

Shape types not specified above (2, 4, 6, etc., and up to 33) are reserved for future use. Currently, shape files are restricted to contain the same type of shape as specified above. In the future, shape files may be allowed to contain more than one shape type. If mixed shape types are implemented, the shape type field in the header will flag the file as such.

The Bounding Box in the main file header stores the actual extent of the shapes in the file:

The minimum-bounding rectangle orthogonal to the X and Y (and potentially the M and Z) axes that contains all shapes. If the shapefile is empty (that is, has no records), the values for Xmin, Ymin, Xmax, and Ymax are unspecified. Mmin and Mmax can contain "no data" values for shape files of measured shape types that contain no measures.

Record Headers

The header for each record stores the record number and content length for the record. Record headers have a fixed length of 8 bytes. Table 4 shows the fields

in the file header with their byte position, value, type, and byte order. In the table, position is with respect to the start of the record.

TABLE1.3 DESCRIPTION OF MAIN FILE RECORD HEADERS

Position	Field	Value	Type	Order
Byte 0	Record Number	Record Number	Integer	Big
Byte 4	Content Length	Content Length	Integer	Big

Record numbers begin at 1.

The content length for a record is the length of the record contents section measured in 16-bit words. Each record, therefore, contributes (4 + content length) 16-bit words toward the total length of the file, as stored at Byte 24 in the file header.

Point

A point consists of a pair of double-precision coordinates in the order X, Y.

```
Point
{
Double X // X coordinate
Double Y // Y coordinate
}
```

TABLE 1.4- POINT RECORD CONTENTS

Position	**Field**	**Value**	**Type**	**Number**	**Byte Order**
Byte 0	Shape Type	1	Integer	1	Little
Byte 4	X	X	Double	1	Little
Byte 12	Y	Y	Double	1	Little

PolyLine

A PolyLine is an ordered set of vertices that consists of one or more parts. A part is a connected sequence of two or more points. Parts may or may not be

connected to one another. Parts may or may not intersect one another. Because this specification does not forbid consecutive points with identical coordinates, shapefile readers must handle such cases. On the other hand, the degenerate, zero length parts that might result are not allowed.

PolyLine

```
{
Double [4] Box              // Bounding Box
Integer NumParts            // Number of Parts
Integer NumPoints           // Total Number of Points
Integer[NumParts] Parts  // Index to First Point in Part
Point[NumPoints] Points    // Points for All Parts
}
```

TABLE 1.5- POLYLINE RECORD CONTENTS

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	3	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little

Note: X = 44 + 4 * NumParts

Polygon

A polygon consists of one or more rings. A ring is a connected sequence of four or more points that form a closed, non-self-intersecting loop. A polygon may contain multiple outer rings. The order of vertices or orientation for a ring indicates which side of the ring is the interior of the polygon.

The neighborhood to the right of an observer walking along the ring in vertex order is the neighborhood inside the polygon. Vertices of rings defining holes in

polygons are in a counterclockwise direction. Vertices for a single, ringed polygon are, therefore, always in clockwise order. The rings of a polygon are referred to as its parts. Because this specification does not forbid consecutive points with identical coordinates, shapefile readers must handle such cases. On the other hand, the degenerate, zero length or zero area parts that might result are not allowed.

The Polygon structure is identical to the PolyLine structure, as follows:

Polygon

```
{
Double[4] Box                    // Bounding Box
Integer NumParts                 // Number of Parts
Integer NumPoints                // Total Number of Points
Integer[NumParts] Parts          // Index to First Point in Part
Point[NumPoints] Points          // Points for All Parts
}
```

TABLE 1.6- POLYGON RECORD CONTENTS

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	5	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little

Note: X = 44 + 4 * NumParts

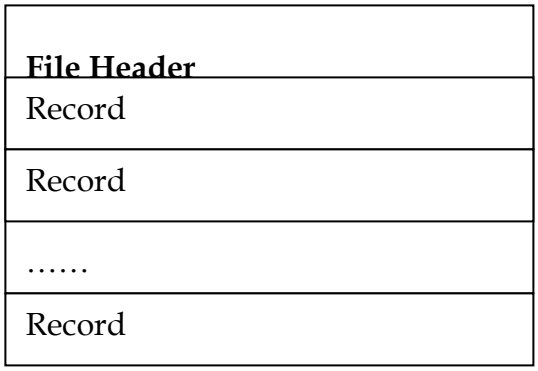

FIGURE 1.3 ORGANIZATION OF THE INDEX FILE

The Index File Header: The index file header is identical in organization to the main file header described above.

The file length stored in the index file header is the total length of the index file in 16-bit words (the fifty 16-bit words of the header plus 4 times the number of records).

Index Records The I'th record in the index file stores the offset and content length for the I'th record in the main file. Table shows the fields in the file header with their byte position, value, type, and byte order. In the table, position is with respect to the start of the index file record.

TABLE 1.7-DESCRIPTION OF INDEX RECORDS

Position	Field	Value	Type	Byte Order
Byte 0	Offset	Offset	Integer	Big
Byte 4	Content Length	ContentLength	Integer	Big

The offset of a record in the main file is the number of 16-bit words from the start of the main file to the first byte of the record header for the record. Thus, the offset for the first record in the main file is 50, given the 100-byte header.

The content length stored in the index record is the same as the value stored in the main file record header.

Organization of the dBase File

The dBASE file (.dbf) contains any desired feature attributes or attribute keys to which other tables can be joined. Its format is a standard DBF file used by many table-based applications in Windows ™ and DOS. Any set of fields can be present in the table.

There are three requirements, as follows:

- The file name must have the same prefix as the shape and index file. Its suffix must be .dbf.
- The table must contain one record per shape feature.
- The record order must be the same as the order of shape features in the main (*.shp) file.
- The year value in the dBASE header must be the year since 1900.

A shape file stores nontopological geometry and attributes information for the spatial features in a data set. The geometry for a feature is stored as a shape comprising a set of vector coordinates.

Because shape files do not have the processing overhead of a topological data structure, they have advantages over other data sources such as faster drawing speed and editability. Shape files handle single features that overlap or that is noncontiguous. They also typically require less disk space and are easier to read and write. Shape files can support point, line, and area features. Area features are represented as closed loop, double-digitized polygons. Attributes are held in a dBASE® format file. Each attribute record has a one-to-one relationship with the associated shape record.

An ESRI shape file consists of a main file, an index file, and a dBASE table. The main file is a direct access, variable-record-length file in which each record describes a shape with a list of its vertices. In the index file, each record contains the offset of the corresponding main file record from the beginning of the main file. The dBASE table contains feature attributes with one record per feature. The one-to-one relationship between geometry and attributes is based on record number. Attribute records in the dBASE file must be in the same order as records in the main file.

The main file (.shp) contains a fixed-length file header followed by variable-length records. Each variable-length record is made up of a fixed-length record header followed by variable-length record contents. The index file (.shx) contains a 100-byte header followed by 8-byte, fixed-length records.

The dBASE file (.dbf) contains any desired feature attributes or attributes keys to which other tables can be joined. Its format is a standard DBF file used by many table-based applications in Windows™ and DOS. The file name must have the same prefix as the shape and index file. Its suffix must be .dbf.

1.2.3 Network

Network: The collection of cables, pipelines, wires, roads that enables the movement of energy, commodities, information, traffic constitutes a network.

Networks can be used to model the flow of goods and services. There are two primary types of networks:

1) Radial
2) Looped

In Radial or tree networks flow always has an upstream and downstream direction. For eg: Stream and storm drainage systems are examples of radial networks.

In looped networks, self intersections are common occurrences. For eg: Water distribution networks are looped by design to ensure that service interruptions affect the fewest customers.

In GIS software systems, networks are modeled as points(For eg: street intersections, fuses, switches, water valves) and lines(For eg: Streets, transmission lines, pipes).Network topological relationships define how lines connect with each other at nodes.For the purpose of network analysis it is also useful to define rules about how flows can move through a network. For eg: in a sewer network flow is directional from a customer (Source) to a treatment plant(sink),but in a pressurized gas network flow can be in any direction.

A network is defined as a directed graph G = (N, A) consisting of a set N of nodes and a set A of arcs with associated numerical values, such as the number of nodes, n=|N|, the number of arcs, m=|A|, and the length of an arc connecting nodes i and j, denoted as l(i,j).

The shortest path problem can be stated as follows: given a network, find the shortest distances (least costs) from a source node to all other nodes or to a subset of nodes on the network. These shortest paths represent a directed tree T rooted from a source node s with the characteristic that a unique path from s to any node i on the network is the shortest path to that node. The length of the shortest path from s to any node i is denoted as d (i). This directed tree is called a shortest path tree. For any network with n nodes, one can obtain n distinctive shortest path trees.

Shortest paths from one (source) node to all other nodes on a network are normally referred as one-to-all shortest paths. Shortest paths from one source node to a subset of the nodes on a network can be defined as one-to-some shortest paths. Shortest paths from every node to every other node on a network are normally called all-to-all

shortest paths. when the goal is to obtain a one-to-one shortest path or one-to-some shortest paths, the Dijkstra algorithm offers some advantages because it can be terminated as soon as the shortest path distance to the destination node is obtained.

1.3 EXISTING SYSTEM

1.3.1 ArcView GIS 3.2 a

It's a powerful, easy-to-use tool that brings geographic information to the desktop. ArcView gives one the power to visualize, explore, query and analyze data spatially.

ArcView is made by Environmental Systems Research Institute (ESRI), the makers of ARC/INFO, the leading geographic information system(GIS) software. It has been helping people solve spatial problems with computers for over 20 years.

One doesn't need to know how to create geographic data in order to use ArcView. ArcView comes with a useful set of ready-to-use data. Additional geographic data sets are available from ESRI and from various third parties to suit almost any requirement one might have. Plus, if one's organization uses ARC/INFO data, one will immediately be able to use ArcView to access all these resources, including vector coverage, map libraries, grids, images and event data.

Working spatially

ArcView can be used by anyone who wants to work spatially. A key feature of ArcView is that it's easy to load tabular data, such as dBASE files and data from database servers, into ArcView so that one can display, query, summarize, and organize this data geographically.

In no time one will be working with the data in a completely new way, seeing patterns not seen before, understanding geographic relationships that were previously hidden, gaining new insights… and achieving new results for business.

Views

With ArcView one works with geographic data in interactive maps called views. Every view features ArcView's unique geographic "Table of Contents", making it easy to understand and control what's displayed. GIS has never been simpler!

Tables

Working with tabular data in ArcView's tables puts one in control. Click on features on a view, and their records highlight in the table showing their attributes. Select records in the table and the features they represent highlight on the view. ArcView's tables have a full range of features for obtaining summary statistics, sorting and querying.

Charts

ArcView's charts offer a powerful business graphics and data visualization capability that is fully integrated into ArcView's geographic environment. One can

simply click on features on a view to add them to the chart. ArcView lets you work simultaneously with geographic, tabular and chart representations of your data.

Layouts

ArcView's layouts lets one create high quality, full color maps by first arranging the various graphic elements on-screen the way one wants them. One will get great looking results on a wide range of printers and plotters. Layouts are smart because they have a live link to the data they represent. When one prints a layout, any changes to the data are automatically included, so one knows everything on the map will be up-to-date.

Scripts

ArcView scripts are macros written in Avenue, Arc View's programming languages and development environment. With Svenue one can customize almost every aspect of ArcView, from adding a new button to run a script one writes, for creating an entire custom application that can distribute.

Projects

All the components of the ArcView session: views, tables, charts, layouts, and scripts are conveniently stored in one file called a project. ArcView's Project window shows the contents of the project and makes it easy to manage all the work.

Arc View Network Analyst:

The ArcView Network Analyst extension enables you to solve a variety of problems based on geographic networks (e.g.., streets, highways, rivers, pipelines, utility lines). Solve problems such as finding the most efficient travel route across town, generating travel directions, finding the closest emergency vehicle or service facility to an incident, or defining service areas or sales territories based on travel time.

Find the best routes

Improve the efficiency of operation by finding the best routes around town or across the country. Best routes can be found based on the shortest distance or time

between where you are and where you want to go.To solve your problem directions are needed. The ArcView Network Analyst allows to generate detailed directions along the route, providing as-you-need-it solutions for common problems.

1.3.2 ArcMap

ArcMap is also a software for desktop GIS and mapping. It provides better functionality for editing as compared to ArcView.
ArcMap provides a rich set of tools that perform many common network analysis tasks on geometric network.

1.3.3 Pragati GIS

Pragati is the project which is being developed at the Bisag. Its main aim is to support the features of the GIS. It is necessary to study and understand ArcView and ArcMap because it aims to include all the functionality of the ArcView and the ArcMap

Pragati is software which is built at BISAG and which is made on the lines of ArcView software. Actually ArcView costs a lot of rupees and hence all the government organization who wants to use ArcView has to purchase it by paying a huge amount of money. So BISAG decided to build a software which is identical to ArcView and hence the result was Pragati software. Till now various facilities has been added to Pragati and still other modules are being added to built it just like ArcView. Our project that is to build and an ActiveX control that implements various Pen and Brush style is also a module to be added to this software.

Pragati software is build using Visual Basic 6.0 as it provides an excellent Graphical user interface. While the whole code is written in Visual c++ as it is very good at graphic application with a rich set of graphic functions. Again for graphic application time to run the application is very critical and VC++ is faster at run time and hence it is the most suitable platform to write the whole logic.

1.4 CLIENT AND TARGETED USERS OF PROJECT

Client: Organization purchasing Pragati GIS.

Targeted users: Employees of the organization using Pragati GIS.

Chapter 2
THESIS PROFILE

- Objective
- Scope
- Thesis Platform
- Summary

2.1OBJECTIVE

Since 1998 BISAG (Gandhinagar) has been working on the task of creating GIS applications, which can be distributed independently throughout the various Governmental Departments of the Gujarat State. They all require GIS software for simplifying their work. There are various GIS softwares available in the market like Arc/Info, Arc View, MapInfo etc, but these softwares are very costly .So BISAG decided to develop its own GIS software (Pragati GIS) that can meet all the needs of government. Arc View includes the facility of Network Analysis which has not been included in Pragati GIS and since Network Analysis helps in solving many real world problems I decided to work on it.

2.2 SCOPE

Project Definition

To develop Network Analyst for GIS using Microsoft Visual C++ and Microsoft Visual Basic 6.0.

Feature

Network analysis is one of the cornerstones of GIS functionality. The shortest path problem is to find the shortest distances (least costs) from a source node to all other nodes or to a subset of nodes on the network. Network Analysis has wide applicability like it helps in selecting optimum paths for inter-cities traveling, transportation vehicles of domestic, directing the bus fleet of tourism corporations and international transportation companies ,So different parts of the routes network are evaluated considering their distance and required time to pass each part to analyze the optimum route to take.

Network Analyst helps in finding the shortest path between any points selected on a network.

2.3 THESIS PLATFORM

2.3.1 Platform Details

Operating System:- Windows 2000

Programming Languages:- Visual C++ and Visual Basic

To implement, test and validate the efficiency of the algorithms concerning the Intersection and Union of the maps we need to develop language skill in

-Visual Basic

-Visual Studio

We need to implement our research work using two different languages because we need to do two things,

- GUI is to be done in **Visual Basic** which has to be integrated in Pragati S/W
- Coding is to be done in **Visual C++** which has to be integrated in GGIS control

As we are using VB to develop User Interface, the user is provided with an excellent Graphical User Interface (GUI) and project is very user friendly. So output or result will be displayed as an image to the user and it is stored in the shape file format. This graphical display of the output is stored .shp format.

As it has attributes related to each feature point, line, polygon stored in the database file, user can have the required attribute values of the resultant map in the database file stored with the .dbf format.

So we have used "Microsoft Visual Studio" for the implementation of research work in Visual Basic and Visual C++.

2.3.2 Motives for Platform

Windows 2000

Windows 2000 Professional is suitable operating system for geographic system development and also for the use of the software related to the geographic system. Also it gives more update facilities as Microsoft's new Web-based resource site, automates driver and system file updates, and provides up-to-date technical support.

Windows 2000 can review device drivers and system software on your computer, compare those findings with a master database on the Web, and then recommend and install updates specific to your computer. You can also revert to a previous device driver or system file using the uninstall option.

Visual C++

The main reason for using Microsoft VC++ as the development tool is that it provides a variety of new and improved features for managing projects and subprojects, editing code and resources, managing classes, code reuse and code generation and also we need to integrate the module with the existing system.

Visual C++ 6.0 includes the comprehensive Microsoft foundation classes, which simplify and speed development of window applications. It includes sophisticated resource editor to design the complex dialog boxes, menu, toolbars, images, and many other elements of modern window application. There is an excellent integrated development environment called developer studio that present graphical views of your application's structure as you develop it. Totally integrated debugging tool lets you examine in minute detail every aspect of your program as it runs.

Visual Basic

Visual Basic 6.0 has emerged as one of the standard windows programming language and it has become must for all software people for developing applications in visual environment, rather than writing numerous lines of code to describe the appearance(GUI) and location of interface elements, VB provides method to simply

add objects and place them on screen also it has rich set of APIs to support easy and quick development.

2.3.3 Features of the languages

Visual C++

The Project Workspace

The Project Workspace window offers three views: in Class View project classes, member variables and functions are shown; in File View files comprising a project are visible and in Resource View resource file components can be manipulated.

Editor

The editor offers improved compatibility with other popular editors.

The Class Wizard and the Wizard Bar

The Class Wizard offers support for OLE control development. Many Class Wizard features can easily be accessed in the Wizard Bar.

Component Gallery

Classes that are created through Class Wizard or entire projects created through AppWizard can be added to the Component Gallery and later inserted to other applications. The Component Gallery comes with many useful tools including OLE controls etc.

Debugging

The integrated debugger has redesigned windows. It also offers a new feature data tips i.e. positioning the mouse cursor over a symbol in an editor window during debugging causes a tool tip style window to appear, displaying the current value of the symbol.

Integration with other Tools

The Developer Studio offers a high level of integration with Microsoft development tools such as the Microsoft Developer Library.

Visual Basic

Visual Basic can serve as an ideal front end tool for the client to interact. It has got connectivity mechanism for all types of database situated far and wide in a network and so it can gratify to the needs of large body of clients. Using the latest ActiveX technologies it can integrate the functionalities provided by other applications. The final application is a true EXE file and so can be freely distributed.

2.4 SUMMARY

Project Title

"Network Analyst For GIS"

Front End Tools

Microsoft Visual Basic

GUI is to be done in Visual Basic which has to be integrated in Pragati S/W

Back End Tools

Microsoft Visual C++

Coding is to be done in **Visual C++** which has to be integrated in GGIS control

Project Guide

Shri Shashikant Sharma

Organization

Bhaskaracharya Institute for space Application and GeoInformatics,
Govt. of Gujarat(Science & Technology),
Gandhinagar.

Submitted By

Neha Rathore

Submitted To

Institute Of Technology,
Nirma University
Ahmedabad.

Chapter 3

SYSTEM ANALYSIS

- Problem definition
- Fact Finding Techniques
- Requirement Analysis
- Feasibility Analysis

3.1 PROBLEM DEFINITION

To develop Network Analyst for GIS using Microsoft Visual C++ and Microsoft Visual Basic 6.0.

3.2 FACT FINDING TECHNIQUES

Fact Finding Techniques

During requirement determination phase, the system analyst has to find out how the current system works and what is expected from a new system, for that it is required to spend considerable time talking to users and gathering all relevant information for the project.

Information Sources

- Users of the system.
- Forms and documents used in the organization.
- Procedure manuals and rulebooks which specify various activities carried out in the organization.
- Various reports used in the organization.
- Computer programs of existing system.

3.3 REQUIREMENT ANALYSIS

The description of the service and the constraints are the requirements of the system and the processes involved in the requirement engineering are:

- Finding out
- Analyzing
- Documenting and
- Checking these services and constraints

The process activities are:

Domain Understanding:

Analyst must develop understanding of the application domain and therefore some initial time was spent for domain understanding like using maps, becoming aware of map manipulation software etc.

Help was taken from software like Arc View 3.2a(includes Network Analyst Extension) which helped to understand what to build and what are the advantages and disadvantages of that. User was considered as a point of view for developing a user friendly system.

Requirement collection:

The conclusion of domain understanding should result into the bulk of information and from that requirement classification is done.

Requirement Classification:

Requirements are classified as follows:

The main requirements are

- User requirements
- System requirements

Above Requirements can be further classified as:

- Functional requirements
- Non-functional requirements
- Domain requirements

User Requirements

There are mainly following types of users:-

- Visitors and tourists

  Visitors and tourists are mainly interested in beautiful locations, emergency services like Banks, police station, hotels etc.

- Citizens

  Citizens want to get information about emergency services like Banks, police station, hotels etc.

- Govt. Officials

  Different departments of any govt. can use the system to acquire information about any place.

- Education Departments

  Distance of nearest Higher secondary , secondary and primary school.

- Health Departments

  Distance of nearest private hospital, govt. hospital.

- Other social Amenities

  Distance of nearest Community Hall etc.

System Requirements

System Requirements state which kind of services, functions and facility should be given to users. Users are allowed to retrieve information of places.

Functional Requirements

Since this project uses database and control, this needs the retrieval of information from the database. It needs the interaction between Visual Basic Container and Visual C++ control.

Non Functional Requirements

Product Requirements

- Efficiency

  The system should provide easy and fast access without consuming more cost.

- Reliability

  User should never be surprised by the behavior of the system and it should also provide meaningful feedback when errors occur so that user can recover from errors.

Conflict Resolution:

Here, requirement conflicts are handled so that users can distinguish themselves. Like some facility may not be used by other user, there must not be any objection from other users.

Prioritization

The modules were developed step by step. A module for displaying any shape file entered by user was developed first . After that the Shapefile which was added onto the Network Analyst was converted into a graph and then the algorithm eas implemented.

Requirement Validation:

Requirement validation is concerned with showing that requirements actually define the system. If this validation is inadequate errors in the requirements will be propagated to the system design and implementation.

Requirement are checked to discover if they are complete, consistent and in accordance with what visitor, citizens, govt. officials and other users want from the projected system.

3.4 FEASIBILITY ANALYSIS

A feasibility study is a short, focused study which aims to answer a number of questions:

- Does the system contribute to the overall objectives of the organization?
- Can the system be implemented using current technology and given cost and schedule constraints?
- Can the system be integrated with systems which are already in place?

Operational Feasibility

Operational Feasibility measures how well the solution will work in the organization and how will end-user and management feels about system. Proposed

system is helpful for tourists, citizen and visitors. It will help them to get appropriate and adequate information.

On studying the operational feasibility of the project the following conclusions were derived:

- Developed system will provide the adequate throughput and all necessary information to end users.
- It will provide advantageous and reliable services.

Thus, it is operationally feasible to develop the proposed system.

Technical Feasibility

Technical Feasibility tries to answer the following questions to make the software feasible to develop.

- The software or tools necessary for building or running the application are easily available or not?
- Compatibility amongst software exists or not?
- Are developers aware of these technologies?
- What about the alternative of these chosen technologies?

Factors Considered

Here we have to consider those tools which will be required for developing the project?

- The tools that are available and tools that will be required are taken into account.
- I have to work on Visual Basic and Visual C++ for GIS. The company's former versions are developed in VC++, So the company already has the licensed version of Visual Studio with MSDN.
- Various Shapefiles are desired as maps are to be displayed with the help of shapefiles. It is desired to have as much as shapefiles possible. BISAG is Govt. of Gujarat enterprise .So, it has all official maps, DBF files as data source and index files are also available.
- Well known softwares like Arc View, Arc Map are available.

Considering all the above points it is technically feasible to carry on the project.

Economical feasibility

Economical feasibility addresses to the following issues:

- Is the organization having suitable budget to develop the proposed system?
- How much profit can be earned from the system by an organization?
- Would it be cost-effective to develop the system or it is worthwhile to remain with the current system?
- As the development tools are available free of cost, there isn't any burden of buying them .Pragati does not have Network Analyst, so it is certainly required .Extra funds are not required to develop the system, so it is economically feasible to the organization.

Implementation feasibility

Under the study of Implementation feasibility following issues are considered:

- Is it possible to install the software within the given environment?
- Will organization, user support for the installation of software?
- Will proposed system cause any harm to the operations of the organization?

Operationally, this system can be installed and it will work according to it's functionality.

Chapter 4

SYSTEM DESIGN

- System design
- Architectural design
- Data flow design
- Data flow diagram
- Flow chart

4.1 SYSTEM DESIGN

During the designing of the system I followed the modular approach .Firstly I had discussed with Senior Scientist the basic needs and usages of the project to be undertaken and then I took help from Arc View 3.2a which includes the Network Analyst extension in order to understand it's operation, so that various parameters and conditions to be considered can be known. After that I started the algorithmic design of the system.

4.2 ARCHITECTURAL DESIGN

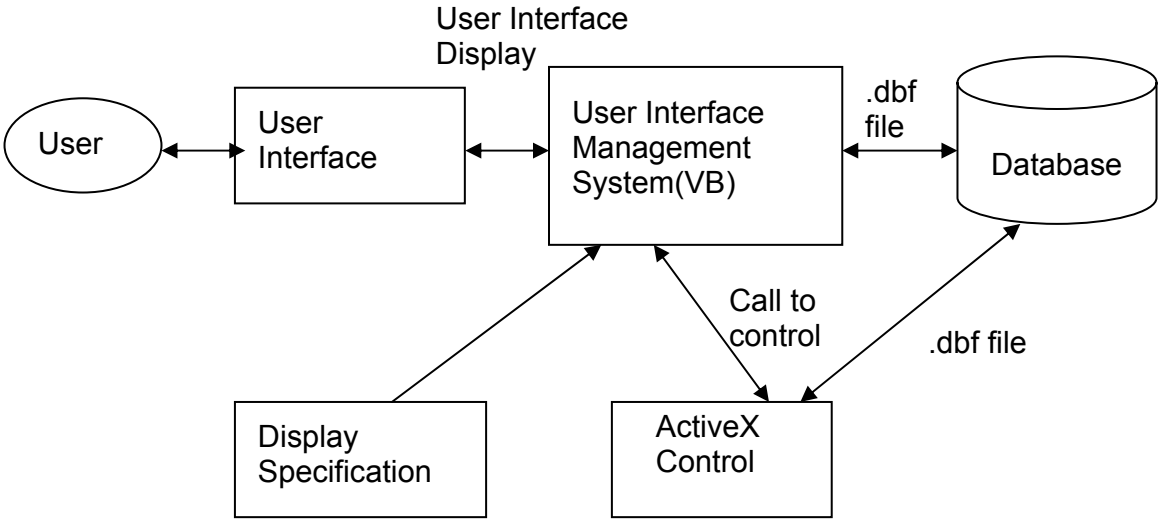

Fig:4.1 Architectural Design of the Proposed System

The figure shows the interaction of user with system.The major components of the model are :-

User:

User interacts with the container(Visual Basic).It does not know the internal operation of the system.

User Interface Management System:

This calls various methods,properties developed in the ActiveX control. It interacts with the ActiveX control.

ActiveX Control:

This does the internal operations.various methods and properties are developed in this.

Database:

Database stores data necessary for the process.

4.3 PROCEDURAL DESIGN

4.3.1 Different Shortest path algorithms:

I Bellman –Ford Algorithm solves the single source shortest- paths problem in the general case in which edge weights may be negative.

```
BELLMAN-FORD(G,w,s)
 INITIALIZE-SINGLE-SOURCE(G,s)
    for <-1 to V[G]-1
     do for each edge (u,v)εE[G]
            do RELAX(u,v,w)
   for each edge(u,v)εE[G]
     do if d[v] > d[u]+w(u,v)
            then return FALSE
    return TRUE

 INITIALIZE-SINGLE-SOURCE(G,s)
 for each vertex vεV[G]
    do d[v]<-∞
            Л[v]<-NIL
  d[s]<-0

RELAX(u,v,w)
If d[v]>d[u]+w(u,v)
    Then d[v]<-d[u]+w(u,v)
```

Л[v]<-u

Complexity of Bellman Ford algorithm is $O(V^2)$

II Floyd-Warshall Algorithm

```
FLOYD-WARSHALL(W)
n<-rows[W]
D<-W
for k<-1 to n
        do for i<-1 to n
                do for j<-1 to n
                        do dij<-min(dij,dik+dkj)
Return D
```

dij be the weight of a shortest path from vertex i to vertex j.

W-input nxn matrix.

D-matrix of shortest path weights.

Complexity of Floyd Warshall algorithm is $O(V^3)$

III Dijkstra's Algorithm for finding Shortest Path

Dijkstra's Algorithm solves the single source shortest- paths problem on a weighted directed graph G=(V,E) for the case in which all the edge weights are non-negative.It turns out that one can find the shortest paths from a given source to all points in a graph in the same time; hence this problem is sometimes called the *single-source shortest paths* problem. This problem is related to the spanning tree one. The graph representing all the paths from one vertex to all the others must be a spanning tree - it must include all vertices. There will also be no cycles as a cycle would define more than one path from the selected vertex to at least one other vertex. For a graph,

G= (V, E) Where V is the set of vertices and E is the set of edges.

Dijkstra's algorithm keeps two sets of vertices:
S the set of vertices whose shortest paths from the source have already been determined and
V-S the remaining vertices.

The other data structures needed are:
d array of best estimates of shortest path to each vertex
pi an array of predecessors for each vertex.

The basic mode of operation is:

1. Initialise *d* and *pi*,
2. Set *S* to empty,
3. While there are still vertices in *V-S*,
    i. Sort the vertices in *V-S* according to the current best estimate of their distance from the source,
    ii. Add *u*, the closest vertex in *V-S*, to *S*,
    iii. *Relax* all the vertices still in *V-S* connected to *u*

Relaxation:

The *relaxation* process updates the costs of all the vertices, *v*, connected to a vertex, *u*, if we could improve the best estimate of the shortest path to *v* by including *(u, v)* in the path to *v.*

The relaxation procedure proceeds as follows:

```
initialise_single_source (Graph g, Node s )
  for each vertex v in Vertices( g )
     g.d[v]:= infinity
     g.pi[v]:= nil
  g.d[s] := 0;
```

This sets up the graph so that each node has no predecessor ***(pi[v] = nil***) and the estimates of the cost (distance) of each node from the source (*d[v])* are infinite, except for the source node itself (*d[s] = 0).*

A further way to store a graph (or part of a graph - as this structure can only store a spanning tree), the *predecessor sub-graph* - the list of predecessors of each node,
pi[j], 1 <= j <= |V|
The edges in the predecessor sub-graph are *(pi[v],v).*

The relaxation procedure checks whether the current best estimate of the shortest distance to *v (d[v])* can be improved by going through *u* (*i.e.* by making *u* the predecessor of *v):*

```
relax (Node u, Node v, double w[][] )
   if d[v] > d[u] + w[u,v] then
      d[v] := d[u] + w[u,v]
      pi[v] := u
```

The algorithm itself is now:

```
shortest_paths (Graph g, Node s)
   initialise_single_source (g, s)
   S: = {0}        /* Make S empty */
   Q: = Vertices (g) /* Put the vertices in a PQ */
   While not Empty (Q)
      u: = ExtractCheapest (Q );
      AddNode(S, u); /* Add u to S */
      for each vertex v in Adjacent ( u )
         relax ( u, v, w)
```

Operation of Dijkstra's Algorithm:

This sequence of diagrams illustrates the operation of Dijkstra's Algorithm.

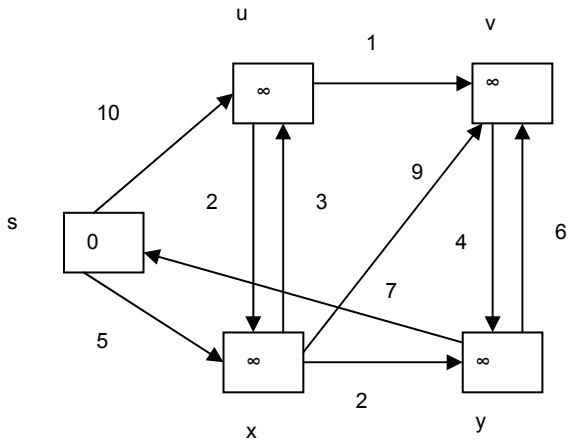

FIGURE 4.2

Initial graph

All nodes have infinite cost except the source

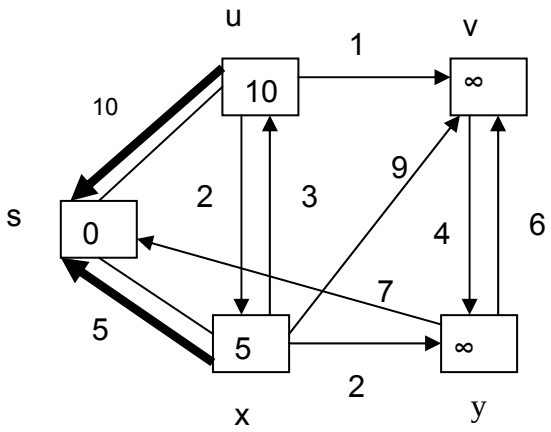

FIGURE 4.3

Choose the closest node to **s**.

As we initialized d[s] to 0, it's **s**.

Add it to **S** Relax all nodes adjacent to **s**.

Update predecessor for all nodes updated.

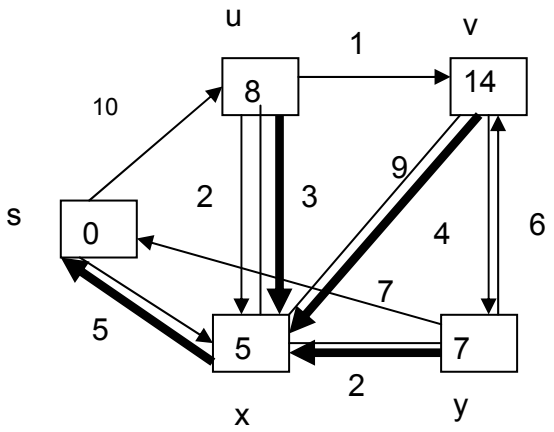

FIGURE 4.4

Choose the closest node, **x**

Relax all nodes adjacent to **x**

Update predecessors for **u, v** and **y**.

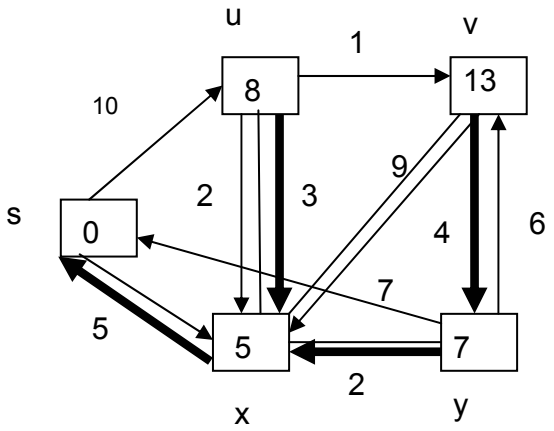

FIGURE 4.5

Now **y** is the closest, add it to **S**.

Relax **v** and adjust its predecessor.

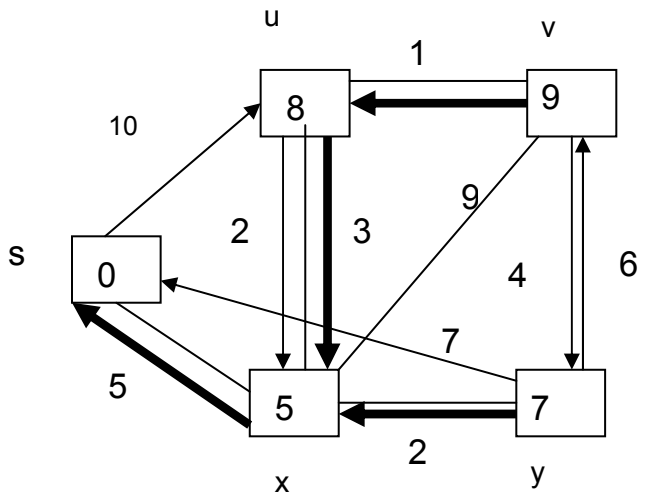

FIGURE 4.6

u is now closest, choose it and adjust its neighbour, **v**.

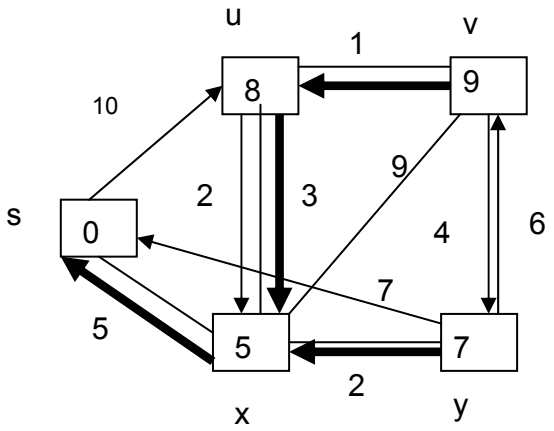

FIGURE 4.7

Finally, add **v**. The predecessor list now defines the shortest path from each node to **s**

Complexity of Dijkstra's algorithm is O(VlogV)

By studying the above three algorithms, it is concluded that Dijkstra's Algorithm is the best suited for real Networks. So, I have implemented Dijkstra's Algorithm to find the Shortest path.

Implementation of shortest path algorithm

```
1.Call Dijkstra_FindPath(NodeNumber,Graph,NodeStartIndex,NodeEndIndex,
   PathlengthMin,PathIndexNumber,PathIndexArray[])
i)For each Node
   DijkstraNodeStructArray[Node].NodeDistance =∞
   DijkstraNodeStructArray[Node].NodePreviousIndex= ∞
   End of for loop
      DijkstraNodeStructArray[NodeStartIndex].NodeDistance = 0
      DijkstraNodeStructArray[NodeStartIndex].NodePreviousIndex = ∞
   ii) Call Dijkstra_Enqueue(NodeIndex, NodeDistance, NodePreviousIndex)
   iii) While (DijkstraclsNumber > 0)
         Call Dijkstra_Dequeue(NodeIndex, NodeDistance, NodePreviousIndex)
            For each Node weight of two adjacent node is assigned to
            NodeCostCurrent.
                  If  NodeCostCurrent != ∞ Then
                        If ((DijkstraNodeStructArray[Node].NodeDistance = ∞) Or
               (DijkstraNodeStructArray[Node].NodeDistance > (NodeCostCurrent
               + NodeDistance))) Then
            DijkstraNodeStructArray[Node].NodeDistance = NodeDistance +
            NodeCostCurrent
              DijkstraNodeStructArray[Node].NodePreviousIndex = NodeIndex
            Call Dijkstra_Enqueue(Temp, NodeDistance + NodeCostCurrent,
            NodeIndex)
                              End of If statement
                  End of If statement
           End of For loop
```

End of while loop

iv) PathIndexArray[PathIndexNumber] = NodeEndIndex

v)MinimumCost= DijkstraNodeStructArray[NodeEndIndex].NodeDistance

4.3.2 Data Structure

Graph is used to represent the Network for which shortest path is to be found out.

A graph is an Abstract Data Type (ADT) that consists of a set of nodes and a set of edges that estalish relationships(connections) between the nodes.In typical graph implementations nodes are implemented as structures or objects. Each road which is uniquely identified by it's road id is considered as a node of the graph.

The two basic choices are adjacency matrices and adjacency lists.

An *adjacency matrix* is an nXn matrix *M* where (typically) *M*[*i*,*j*] = 1 if there is an edge from vertex *i* to vertex *j* and *M*[*i*,*j*]=0 if there is not. Adjacency matrices are the simplest way to represent graphs. However, they use $O(n^2)$space no matter how many edges are in the graph.Although there is some potential for saving space by packing multiple bits per word or simulating a triangular matrix for undirected graphs, these cost some of the simplicity that makes adjacency matrices so appealing.

An *adjacency list* consists of an *n*-element array of pointers, where the *i*th element points to a linked list of the edges incident on vertex *i*. To test whether edge (*i*,*j*) is in the graph, we search the *i*th list for *j*. This takes O(di), where di is the degree of the *i*th vertex. For a complete or almost complete graph, di=Θ(n), so testing the existence of an edge can be very expensive relative to adjacency matrices. The main drawback is the complexity of dealing with linked list structures. Things can be made arbitrarily complex by adding extra pointers for special purposes. For example, the two versions of each edge in an undirected graph, (*i*,*j*) and (*j*,*i*), can be linked together by a pointer to facilitate deletions.

4.3.3 Data Flow Diagram

Data flow design is concerned with designing a sequence of functional transformations that convert system inputs into the required outputs.The design is represented as data flow diagrams. These diagrams illustrate how data flows through a system and how output is derived from the input through a sequence of functional transformations.

Data-flow diagrams are a useful and intuitive way of describing a system.They are normally understandable without special training,especially if control information is excluded. They show end to end processing.

Conventions used in drawing the data flow Diagrams are:-

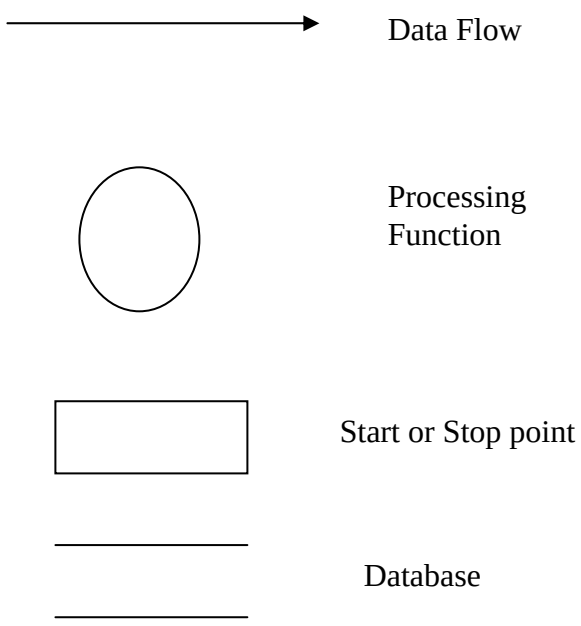

FIGURE 4.8 CONVENTIONS USED IN DATA FLOW DIAGRAM

Context Level Diagram

Context level diagram shows the main task of the process.

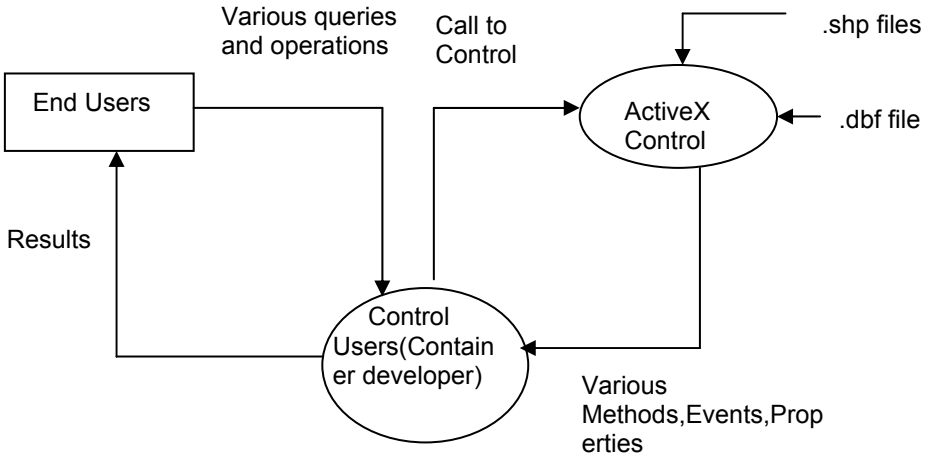

FIGURE 4.9 CONTEXT LEVEL DIAGRAM

End User: The user of the system.

ActiveX Control: This is developed in microsoft Visual C++ for internal operations and is not application dependent.This can be used for any type of software.

Container(User Interface): This is developed in Microsoft visual Basic and is visible to user.This is user friendly and easily understandable to both technical and non technical users.

First level Data Flow Diagram

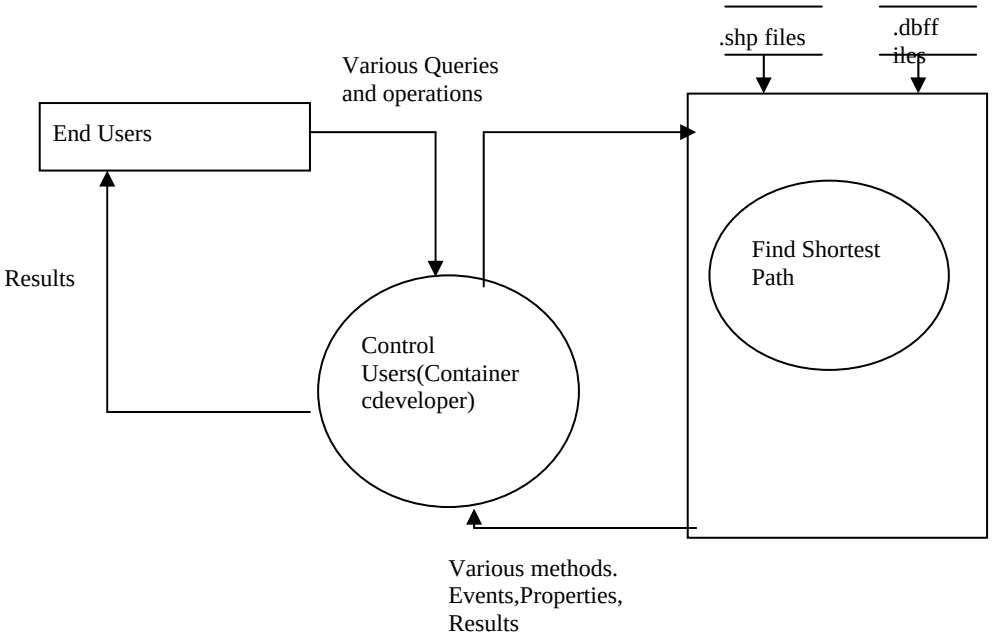

Fig: 4.10 FIRST LEVEL DATA FLOW DIAGRAM

Second Level Data Flow Diagram

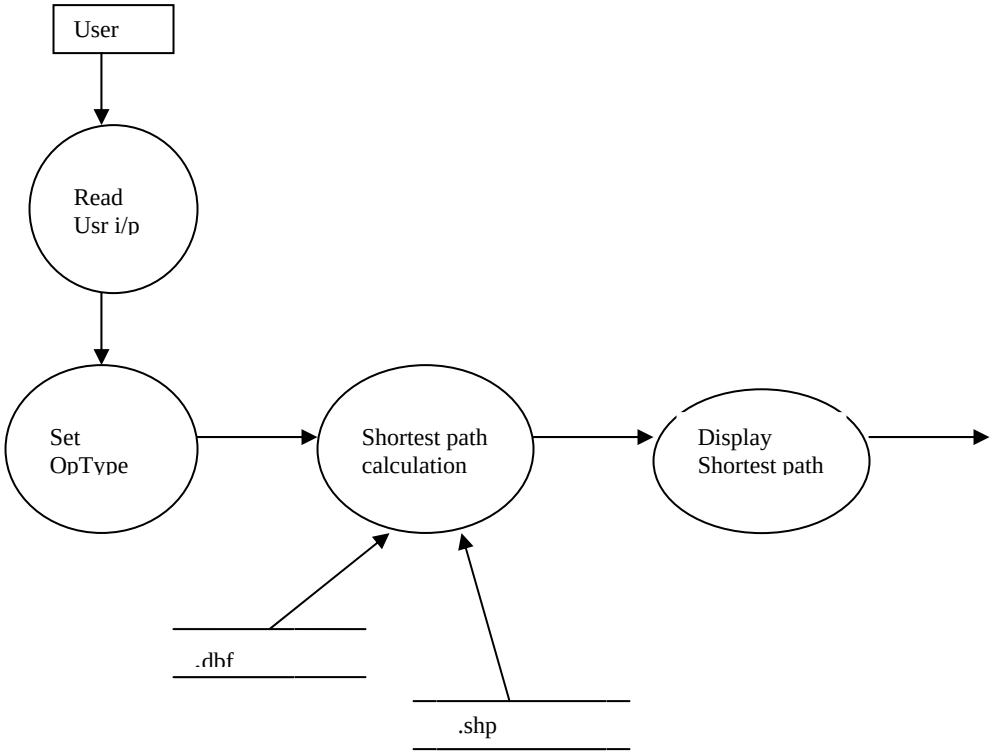

FIGURE 4.11 SECOND LEVEL DATA FLOW DIAGRAM

4.4 FLOW CHARTS

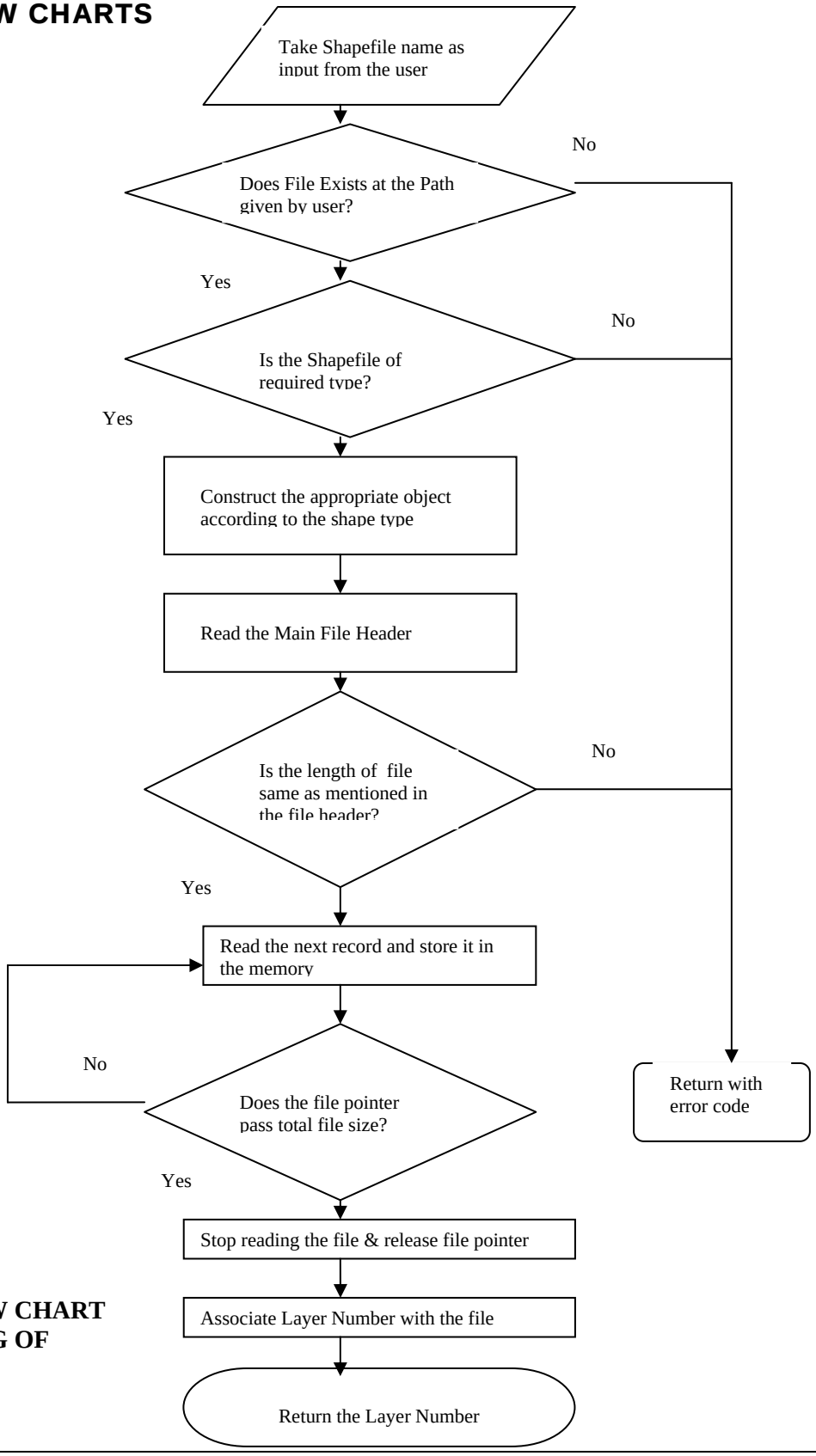

FIG 4.12 FLOW CHART FOR READING OF SHAPEFILE

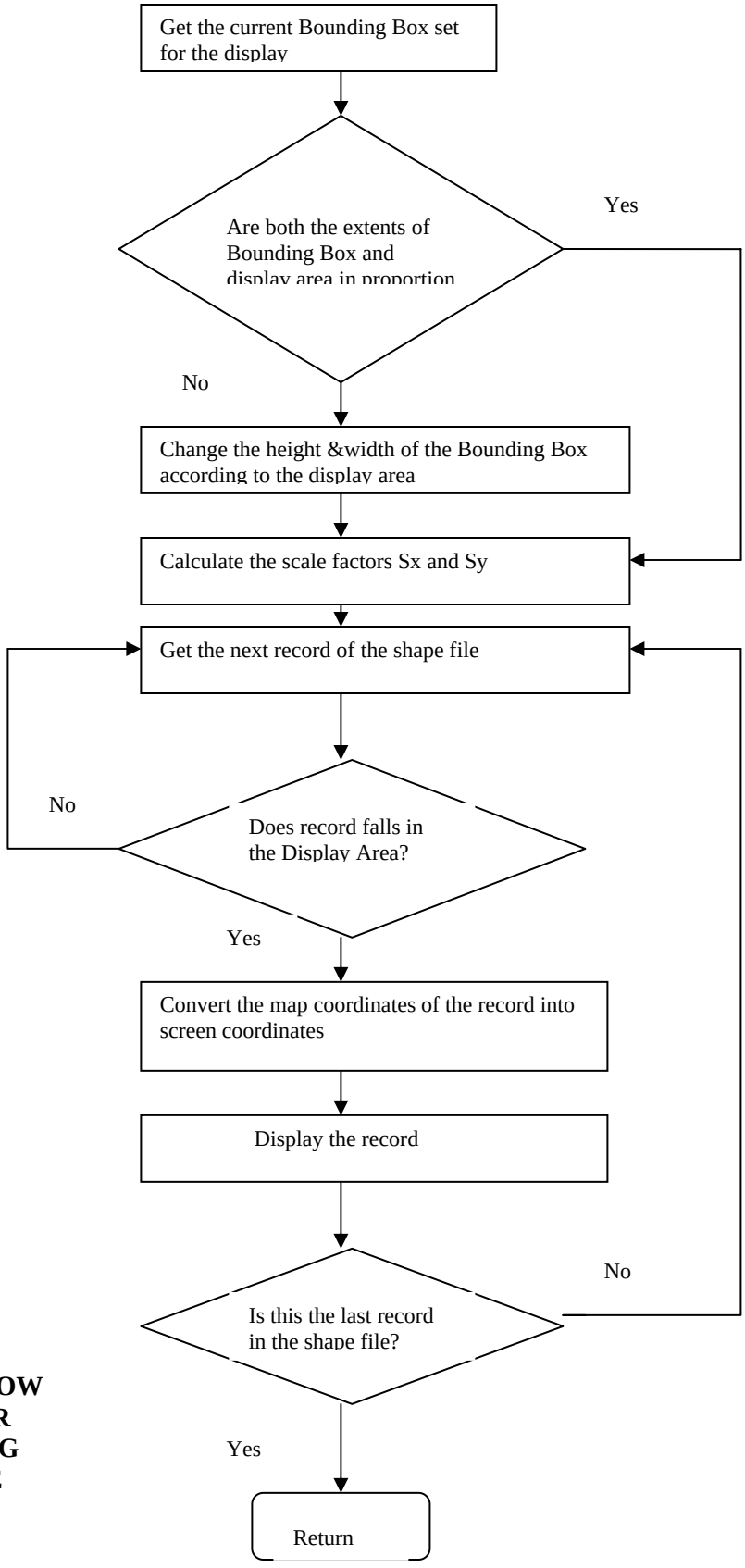

FIG:4.13 FLOW CHART FOR DISPLAYING SHAPEFILE

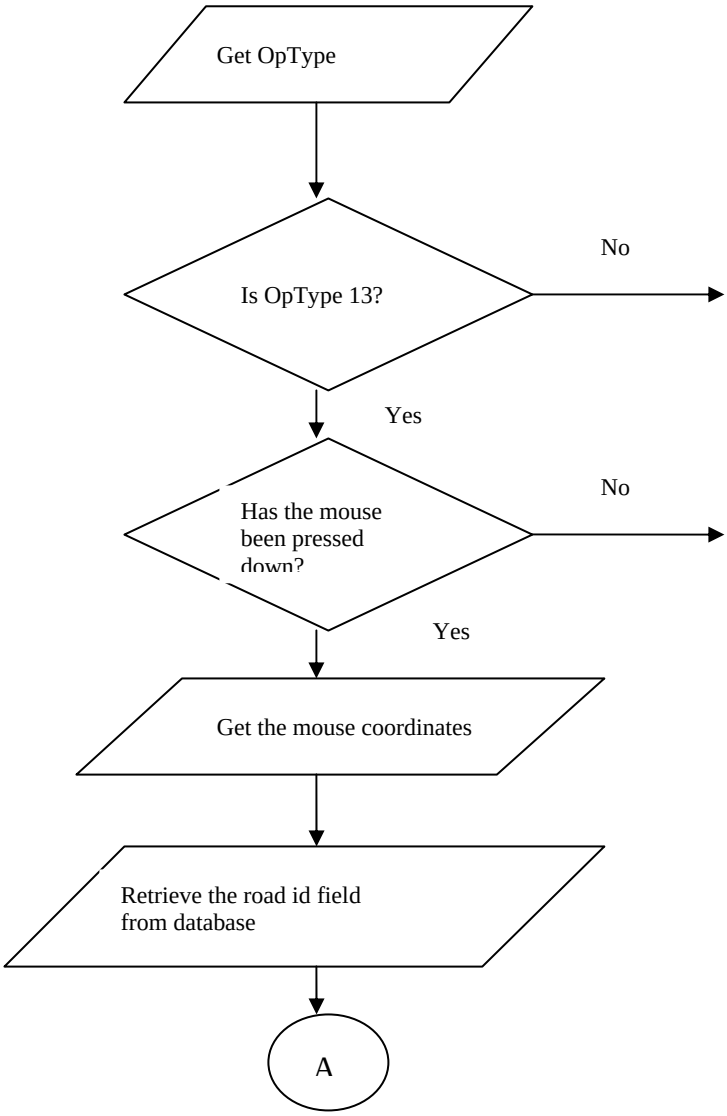

FIG:4.14 FLOW CHART FOR FINDING SHORTEST PATH

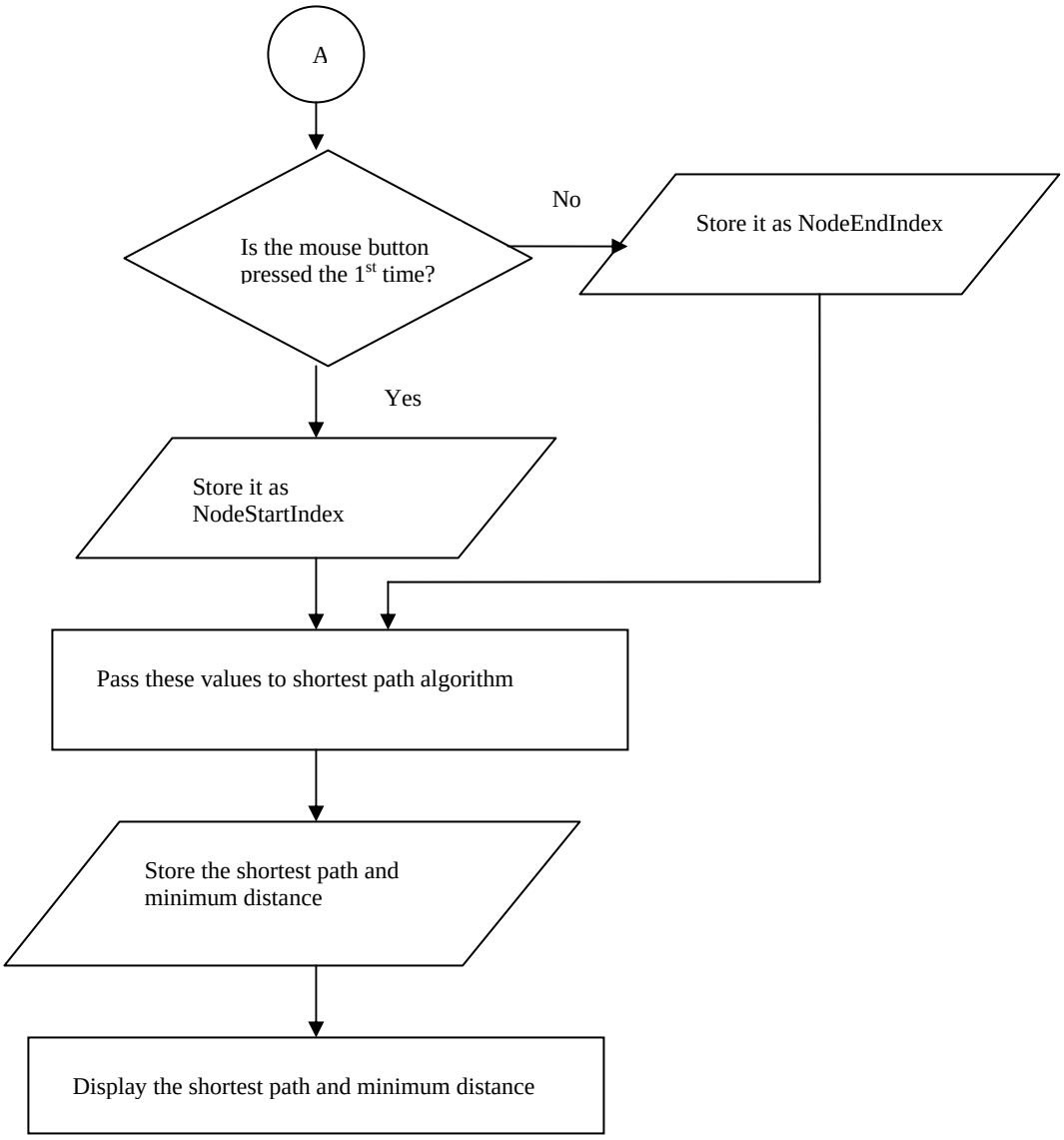

Fig: 4.15 Flow chart for finding the shortest path

4.5 OBJECT ORIENTED DESIGN

The object model describes the structure of objects in a system-their identity,their relationship to other objects, their attributes and their operations.

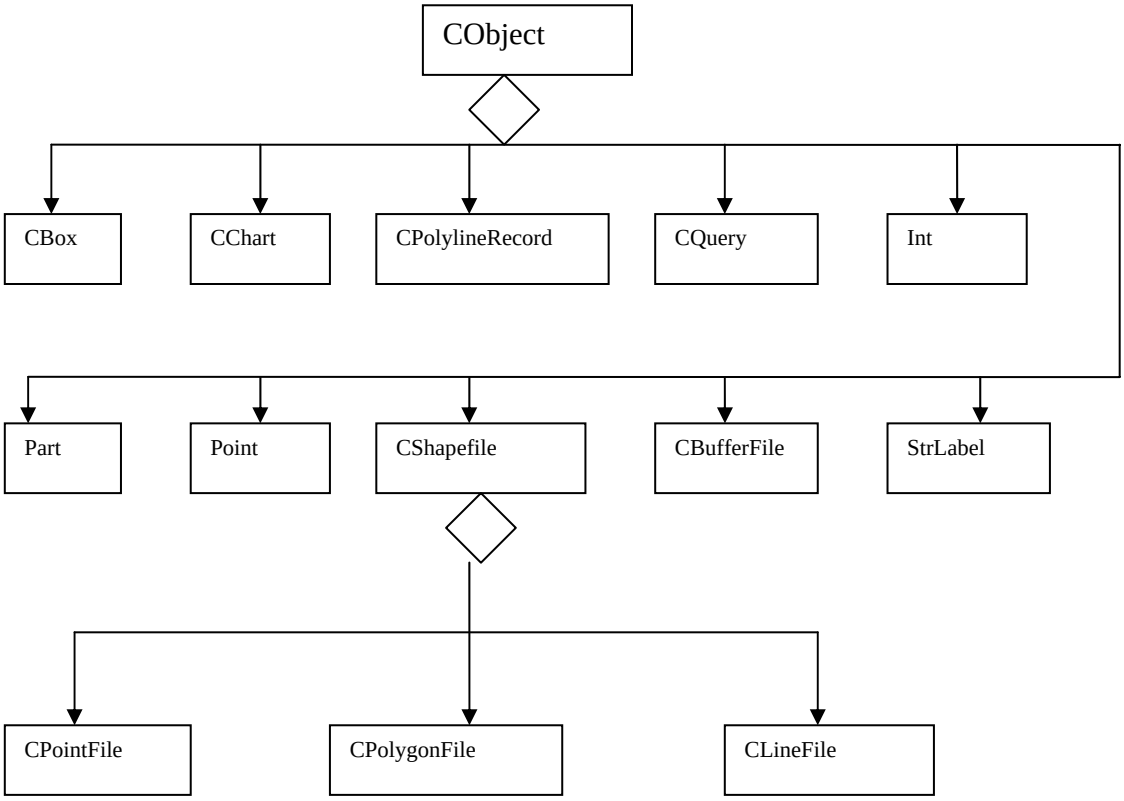

FIG: 4.16 OBJECT MODEL FOR ACTIVEX CONTROL

4.6 STRUCTURE ORIENTED DESIGN

Structural model shows how a function is realized by a number of other functions it calls. Structural charts are the graphical way to represent decomposition hierarchy.

Structured Charts

Structured Diagram steps

1) Identify system processing transformations
2) Identify input transformation.
3) Identify output transformation.

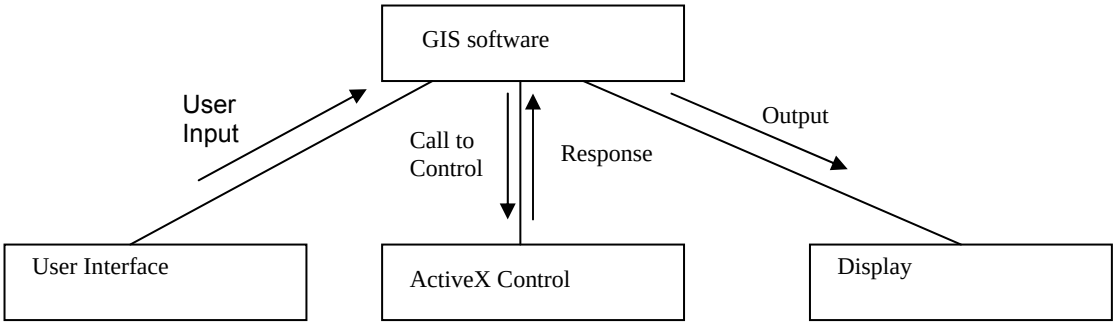

FIG: 4.17 INITIAL STRUCTURED CHART OF THE SYSTEM

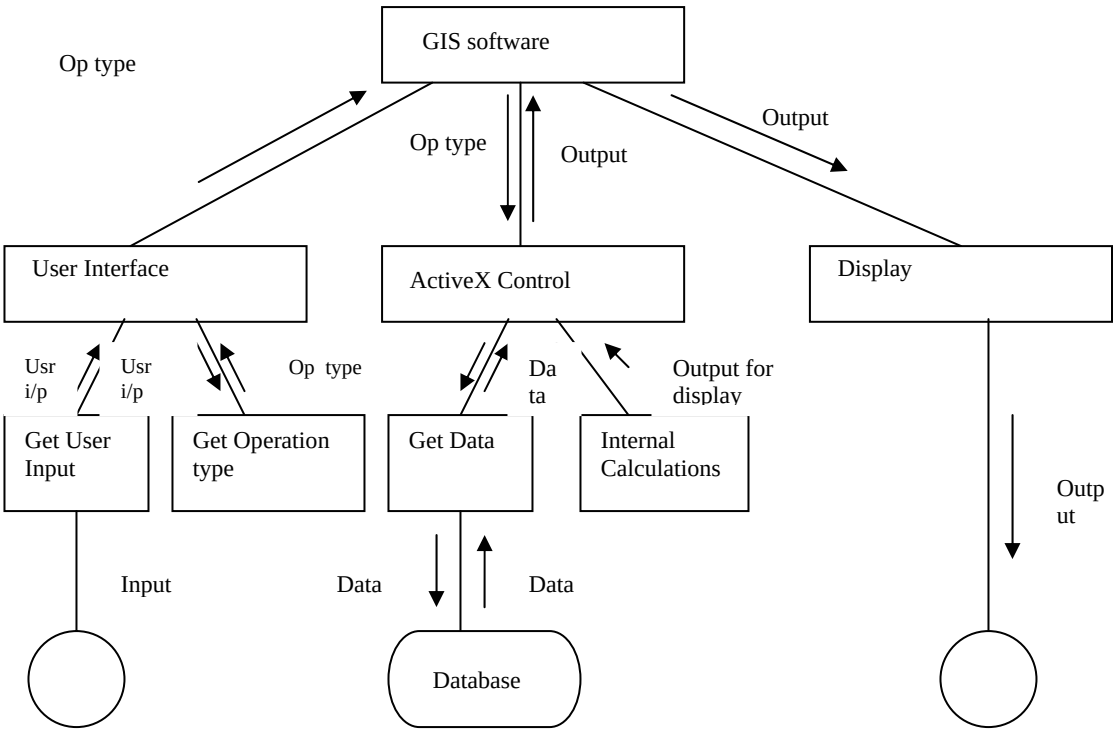

FIG 4.18 FINAL STRUCTURED CHART OF THE SYSTEM

4.7 DYNAMIC MODELING

State diagrams are used to represent dynamic modeling.

Conventions used in state diagram:

Event:

Event is something that happens at a point of time.

States:

The attribute values and links held by an object are called states.

Conventions used in state diagram are:-

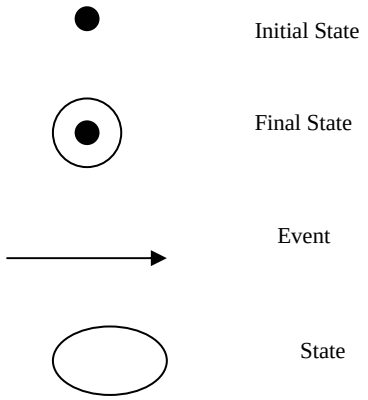

FIGURE 4.19 CONVENTIONS USED IN STATE DIAGRAM

State Diagram

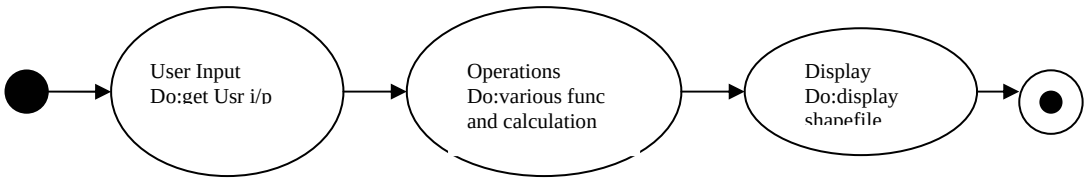

FIG: 4.20 STATE DIAGRAM FOR FINDING SHORTEST PATH

4.8 ENTITY RELATIONSHIP DIAGRAM

Any Business system description must start with their entities-people,places and things. These entities interact with each other in various ways and those

interactions are called Entity Relationship. In System Analysis entity relationship analysis is one of the conceptual modeling methods.

The ER data model uses a few basic concepts in producing ER diagrams.They are:-

Entity:

An entity is a person , place or thing in an enterprise eg customer ,an employee , a project, an order etc. Similar objects or things are grouped together into entity sets.

Relation:

A relationship[is meaningful association or linkage or connection between entities.

Attributes:

An attribute is any aspect, quality and characteristic, descriptor of either an entity or a relationship. An attribute tells what an entity is, entity has, entity contains or the entity does.

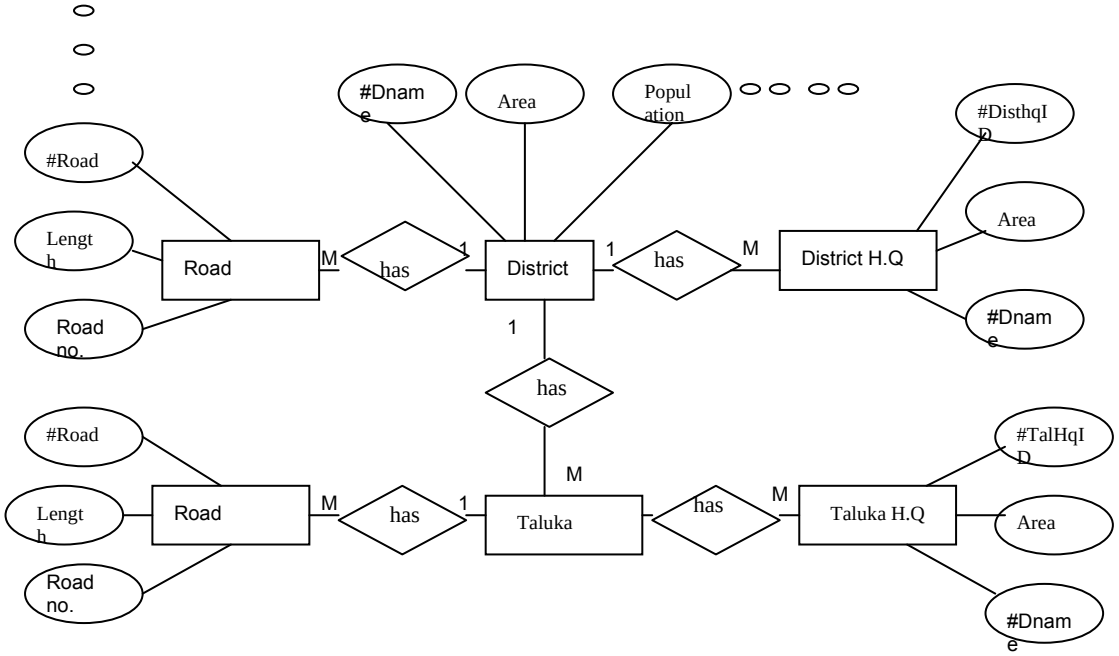

FIG:4.21 ENTITY RELATIONSHIP DIAGRAM

Chapter 5

FUNCTIONAL SPECIFICATION

- Functioning of Network Analyst
- Output

5.1 FUNCTIONING OF NETWORK ANALYST

Steps for finding shortest path include:-

1) Reading the shape file.

2) Selecting the source and destination nodes.

3) Conversion of the shapefile to a graph.

4) Applying shortest path algorithm.

1) Reading the Shapefile:

The road shapefile read and added as a layer in the network analyst and each layer is assigned a layer index.

2) Selecting the source and destination nodes:

Any point in the Shapefile can be selected as a source and a destination node.Each road in the shapefile is uniquely identified by it's road id and this value is stored when a particular source or destination node is selected.

In order to get the value of the node selected the following steps are performed:-

1)The index of the active layer is passed to a function which returns the CObject pointer element currently at that index.

2) The path of the file is stored alongwith the filename.

3) The particular point is located in the shapefile and the values of the bounding box are stored ,bounding box stores the region bounded by the shapefile.

4) The value of road id is retrieved from the database table and stored in a variable.

3) Conversion of the shapefile to a graph:

A graph is an Abstract Data Type (ADT) that consists of a set of nodes and a set of edges that estalish relationships(connections) between the nodes.In typical graph implementations nodes are implemented as structures or objects.

When a shapefile is added in the Network Analyst to find the best route road id's of each road is extracted from the database table.

Each road which is uniquely identified by it's road id is considered as a node of the graph.

From the database table distance or traffic parameter is extracted depending upon user's

Choice and that value is assigned as weight of the edge between the adjacent nodes.

4) Implementation of shortest path algorithm on a graph:

The aim of my project is to find shortest path between the points selected on a network(ie road shapefile which is added as a theme) .I have implemented Dijkstra's Algorithm for finding Shortest Path.

5.2 OUTPUT

Following snapshots show the output of Dijkstra's algorithm implemented on a Shapefile which is added in the Network analyst.

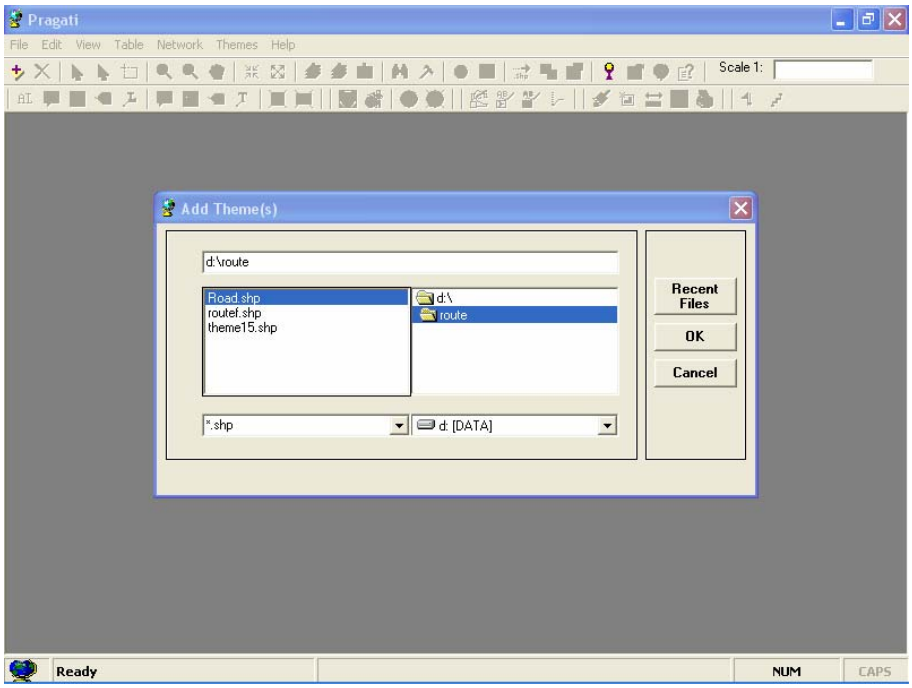

FIG:5.1 ROAD NETWORK FOR WHICH SHORTEST PATH IS TO BE CALCULATED IS ADDED IN THE NETWORK ANALYST.

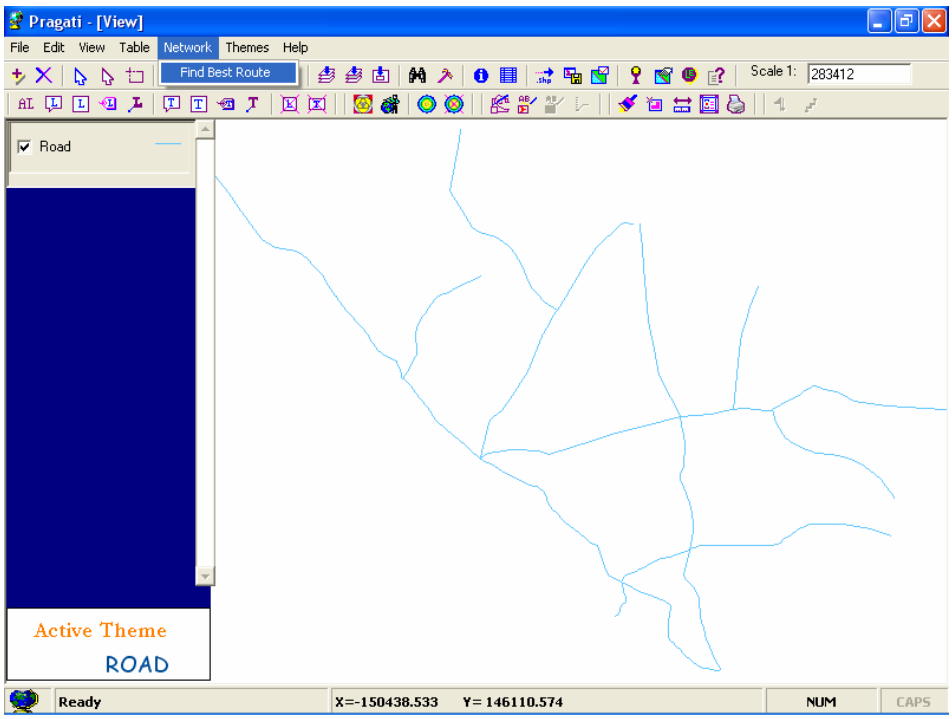

FIG 5.2 ROAD NETWORK SHAPEFILE.

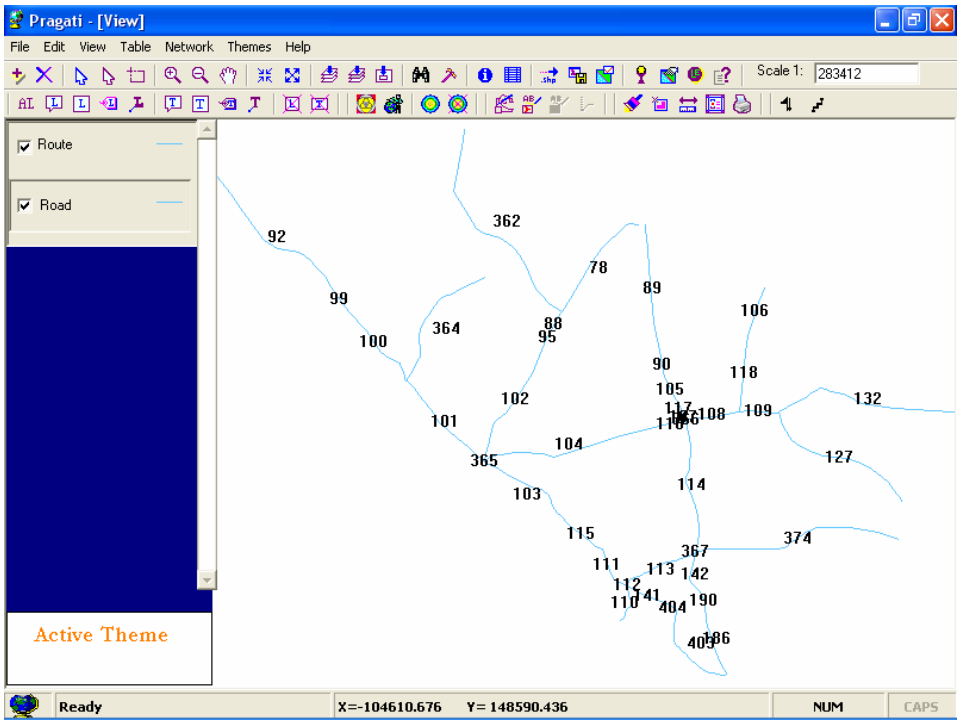

FIG:5.3 ROAD NETWORK DISPLAYING THE ROAD ID'S.

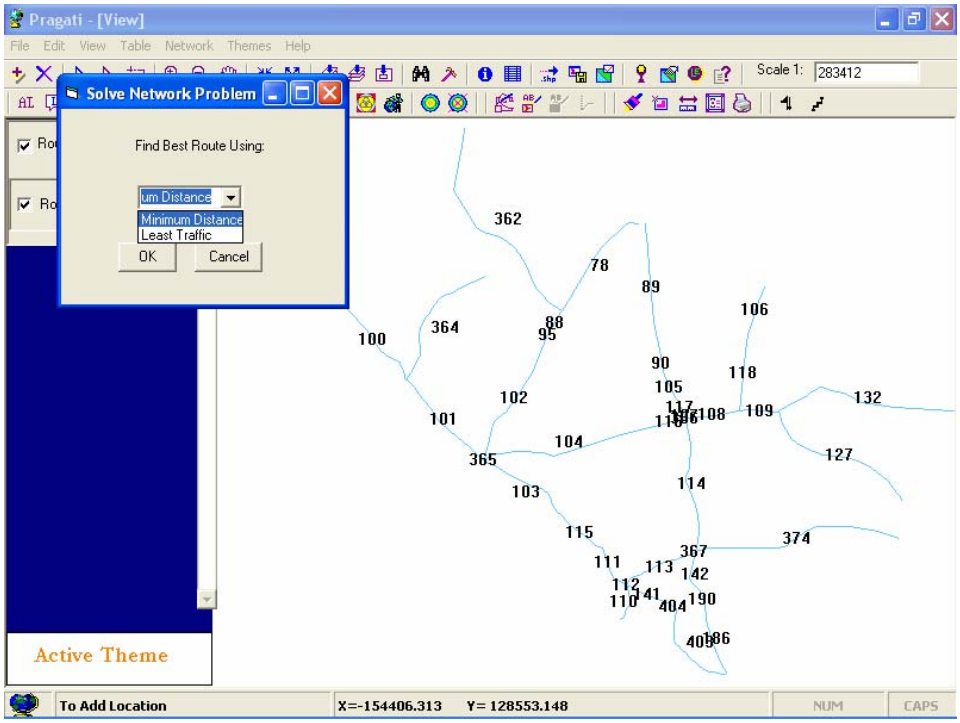

FIG 5.4 USER IS GIVEN CHOICE FOR FINDING THE BEST ROUTE.

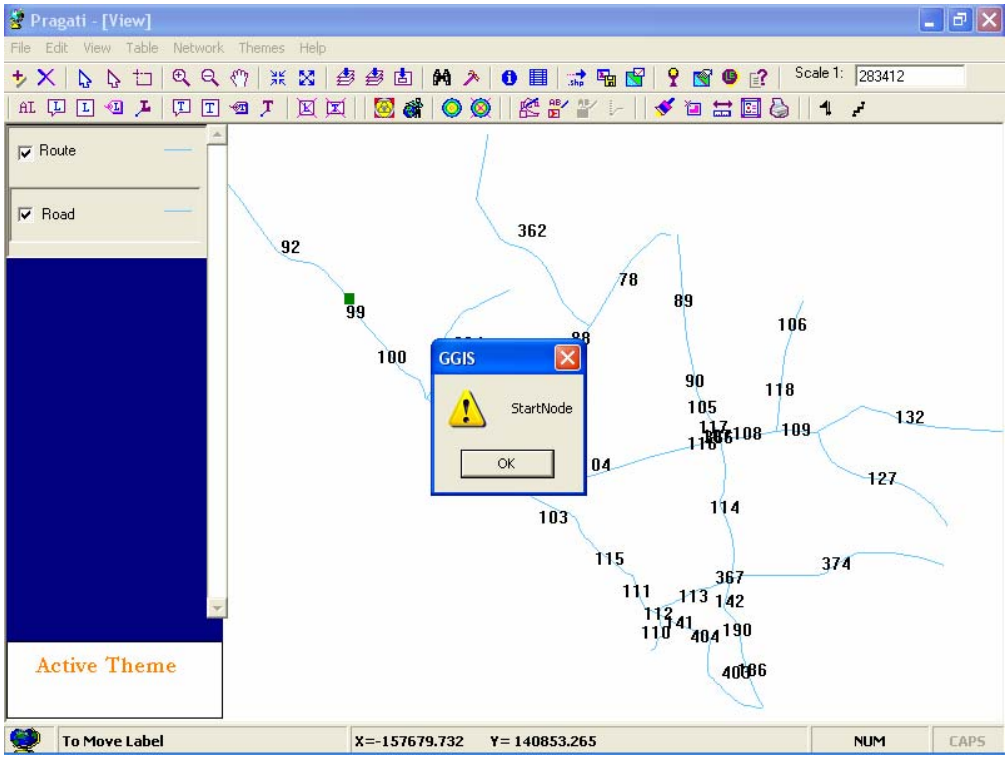

FIG 5.5 THE START NODE IS SELECTED AS THE ROAD WITH ID 99.

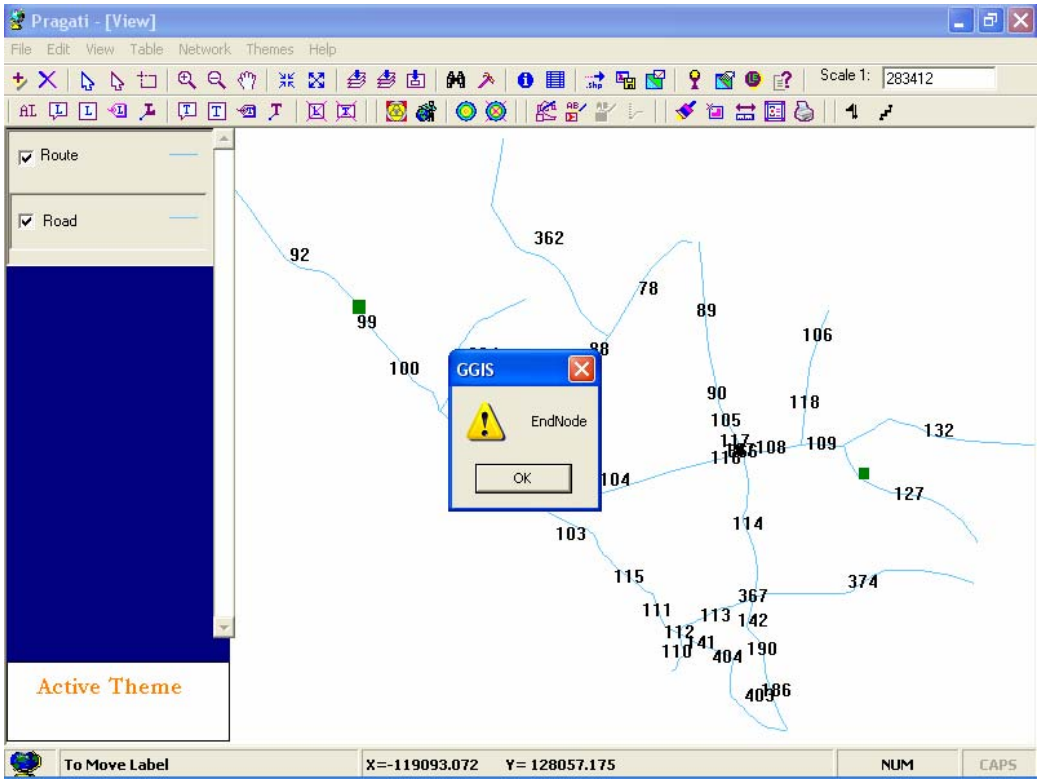

FIG 5.6 THE END NODE IS SELECTED AS THE ROAD WITH ID 127.

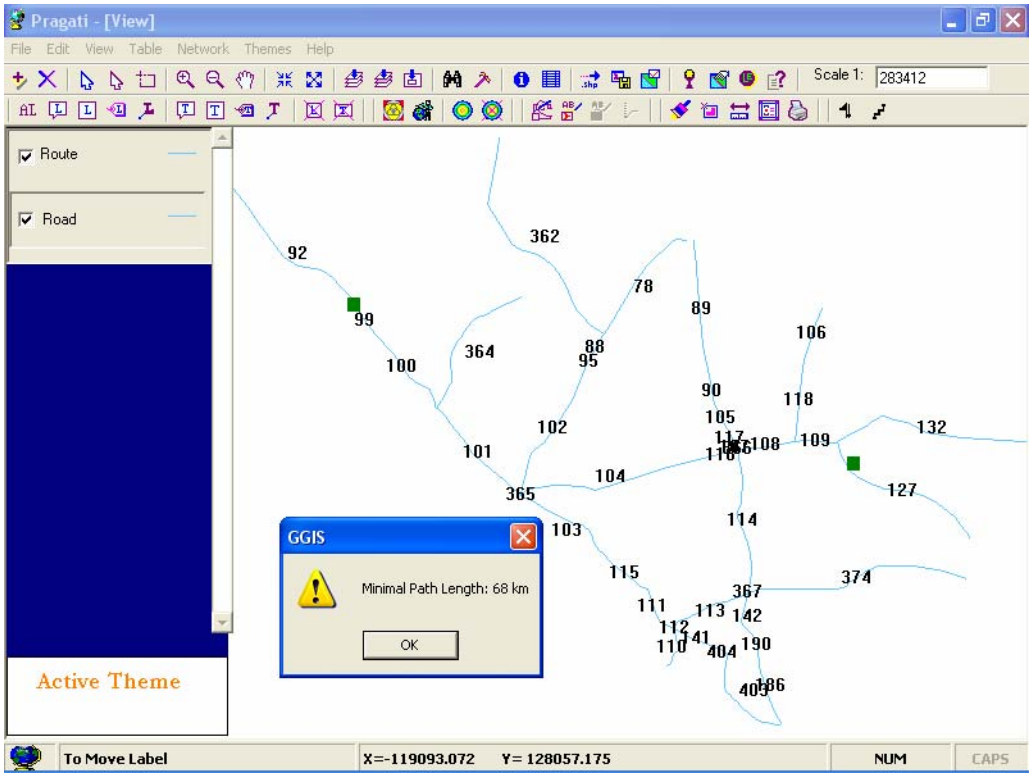

FIG 5.7 MINIMUM PATH LENGTH FOR THE BEST ROUTE IS 68 KM.

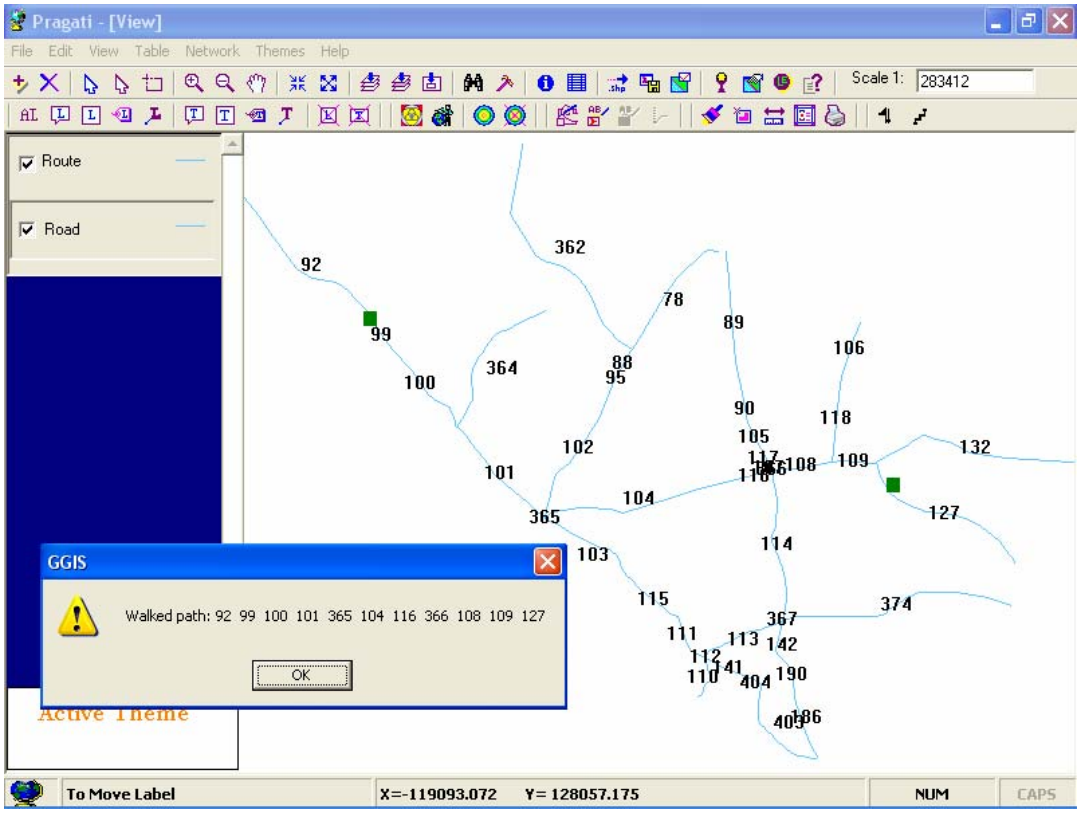

FIG 5.8 THE BEST ROUTE FROM ROAD WITH ID 99 TO THE ROAD WITH ID 127 TRAVERSES ROADS WITH ID'S 99,100,101,365,104,116,366,108,109,127.

Chapter 6

IMPLEMENTATION

- Member Variables
- Member Function
- Implementation Issues

6.1 MEMBER VARIABLES

The member variables used in DijkstraNodeStruct of CGGISCtrl are:-

NodeDistance – keeps track of the shortest distance of each node traversed so far.

NodePreviousIndex – keeps track of the previous index of each node in order to traverse the path.

6.2 MEMBER FUNCTIONS

1) Void CGGISCtrl::SetOpType(short OpType)

   Description: This function sets the operation type according to the button clicked.

2) Void CGGISCtrl::MouseDown(short Button, short Shift, long x, long y)

   Description: This function performs many tasks according to which button is pressed and which operation type is decided. If the operation type is set to 13 then on Mouse Down event the road id of the selected nodes is retrieved from the .dbf file and Dijkstra_FindPath function is called.

3) CString CShapeFile::AddLocation(LPCTSTR m_File,long sx, long sy)

   Description: This function extracts the road id of the node selected on the network.

4) Void CGGISCtrl::Dijkstra_FindPath(long NodeNumber,long NodeLinkMatrix[][],long &NodeStartIndex, long &NodeEndIndex,long &PathLengthMin,long &PathIndexnumber,long PathIndexArray[])

Description: This function first initializes the node distance and nodepreviousindex of every node to ∞ and then sets the node distance of StartNode as zero. After that all the adjacent nodes are traversed ,their distance and previous index is set according to the algorithm by making use of Dijkstra_Enqueue and Dijkstra_Dequeue functions and the Best route is found.

5) Void CGGISCtrl::Dijkstra_Enqueue(long &NodeIndex, long NodeDistance, long NodePreviousIndex)

   Description: This function maintains a list of nodes to be traversed and enqueues the nodes to be traversed.

6) Void CGGISCtrl::Dijkstra_Dequeue(long &NodeIndex,long NodeDistance,long NodePreviousIndex)

   Description:This function dequeues the node which has been traversed and makes the next node on the list to be traversed as the main node.

7) Void CGGISCtrl::AddLayer(LPCTSTR File, OLE_COLOR Color)

   Description: This function adds a layer .It reads file and brings it into memory. Shapefile is read in order to determine the type of object created for the shapefile. For a file this function must be called once. Record for the layers is updated once.

8) Void CGGISCtrl::RemoveLayer()

   Description: This function is used to remove active layers from memory.

9) Void CGGISCtrl::DisplayShape(CDC *pDC, CRect rcBounds)

Description: This function is used for displaying the shape. Here, initial scale is assigned after a new theme is added. It is done only for the first time. Here , conversion of map coordinates is done from meter to cm. The initial scale is calculated.

10) Void CGGISCtrl::DrawShapes(CDC *pdc, const CRect &rcBounds, const CRect &rcInvalid)

Description: This function is meant for drawing the shapes. Here , initial scale of the map is displayed for the first time.

6.3 IMPLEMENTATION ISSUES

Languages and tools

To implement, test and validate the efficiency of the algorithm concerning the Network Analysis we need to develop language skill in

-Visual Basic

-Visual Studio

We need to implement our research work using two different languages because we need to do two things

1) GUI is to be done in **Visual Basic** which has to be integrated in Pragati S/W
2) Coding is to be done in **Visual C++** which has to be integrated in GGIS control

So we use "Microsoft Visual Studio" for the implementation of research work in Visual Basic and Visual C++.

We found wide range of facilities provided by VC++ that would be of great help while we progressed further with the work.

Visual C++ 6.0 includes the comprehensive Microsoft foundation classes, which simplify and speed development of window applications. It includes sophisticated resource editor to design the complex dialog boxes, menu, toolbars, images, and many other elements of modern window application. There is an excellent integrated development environment called developer studio that present graphical views of your application's structure as you develop it.

Totally integrated debugging tool lets you examine in minute detail every aspect of your program as it runs.

Platform Dependencies :

Current Platform

I am implementing my work using the Windows 2000 Professional.

Windows 2000 Professional is suitable operating system for geographic system development and also for the use of the software related to the geographic system. Also it gives more update facilities as Microsoft's new Web-based resource site, automates driver and system file updates, and provides up-to-date technical support. Windows 2000 can review device drivers and system software on your computer, compare those findings with a master database on the Web, and then recommend and install updates specific to your computer. You can also revert to a previous device driver or system file using the uninstall option.

Dependency

As we use Microsoft Visual Basic and Microsoft VC++ ,our developed system is dependent on the "Windows" platform.

Constraints:

- Memory size
- Processor speed
- Maps are stored as shape file
- Dependency on the Windows platform

Shape file is a constraint as we can only use the information of the map which is stored as shape file to get the information from it and store the processed map in only shape file format.

So we can not use maps stored as image in other format like jpg , gif etc.

Chapter 7

MAINTENANCE

- Corrective Maintenance
- Adaptive Maintenance
- Perfective Maintenance
- Preventive Maintenance

7. MAINTENANCE

Once you develop a project a duty is not completed. Sometimes the application is unstable. It still works, but every time a change is attempted, unexpected and serious side effects occur. Yet the application must evolve.

The maintenance of the existing software can account for over 60 percent of all effort expended by a development organization, and the percentage continues to rise as more software is produced.

Much of the software we depend on today is on average 10 to 15 years old. Even when these programs were created using the best design and coding techniques known at the time, they were created when program size and storage space were principle concerns. They were then migrated to new platforms, adjusted for changes in machine and operating system technology and enhanced to meet new user needs all without enough regard to overall architecture.

The result is the poorly designed structures, poor coding, poor logic and poor documentation of the software systems we are now called on to keep running.

The ubiquitous nature of change underlies all software work. Change is inevitable when computer-based systems are built; therefore, we must develop mechanism for evaluating, controlling and making modifications.

"Software Maintenance" is, of course, far more than "fixing mistakes". We may define maintenance by describing four activities that are undertaken after a program after a program is released for use. We have defined four different activities:

7.1 Corrective maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct defects.

7.2 Adaptive maintenance

Over time, the original environment (e.g., CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate to its external environment.

7.3 Perfective maintenance

As software is used, the customer/user will recognize additional functions that will provide benefit. Perfective maintenance extends the software beyond its functional requirements.

7.4 Preventive maintenance

Computer software deteriorates due to change, and because of this, preventive maintenance must be conducted to enable the software to serve the needs of its end users. In essence, preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted, and enhanced.

Only about 20percent of all maintenance work is spent "fixing maintenance". The remaining 80 percent is spent adapting existing systems to changes in their external environment, making enhancements requested by users, and reengineering an application for future use.

Chapter 8

CONCLUSION AND SCOPE OF FUTURE WORK

- Conclusion
- Scope of future work

8.1 CONCLUSION

The Network Analyst finds shortest path and minimum distance between places and is helpful in solving many problems like it helps to find

1) The bus route to be followed in order to reach a place.
2) The emergency route in case of disaster.
3) The minimum distance to reach a hospital so that decisions about the path to be followed can be taken in case of emergency treatment.
4) The Best path and minimum distance to reach an education department, Community Hall ,Bank ,Police Station etc.

8.2 SCOPE OF FUTURE WORK

Arc view Network Analyst 3.2a solves a variety of problems based on geographic networks. It solve problems such as finding the most efficient travel route across town, finding the closest facility like Dispatch the closest vehicle to a location, or find the nearest valve in critical situations.

I have worked on solving the shortest path problem and developed an algorithm for it, work can be extended to locate the closest facility and added into Pragati.

REFERENCES

[1] Pal A.Longley,Michael F. Goodchild,David J.Maguire,David W.Rhind,*Geographic Information Systems and Science,*2005.

[2] Michael Workboys and Matt Duckham, *GIS –A Computing Perspective* ,Second edition,CRC Press.

[3]Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest, Clifford Stein, *Introduction to Algorithms,* Second Edition ,Prentice Hall India.

[4] Chris H. Pappas and William H. Murray III, *The complete reference Visual C++*,TMH publication,2003

[5] Jonathan Bates and Timothy Tompkins, *Practical Visual C++ 6*,Prentice-Hall of India,1999.

[6] Ivor Horton's , *Beginning Visual C++6*,Wrox Press Ltd, Third edition,2003

[7] Michael J. Young , *Mastering Visual C++ 6*, First Edition,1998.

[8] Julia Case Bradley, Anita C.Millspaugh , *Programming in Visual Basic 6.0*,TMH Publication,2000.

[9] Steve Brown , *Visual Basic 6.0 in Record time*, bpb Publication,1998.

[10] Introduction to GIS, User's Guide,ESRI.

[11] ESRI Shapefile Technical Description, An ESRI white paper, July 1998.

[12] MSDN Library July 2000.

[13] [http://www.bisag.gujarat.gov.in/]

[14] [http://www.esri.com/]

[15] [http://www.programmerworld.netfirms.com/books/visual_c.htm]

[16] [http://www.programmerworld.netfirms.com/books/visual_basic.htm]