The development of a Geographic Information System for traffic route planning using Location Based Services

by

Matthew Pulis

Supervisors:

Dr. Maria Attard

Mr. Joseph Vella

Department of Computer Information System

University of Malta

September 2008

A dissertation submitted to the Faculty of Science, University of Malta, in fulfilment of the requirements for the degree of Master in Science.

Declaration

Plagiarism is defined as "the unacknowledged use, as one's own work, of work of another person, whether or not such work has been published" (Regulations Governing Conduct at Examinations, 1997, Regulation 1 (viii), University of Malta).

I the undersigned, declare that the thesis submitted is my work, except where acknowledged and referenced.

I understand that the penalties for making a false declaration may include, but are not limited to, loss of marks; cancellation of examination results; enforced suspension of studies; or expulsion from the degree programme.

____

Matthew Pulis

September 2008

Abstract

Studies suggest that a large percentage of traffic congestion occurs during peak hours (Downs, 2004). Many location based services and systems today exist, that assist traffic routing to maximise on travel time and reduce trip lengths (e.g. Garmin, TomTom). The objective of this study looks at enhancing Open Source routing algorithms by adding functionalities and variables that favour fuel savings as well as shortest route. A system is developed using Open Source GIS (using the PostGIS and pgRouting extensions for the DBMS: PostgreSQL), whereby a number of variables, including weather conditions, road network, traffic status, vehicle specification data and road gradient are manipulated by the user to achieve shortest and most fuel efficient route. This study also provides neighbouring techniques for emergency response teams, so as to traverse the shortest route to an accident.

The project started with a Literature Review studying the historic advancements of Location Based Services and Geographic Information Systems, in particular Open Source GIS. Case Studies were reviewed so as to gain knowledge from past experiences. The methodology used for this project followed the DSDM methodology and requirements were drawn following the MoSCoW priorities. A full working version of the project which is presented in a Web Interface can be accessed online.

Dedicated to all Open Source developers

Acronyms

ACSM	American Congress on Surveying and Mapping
AGI	Association for Geographic Information
A-GPS	Assisted Global Positioning System
CID	Cell Identity
DBA	Database Administrator
DML	Data Manipulation Language
DSS	Decision Support System
ESRI	Environmental Systems Research Institute
ETA	Estimated Time Of Arrival
FCC	Federal Communications Control
GIS	Geographic Information System
GiST	Generalized Search Tree
GIS-T	Geographic Information System applied to Transport
GML	Geography Mark-up Language
GPRS	General Packet Radio Service
GPS	Global Positioning System
IEEE	Institute Of Electrical And Electronics Engineers
ITS	Intelligent Transportation System
IVHS	Intelligent Vehicle Highway Systems
LADS	Location Aware Devices and Services
LAN	Local Area Network
Lat/Lon	Latitude/Longitude
LBS	Location Based Services
MEPA	Malta Environment Planning Authority
NAD	North American Datum
ODBC	Open Database Connectivity
ODC	Open Data Consortium
OGC	Open Geospatial Consortium
OS	Operating System; Open Source; Ordnance Survey (U.K.)
PGSQL	PostgreSQL
PHP	Hypertext PreProcessor
PL/pgSQL	Procedural Language/PostgreSQL Structured Query Language
PostGIS	PostgreSQL Geographic Information Systems
RDBMS	Relational Database Management Systems
SQL	Structured Query Language
SSH	Secure Shell
TDOA	Time Difference of Arrival
TIGER	Topologically Integrated Geographic Encoding and Referencing
TOA	Time of Arrival
URL	Uniform Resource Locator
WGS	World Geodetic System
XML	eXtensible Mark-up Language

0T**ACKNOWLEDGMENTS**

I would like to express the deepest appreciation to the following people and acknowledge their help as it has been a crucial part in the making of this work:

My advisors, Dr. Maria Attard and Mr. Joseph Vella for their continuous guidance, patience, mentoring and support throughout the thesis. Learning to do research in Computer Information Systems was one of my goals when I decided to pursue graduate studies. Much of this work was carried out in close collaboration with my advisors.

The various people who helped me whenever I had a problem on the various mailing lists and IRC channels.

My best friends Karl Farrugia and Karl Spiteri for giving me courage whenever I really needed it.

My cousins Odette, Adrian and Ivan for their technical help.

My parents and my sister for their continuous support throughout the thesis.

My grandmother whose never-ending support helps me persevere in every task I undertake.

Matthew Pulis

September 2008

Contents

List of Figures

List of Tables

Chapter 1 : Introduction

1 Introduction

1.1 Background

The area of study of this dissertation is route planning using geographic information systems. Studies show that a large percentage of traffic congestion is due to peak-hour travel (Texas Transportation Institute, 2005). The approach used in this study demonstrates one method of how traffic congestion can be mitigated by managing demand and routing of traffic. The method applied to tackle the problem is better use of urban intersections, in situations where this is necessary. By adapting a route to each individual vehicle on the network, it is believed that a degree of load balancing would be achieved.

The main concern is to propose a study which provides an easy interface for the user and enabling the system to propose route planning as per user requirements. Consideration to the environment in which the system is running is also an important factor for the system. Therefore the system needs to respond to the variables which keep changing over time. The main variables, which shall be studied in the research, are weather conditions, and road impediments. Information Technology is an important factor in today's life, especially when it used to make life easier for the general public. The system should provide the day to day running of road network in a studied geographic area. One can view the system as a value-added product to the road network, since it enhances on the already available route planning methods available on the market. In a nutshell, the extra advantages gained from the proposed system would be that variables are taken into consideration before being included in the response.

1.2 Research Objectives

The objectives were established after thorough research into the main problems identified in the current routing applications available on the market. Unfortunately, many routing applications provide a routing mechanism which often resides on pre-computed routes, and little to none base the route computation on live data. Due to personal inclination towards Open Source projects, it was decided that this project tackles the problem by using only Open Source software. The main aim of this is to enhance Open Source projects so as to maximise the benefit of using both live data and also user preferences. Different users may require different routes, and not always the shortest one. Thus the idea is to enable routing on geographic information systems based on the users' requirements. Another objective is to have a system aware of the variables that normal vehicles can find in their route. Apart from a mere point to point route planner, the system needs to have a way to incorporate other variables when the route is being planned. These variables are:

- Time: This variable is used in case the route should be planned on arriving at the destination as quickly as possible; thus, try to take high speed roads. By making use of shortest distance in combination with a highway hierarchy, thus combining the actual flow of traffic to the speed limit, it will result in arriving at the shortest time possible.
- Consumption: This variable is used in case the best route should be planned on consuming the least fuel possible.
- Shortest Distance: This variable is used in case the route should be planned on arriving at the finishing point covering the shortest distance on the network as possible.

Depending on the time of the day, peak consumption of road network is set. Also arriving at the destination is affected if there are road works or any other type of blocking, including but not limited to traffic accidents, road blocks and weather conditions, all of which would form part of the variables taken into consideration.

What differs in this research when compared to what other systems is that the proposed system is information technology oriented, thus emphasis is given to sharing and using the most recent data possible. The project is aimed to capture data from various feeds and use it in on-line so as to update the road network. This makes the research also market driven, since the clients are the main benefiters from such updated routes, by optimising their driving route. Another advantage is gained by emergency response teams since such software will propose the shortest route to attend to an accident.

The research objectives are therefore:

- To design and build a Geographic Information System that will act as the backend for the system offering these services. One must note that computation limitations exist.

- To allow live data being fed directly to the system. This objective will be possible by enabling the network to ping different sources, be it road network monitors and weather stations, to allow for new data feeds. This will make sure that route planning is always computed on the latest data available.
- To build a Location Based Service for mobile customers which act as an intelligent route planner. Such a route planner allows the user to specify which criteria is the most attractive to be used depending on the main variables in the system (time, consumption and distance) and build the Information System around these variables.
- To identify the main variables in the system (time, consumption and distance) and build the Information System around these variables.
- To identify the main objects in the system, thus which vehicles form part of the system and how these are grouped.
- To design and build a Web-Interface front end for the users to communicate with the backend. Such a Web-Interface should be accessible from mobile devices for the route planning exercise. Another front-end using Open Source readymade software will be used to allow Street Management in case congestion is reported on a street. One must note here that Congestion reports will be thought of as being reported by third parties and the Street Management interface should just update the current status of the road.
- To provide an updated route every fixed time interval, minimising the risk of driving through a congested road.
- Once the project is done, it is aimed also to propose suggestions for useful additions and improvements.

Further details about the requirements, can be found in Appendix I – Software Requirements.

1.3 Introducing the terms

In this section a brief high level introduction of the main terms which the document is to use is going to be given. The main terms under investigation are Geographic Information Systems (GIS), Location Based Services (LBS) and Vehicle Routing. More detail can be found later on in Chapter 2.

Geographic Information Systems

A GIS can be regarded as a combination of “*hardware, software, and data for capturing, managing, analyzing, and displaying all forms of geographically referenced information”* (ESRI, 2007). When compared to maps, GIS has the advantage that data presentation and storage are separate (Bernhardsen, 2002). This results in having data being presented in several ways such as zooming on a part, panning, make distance calculations and request other non-spatial data. Such an IS can be used for resource management, vehicle routing, environment impacts assessment, geography planning to name a few.

Location Based Services

One of the best services an organisation can to do to mobility is to offer a service based on the location (MobileIn, 2004). Such a service is a personalised service to the user, and unique due to the current location the user is at. A sample LBS would be giving the nearest restaurants within a certain proximity to the user. Making use of such services can “*save time and aggravation in our increasingly wireless world*” (Capella & Bennett, 2002). Thus LBS can be seen as applications that use the user’s physical location to provide a custom service.

Vehicle Routing

The last years have seen an increasing utilisation of algorithms to provide routing for vehicles. Such routing practices can be used in vehicles to provide the route where there driver needs assistance and also to provide the best route to cover all points in a map which need to be visited (Travel Sales Person problem). By applying GIS to such a problem vehicle routing can be simplified (Marmar, 1999). A GIS can be used in scenarios where Vehicle Tracking, Dispatching, Routing and Scheduling need to be done.

1.4 Summary

In this chapter, the main requirements and software used to tackle the problem were discussed. In the following chapter, the main theoretic parts involved in this study are going to be covered, thus allowing the reader to be further familiarised with the project scope. Literature regarding Location Based Services and Geographic Information Systems will be discussed and compared thus giving a better idea where the technology stands at the moment.

Chapter 2 : Location Based Services

2 Location Based Services

2.1 Introduction

This chapter is aimed at introducing what Location Based Services are and how they can be used to provide an added value to their users. An overview of different Location Based Services is given, followed by a detailed analysis of Geographic Information Systems, techniques and scenario where they were used. Finally a number of existing Location Based Services are discussed and an outline how actual Geographic Information System implementation enhanced the added service.

2.2 Definition of Location Based Services

Location Based Services or as sometimes they are referred to as Location Specific Services, are considered by many as one of the leading driving force of Mobile Commerce (m-commerce). Such services are normally based on the clients' locations, hence the name Location Based (Schiller & Voissard, 2004). The geographic position can be acquired using many techniques, naming but not limiting, the GPS format. According to (Research and Markets, 2006), by 2010 it is estimated that revenues generated from Location Based Services will reach €622 million as opposed to 2005 which generated €144 million, thus an increase of more than 330 per cent. In another research, it is thought that by 2012, Europe and Asia will be the leading continents in Location Based Services (ABI Research, 2008). It is expected that in Europe more than 100 million users will be making use of LBS by 2012 (Berg Insight AB, 2008). Counting such increase in figures, with the more than 185 million Asian foreseen users, will amount to the budgeted figure of more than 13 Billion US Dollars (ABI Research, 2008). As one can see from both Figure 1 and Figure 2, the percentage of Location Based Services are increasing every year. The high increase in expected expenditure in the coming years is due to a high increase in the Asian markets (Figure 2).

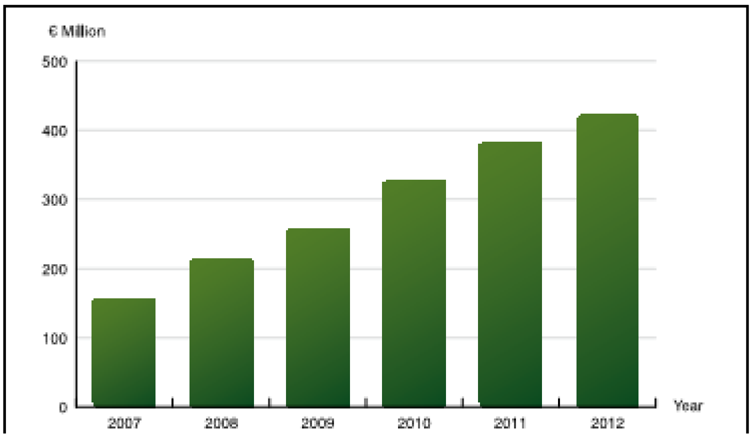


Figure 1 : Location Based Services Forecasted Revenues in Europe (Berg Insight AB, 2008)

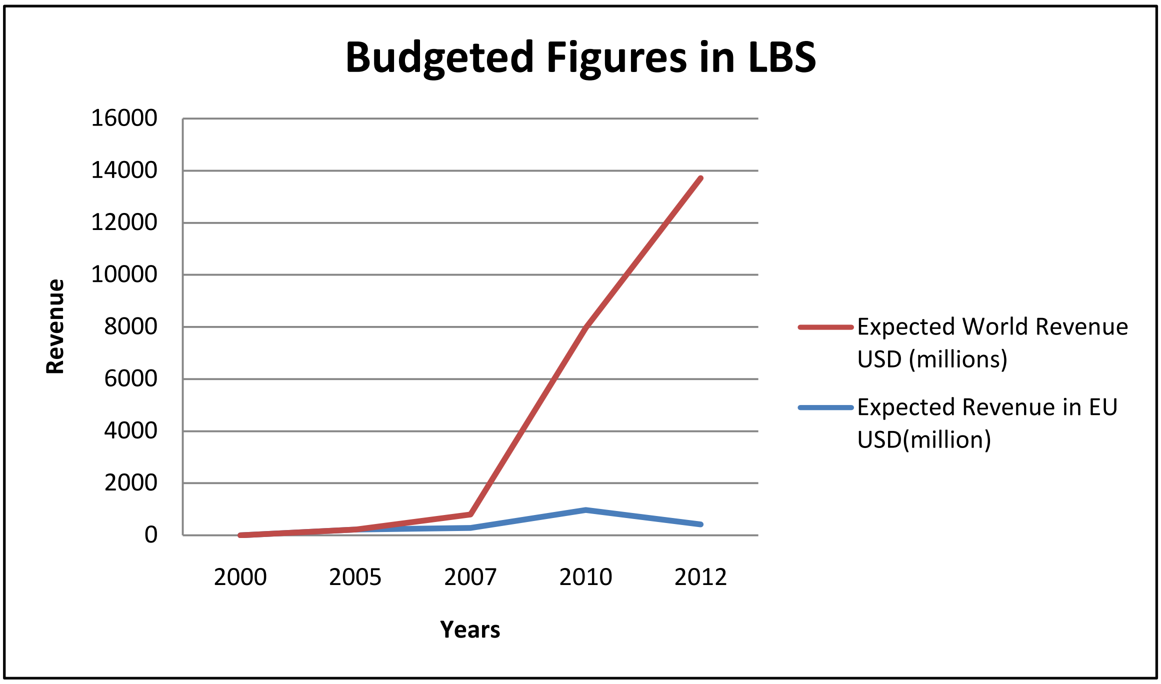


Figure 2 : Projected European and Worldwide Revenue figures for LBS (Winter & Goasduff, 2008) (ABI Research, 2008) (Berg Insight AB, 2008)

2.3 Benefits to the Network Operator

Network Operators, are the organisations, profit and non-profit, which offer the LBS as an added value to the services they usually offer to their clientele. In this section, the advantages and benefits to the Network Operator offering Location Based Services will be discussed. Just to mention an example of a Network Operator which is offering many Location Based Services is T-Mobile, including, route planning services (T-Mobile, 2003) and Child location service which notify the parents of the whereabouts of their kids (T-Mobile, 2008).

2.3.1 Business Benefits

Such higher pricing can easily be used to recoup expenses made for infrastructure and allow the Network Operator better profits, whilst enabling the entity to promote itself in the market enabling a marketing superiority and also a better marketing position when compared to its competitors. Such services allow the Network Operator to retain more loyal customers (Ververidis & Polyzos, 2002).

2.3.2 Increase Average Revenue per Annum

By offering premium value added services, such as Location Based Services, the Network Operator can reduce churn and increase the revenue. For every request / response sent to the client who is making use of Location Based Services, revenue is generated to the Network Operator, since many times it is billed on bandwidth traffic. Differentiating applications, based on target markets, such as youths or organizations using the "Travelling Sales Man" will facilitate premium pricing. By "Travelling Salesman" problem it is understood visiting different nodes at the least cost, and visiting each only once and then returning back to the starting node.

2.3.3 A value added service

Location Based Services are not limited to organisations to enhance their profits, but can also be offered by non-profit organisations. States in USA, Canada and in Japan have introduced different exercises, such as the E-911 service in USA (Federal Communications Commision, 1999), Canada introducing free route directions via two web portals 'Mobile Location Based Services Network'[1] and 'TravelGIS Driving Directions Service'[2] (Zhou, 2003) Tokyo experimented with sending highly tailored information to users based on location (Srivastava, 2004). Public transport could also greatly benefit from Location Based Services. Commuters could be informed of the likely arrival time of the next buses / Metro / Train and also notified of any delays or deviation from schedules that may have been encountered.

[1] Mobile Location Based Services Network (http://mlbs.net/demo/)
[2] TravelGIS Driving Directions Service (http://www.travelgis.com/directions/)

2.3.4 Increase in mobile marketing

Another usage of Location Based Services is to direct the appropriate advertising to appropriate persons which can lead to increase sales. An example of such marketing can be a shop, situated on famous avenues like Champs Elysees which can send out text messages with coupon codes to people approaching the shop. Another example can be attaching RF-IDs to items in a supermarket, and whilst shopping, a reminder to buy complementary goods is raised. Such marketing techniques have only the boundaries of one's imagination as a limit, and obviously costs. Many mobile users receive wide-sent messages with advertising, in a market which is seen by some as the "it" word in the telecoms industry. Informa Telecoms & Media forecast, that global mobile advertising market is going to be worth $871 million in 2007, and will jump to $11.35 billion in 2011, thus more than 1000 per cent in just 4 years (Knox, 2007). In Japan only, which is regarded as the second biggest advertising market post US, and the first country to exceed the 50 per cent 3G penetration, estimates are that mobile advertising revenues are to reach $600 million in 2009 (Knox, 2007). Another interesting figure is that almost 60 per cent of Japanese consumers already avail themselves of mobile coupons and discounts more than once a month (Knox, 2007). Western Europe is indicated to reach such high penetration, 60 per cent, in 2010 (Hang, 2006), seeing Italy and UK on the high ranks of such penetration. Following such increases around the world, one should also increase the view with respect to Location Based Services.

2.4 Benefits to the client

The users are the pushing factor of any Location Based Service. This means that any LBS have to be built around the user in order to make sure that an added service is enjoyed. Such services must also make sure that information is given only on demand. This means that no forced advertisings should be used with any LBS offered.

Location Based Services can be useful not only for routing queries, but also for socializing with peers, amongst other diverse applications. Many services associated with Location Based Services, are requested on demand, thus the clients need not pay for the service on monthly or renting bases, but use and thereafter pay for the service usage. The service offered can be personalised for the customer's need, thus providing a better service.

Business users thus supported by commercial organisations benefit from such value added services too. In the case of a company which has a fleet to manage will truly benefit from these services since there can be knowledge where each person is, or even planning the best route to take (using techniques such as the Travelling Sales Person algorithms) enhances the product and service offered.

Another benefit would be seen into the Emergency Response teams, where a call for aid will also carry with it its location, to help the ER teams to provide faster location of the incident. The Federal Communications Commission (USA) has embarked on the e911 project in 1996, which is said to ask from the carriers to provide location within 50 to 300 meters accuracy, depending on the type of technology used (FCC, 2007). Unfortunately reports show that such initiative is facing difficulties in being implemented due to reluctance from major mobile manufacturers claiming that the new chips involved would be too expensive, and there is no added value (Leite & Pereira, 2006). The EU commision followed this step and planned 1st January 2003 as the target date for such service in all Member States (EU Commission, 1999), but until today there is no proper working mechanism in the EU, despite its efforts in trying to pressure the member states.

According to Mr. Jesmond Camilleri (2007), Malta has embarked on setting up a Task Force to deal with such a service. The principal objective of the Task Force is to analyze the present situation in Malta and propose techniques to enrich the service. The authorities are not ruling out any option, and are open to any opinion. Following the response from Mr. Camilleri, it was decided to contact the Ministry of Justice, to ask further information. Mr. Charles Deguara, Permanent Secretary in the Ministry, has confirmed that Malta already has its own collaboration between the Police Force (who is responsible for the operational management of 112 in Malta) and the two main cellular providers, thus Go and Vodafone, when the exact caller location is needed. According to Deguara (2007) (2007), the Task Force is projected to provide the Government with relevant information about the current situation with regard to emergency response, and present "realistic and doable" proposals which can enhance the service offered. Also the authorities are not limiting themselves on the financing on the project, according to Camilleri (2007) (Camilleri, 2007).

2.5 The Importance of finding the location

Location Based Services, as the name itself suggest, allow for the capability to find the geographical location of the user and provide with the service requested. Based on the current location of the user, the service is altered to meet the user's needs. Therefore, one of the main targets of an organisation embarking on an LBS project should be to acquire the current position of the user, so as to provide the service against that position. In this section different techniques which are used to capture one's location are to be discussed, each useful for particular scenarios and carry a cost and benefit ratio.

2.5.1 Global Positioning System (GPS)

The terms global positioning, quickly bring into mind the famous GPS (Global Positioning System). This system works by availing of a GPS transceiver (transmitter and receiver) resulting with a location. GPS is a worldwide radio navigation system consisting of 24 satellites, having a minimum of five visible from any spot on Earth (Shukla & Magon, 2006). Such technology is based on the concept of Trilateration (Shukla & Magon, 2006), which is a basic geometric principle stating that you can find the location of a point if you know the distance from other known locations. For Trilateration to succeed one just needs three signals, and calculates the point by receiving the signals from the satellites, and checks the time taken by the signal to be delivered. According to Wolf et al (1999), GPS results allow for ±ten meters of tolerance in order to be map matched correctly in a road network. This maximum allowance applies mostly in urban areas.

2.5.2 Assisted GPS (AGPS)

This technology is an amalgamation of mobile technologies and GPS. This means that the reading of the location does not only depend on the GPS receiver but is aided using other outside sources, such as assistance servers and reference networks. The assistance server has the main task to compute the exact location whilst collating data from the reference networks. Such technology can be accurate up to ±ten meters (Shukla & Magon, 2006), but this comes at an expense to the end customer. Compared to the above mentioned technology, AGPS need more than three satellites for it to function, which makes it difficult in areas where high tall buildings distort the line of sight. This means that the GPS receiver will then be connected via a wireless connection to the assistance server, and leaves the GPS receiver with the sole job of collecting the measures computed from the assistance server. An example of a AGPS system is Navizon[3], which can either assist your GPS enabled phone for an exact location, or else use cell towers to get a "*rough* fix" (Jeff, 2007). Such data can then be passed to Google Maps to plan routes. Another example is Benefon Esc!. An innovative feature of this phone, allows the user to download topographical maps, find friends using Short Messaging System (SMS) and Mobile Phones Telematic Protocol (MPTP)[4] to exchange location. (LaMance, Jarvinen, & DeSalas, 2002).

2.5.3 Cell of Operation (COO)

This geographic point identification is the easiest to start operating with, since it is already embedded in the mobile technology. This method is used by network base stations to identify from which cell a service is being asked for, be it a telephone call or a message. Mobile cells are dispersed around the network covered areas, thus accuracy can be increased up to 150 meters for an urban area (Shukla & Magon, 2006). Such a method is not suitable for emergency response teams, or for routing, but is an easy option for many entrepreneurs who do not need accurate information about their fleet. The main advantage of this mechanism is that the technology is already deployed and therefore there are little initial costs.

2.5.4 Time of Arrival (TOA)

Making use of measuring distance, like in the case of GPS, this technique relies on measuring the difference in the time taken for a signal transmitting from a mobile device to more than one base station. The position can then be calculated by hyperbolic trilateration. This method has found a lot of opposing ideas since the cost of implementing the technology involved, is very high, as opposed to the performance gain. The reason for such high costs is the need for location measurement units (LMU). As one can easily guess, the accuracy provided by such method is far better than that provided by the Cell Of Operation

[3] Navizon: http://www.navizon.com
[4] More details on MPTP : http://www.benefon.de/pdf/developer_zone/mptp_intro.pdf

one; however it is still not as accurate as desired. Since the signal travels at the speed of light, interference and reflection can create anomalies in the result and often give a shaded area of intersection rather an exact location (Khokhar & Nilsson, 2006).

2.5.5 Angle of Arrival (AOA)

Angle of Arrival is another technique which is useful in scenarios where exact location is not needed. Making use of triangulation, this method needs at least two directional antennas to locate the mobile device. Such method involves making use of very expensive network-side equipment and thus has little to none usage (Dutta, 2006). Another disadvantage of this method is that due to high buildings interrupting the signals, Angle of Arrival works poorly in urban areas (Shukla & Magon, 2006).

2.5.6 Enhanced Observed Time Difference (EOTD)

Combining Time of Arrival with Distance, this method measures the time taken for the signal from at least three known geographical positions. Such measurements are taken at the mobile device, so this means that the device needs to have this technology installed. In some cases this technology requires that the mobile device be assisted by the network, thus passing the time-difference values and triangular computations at the network level. Computations on EOTD can be done using hyperbolic or circular. Most GSM network providers prefer this method to identify the location of the mobile device, since no specialised mobile devices are needed; only investments on the network layer (Dutta, 2006). EOTD is regarded as one of the most accurate non-GPS techniques, however the accuracy limits itself on the visibility of the base stations to the mobile device, and thus one sees a degraded service in urban areas (Shukla & Magon, 2006).

In Hyperbolic method, the Observed Time Difference (ODT) is measured by the mobile device and amounts to the delay of signal between two different known geographic locations. The Real Time Difference (RTD) amounts to the difference between their clock offsets, and the Geometric Time Difference (GTD) amounts to the difference between the time of arrival due to the distance in geometry, and should be equal to the difference between OTD and RTD, thus the mobile device for optimum results should be on a hyperbola. The location of where the two hyperbolas intersect mark the location (Dutta, 2006).

In the Circular method, both arrival times are measured by the mobile device. The distance from the geographic positions is calculated from the result and forming a circle for each geographical position. The estimated position is the intersection of the three circles. To formulate this method, three signals are needed.

2.6 What is a Geographic Information System?

Before discussing how a Geographic Information System can enhance a Location Based Service, it is better to further analyze what constitutes a GIS, how a GIS operates and the advantages a GIS offer.

2.6.1 History of GIS

The first projects that laid the foundation of a GIS date back to the late 50s, early 60s, when Waldo Tobler outlined a model called MIMO (map in-map out) for applying the computer to cartography. The concepts described by Tobler, laid the origins for geocoding, data capture, data analysis and display (Centre for Advanced Spatial Analysis, 2007). During the mid-1960s, the principles outlined by Tobler were enhanced by Howard Fisher at Harvard University. The first lectures were taught during 1966 as a "collaborative regional-scale studio" using SYMAP in a landscape-planning study. Since then GIS started its evolving process, and soon were followed with projects by CALFORM (late 1960s), SYMVU (late 1960s), GRID (late 1960s), Polyvrt (early 1970s) and ODYSSEY (mid 1970s), all projects being developed at the Harvard Lab (GISdevelopment, 2000). Following such lectures, Jack and Laura Dangermond founded Environmental Science Research Institute (ESRI) in 1969 as a private consultant company, later to be one of the main players in the GIS industry.

During the 70s, the first LandSAT satellite was flown, with its main scope being to map the surface of land and sea. Another breakthrough during the 1970s saw the National Informatics Centre (NIC) in India being set up in 1976. Such centre has the focus to create informatics awareness. NIC now operates the largest one of the largest VSAT-based networks of its kind in the world codenamed NICNET.

During the 80s, four students from the Renselaer Polytechnic Institute, USA, embarked on the concept of using GIS to support decision makers, with their software MapInfo (GISdevelopment, 2000). Following this move, ESRI embarked also on their Integrated Cartographic (Arc) / RDBMS (Info) project (Kemp & Goodchild, 1999). In 1984, the First International Spatial Data Handling Symposia was held, outlining the key factors of GeoSpatial data and outlining the future. One year later, the first Open Source GIS Geographic Resources Analysis Support System (GRASS), begins development by the US Army Construction Engineering Research Laboratories (Centre for Advanced Spatial Analysis, 2007). Such project is still under development. The mid-80s saw also another milestone being surpassed, with the Department of Defence (DoD), USA, opening the GPS project for the public (GISdevelopment, 2000). In 1988 the first release to the general public of the Topographically Integrated Geographic Encoding and Referencing (TIGER) digital products was made by the US bureau of Census. Such data is still being updated.

Following the first edition of GIS World, in 1988, in 1992, GIS Europe, a European centred journal was launched, and in 1995, Geo Asia Pacific gets first published (Centre for Advanced Spatial Analysis, 2007).

After a brief look at the history of GIS throughout the last century, it is time to start looking at where this idea is headed to. Geographic Information Systems are growing on a steep scale (Thomas & Taylor, 2006), (Lloyd, Queen, & R., 1993). Many industries are making use of GIS, naming but not limiting, public safety, resource management, governments, road networking amongst others. As the case with other types of Information System the main pulling factor is the increase demand for more information, thus forcing hardware and software advances enabling more functions.

2.7 Defining a GIS

A Geographic Information Systems, can be seen as a stacked set of different map layers, be it raster, vector or otherwise, where each layer is geographically aligned to the other layers (Lloyd, Queen, & R., 1993). One can regard GIS as a digital version of a traditional map, but in reality GIS is far more complex and not as basic. A fore mentioned advantage is the map overlay, which theoretically is infinite, thus one can add more details to the map as he/she likes. Retrieving, storing and analyzing data are the main benefits of a GIS. A GIS does not necessarily mean that data is stored on a RDBMS, but other measures exist, like shape files which lack topological information (Theobald, 2001) just to mention another example. The main ingredient for a GIS to be functional is that the maps (in whatever format) are geographically aligned and referenced accordingly, and then using different other technologies one can change from a format to another without losing the scope of the data. Similar to any other Information Systems, a GIS does not only need the data (the maps), but also need supporting hardware and software. An operating GIS, needs also users since it is up to the users to ask the decision making queries out of the data.

2.8 GIS Jargon

Any technology has its own jargon, so does GIS. In this section, different jargon will be explained and keeping a close reference to GIS will explain what kind of work it is useful for.

Reference to **projection** has already been made in this chapter. By projection it is understood that layers are exactly on top of each other so as to make sure data alignment integrity remains intact. Since data can be acquired from different sources, the needs for standardisation arise. Every data set comes with an extent and a coordinate system, so it is the responsibility of the GIS users to make sure that the projection is kept equal for all layers. Just to mention an example, some maps might use a coordinate system of x, y, z; others might use a longitude and latitude projection, thus for having interoperable data layers, such standardisation is vital.

Since on its simplistic form a GIS can be regarded as a digital map, the need for a scale still remains a vital concern. **Scale** is the mathematical relationship of distance in real life (distance on earth) to the distance projected on the map. Standardisation in such respect is also very important, since all layers of the dataset has to have the same scale.

Following advances in GPS, the need to extend the earth's spheroid became more important. A **datum** can be regarded as a projection of a spheroid which helps in collecting positions on earth. Different datum exist, and like any other geospatial process of converting maps to digital data, it all depends on the map creators. The World Geodetic System of 1984 (WGS84) is the worldwide datum used by all GPS receivers (Price, 2001). Changing from different modules can be quite a tricky task, especially when it comes to combine old and new datum (post 1983 datum can be regarded as new datum). The most important thing is to determine the way that data was collected from the native format, and the process taken to digitalising it. The datum used by the map set for this project is North American Datum of 1983 (NAD83).

North American Datum of 1983 (NAD83) is an earth-cantered datum based on the Geodetic Reference System of 1980. The size of earth was determined using satellites and is said to be as accurate as two meters (Mentor). The World Geodetic Reference System of 1984 (WGS-84) is the mapping datum used by the Department of Defence, the same body who started the GPS project back in the 80s. Little differences exist between these two datum, and data using either of this datum can easily interoperate with just small differences. It is reported that values can differ by more or less a meter (Schwarz, 1989), which is just but a mere difference for most applications. The main difference between both datum is how both are dealing with independent determination of the same thing. Despite having "North American" in its name, NAD 83, it can still be extended worldwide, if accurate data on extending coordinates is available. Point in case, studies have projected areas such as Greenland, Puerto Rico and Hawaii using NAD 83 (Schwarz, 1989). Datum need not be regarded as the solution to everything. discuss that accuracy of Datum points varies depending on the applications' requirements. Table 1 gives an idea on the tolerance expected from Datum used in a certain field. The Future figures presented in Table 1 were further decreased in a study by (McDonald, 2002) to 5 – 10 meters accuracy from civil standalone GPS devices.

Period	Locale	Lateral Tolerance (m)	Longitudinal Tolerance (m)
Present (2000)	Open Highway	25	50-100
	Urban / Interchange	5	50
Future (2015)	Open Highway	1	10-20
	Urban / Interchange	1	5

Table 1 – Error tolerance in ITS applications, present and future (Goodchild & Noronha, 2000)

2.9 GIS and RDBMS

As outlined earlier on in this document, a GIS can provide decision analysis using flat files, shape files and even images. However, many Relational DBMS have embarked on projects to incorporate Geospatial extensions to the core functions. The first discussion of this section will be an overview of how the main RDBMS rolled spatial extensions in their packages.

RDBMS	SPECIFICATIONS
Oracle	• Launched its spatial extension under the name Oracle Spatial, and a lightweight version as *Oracle Locator*; • Oracle Spatial can be used on Oracle 9*i* Enterprise edition; • Provides spatial functions such as area, buffer, centred, and the like; • Supports advanced coordinate systems; • Support aggregate functions and linear referencing; • Topology started to feature since Oracle 10*i* (Sonnen, 2003), (Penninga, 2004); • Included with the package, users can avail themselves of a Map View that can even support Macromedia MX format to render data as a Flash plug-in; • The spatial user base of Oracle is somewhat hard to determine due to the fact that mixed with these functions are others, though it is regarded as the main target are government agencies (Sonnen, 2003).

PostgreSQL	• Widely known open source RDBMS that can be extended with PostGIS, another open source project; • The main core functions of PostGIS were developed by Refractions Research and follow the OpenGIS "Simple Features Specification for SQL"; • PostGIS supports several spatial functions naming just a few such as basic topology support (planned to have full support in the future), coordinate transformation, linear referencing and aggregate functions, besides from spatial functions; • The current state of the topology support is described as on hold (since 2005) and Refractions Research are waiting for an organisation to finance the project and is described as at a pre-alpha stage (Neufeld, 2007); • There is some interest from a third party developer who is interested in supporting water / wastewater networks, but is still waiting for a winning bid to his project (Macke, 2007); • In the pipeline there is planned raster support, networks and routing (though this can be done by a 3rd party plug-in – pgRouting), three dimensional surfaces, curves and splines (Refractions Research, 2007); • PostGIS supports all the standards that are supported by Open GIS Consortium (OGC, OGC SQL functions, OGC Simple Features) (Schutzberg, 2006).
IBM Informix	• Makes use of DataBlade technology, by Illustra, as its spatial data extension; • Thus this allows Informix to offer both 2D and 3D Spatial data handling; • “Spatial” is a free to download module providing location-based data (IBM); • “Geodetic” is another module which costs $50,000 per CPU; • Data is referenced by latitude-longitude coordinates and uses a round earth model; • Supports the Open Geospatial Consortium standards and make use of R-Tree search (Spang, 2007).
IBM DB2	• Also makes use of DataBlade’s “Spatial” and “Geodetic” extenders; • The Spatial Extender extends DB2 by providing a set of spatial data types such as points, lines and polygons as well as a set of functions; • The Geodetic Extender, deals with a rounded shape of earth;

	• Using the Geodetic Extender, text can be converted to a geographical position, such as addresses. This extension is available only to DB2 ESE (IBM, 2007).
MySQL	• Is an Open Source RDBMS which is in its early stages venturing the areas of spatial data; • MyISAM engine already supports simple functions to allow generation, storage and analysis of spatial data; • As from version 5.0.16 such functions are supported on all engine types InnoDB, NDB, BDB, and ARCHIVE; • MySQL, thanks to Alexey "Holyfoot" Botchkov, now is able to support (although at a very beta stage), the Open Geospatial Consortium standards (MySQL Forge, 2007).
Microsoft	• Microsoft have been dealing with spatial objects for some time, with their successful project "Virtual Earth", however only on March 2007 was the first mentioning that the new SQL Server 2008 codenamed "Katmai" was to include spatial data in it; • A function which other DBMS might lack is to support office suites. Katmai is promised to support Microsoft Office 2007 (Katibah, 2007) (Microsoft is said to enjoy a 95 per cent of the market share in office suites (W. P. Carey School of Business, 2007)); • SQL Server 2008 allows for both flat earth (planar) and round earth (spherical) data types; • The third preview of SQL Server 2008, is already compliant to the Open Geospatial Consortium Simple Features for SQL; • SQL Server 2008 should cost as much as its predecessor (2005), $24,999 per CPU (Fontana, 2007).
ESRI	• A pioneer project in the field of GIS was that by ESRI (Refer to Section 2.6); • This closed source project deals differently with data compared to the prior RDBMSs; • Spatial data in ESRI environment is called "geodatabase"; • Using ArcGIS, data can be stored and retrieved from a variety of other industry standards RDBMS, discussed above; • As from ArcGIS 9.3, using ArcSDE data can be stored on any of these:

	IBM DB2, IBM Informix, Microsoft SQL Server, and Oracle. Using third party plug-ins, such as zigGIS[5] and PgARC[6], data can be retrieved from PostGIS too; • ArcSDE is an optimized platform strategy to store and retrieve data using a binary storage format, which according to ESRI's experience is found to "*be the most compact storage of any known technology*" (ArcNews Online, 2005).
SQLite	• SQLite is a server-less Database engine, which is nothing but a software library; • The main advantage of SQLite is that being so compact; it can be used in a numerous devices, such as, cell phones, and PDAs; • Another advantage is that it needs Zero-Configuration and no administrator is needed (SQLite, 2008); • This C library, which is cross platform, has been enhanced also with spatial methods, allowing it to feature perfectly in mobile devices to act as a local geospatial database; • SpatiaLite, is a plug-in library for SQLite, and features most of the OGC Simple features functions; • Allows transformation of Geometries, and also many functions featuring in the OpenGIS Specification; • SpatiaLite allows exporting and importing from shape files, using OGR tools, thus making it quite versatile too, despite its small size; • The coordinate projections allowed in SpatiaLite are Proj.4 and EPSG (Furieri, 2008); • Another project developed by the same contributor, Alessandro Furieri, is VirtualShape, which when used with SpatiaLite library, enables SQLite to access Shape files as virtual tables, thus allowing standard SQL queries.

Table 2 : Comparison of (R)DBMS's

2.10 Storing Data in Shape Files

Data need not be stored only in a DBMS but can also be stored on files, such as shape files. There are cases that data stored on a shape file is more convenient, and can be processed faster if there is change in data. Before one can start comparison, a more detailed description of what is exactly a shape file follows. Shape files store geometric location and associated attribute information without storing topological relationships. Different type of

[5] zigGIS Site : http://code.google.com/p/ziggis/
[6] PgARC Site : http://pgarc.sourceforge.net/

shapes (points/lines/polygons) can be stored together in a Shape file collection. Such a collection will also contain attribute information and also how data should be represented (Theobald, 2001). The most common files featuring in a Shape file collection are (ESRI Support Centre, 2008):

- .shp : Storing the feature geometry
- .shx : Storing an indexed version of the geometries so quick seeking is allowed between geometries
- .dbf : Storing attribute information

2.10.1 Contrasting Data Types

Shape files are composed of both Vector and Raster data which means it is their intention to be built to correlate together, instead of replacing each other. Before starting the process to compare such data types, a brief introduction on each is to follow.

Vector data features are defined by an x, y (z if needed) location. Each vector feature can project location, point, lines, multiline and polygons. Lines or arcs are created by joining nodes each with a beginning and ending point. Vector data storage involves storing location of nodes and topological structure separate.

Raster data features are defined by a grid structure, which is formed by dividing data into rows and columns. Every cell must be rectangular in shape but need not necessarily be of a square shape. Similar to Vector data, each cell is referenced by a location coordinate. Different from vector data, cell's location is implicitly contained within the matrix. This leads to have spatial data being divided into discrete units. Area calculations, overlays and elevation points are typical uses of such data (Ellis, 2001).

Table 3 can outline the advantages, and the disadvantages of both data (Brooks, 2003). As one can see from such table, different usages require different data, thus there is no better data feature.

	Vector Data	Raster Data
Precision in Graphics	Good representation of phenomenological data	Good Representation of Continuous Phenomena (e.g., elevations, reflectance);
Graphical Representation	Retrieval, update & generation of graphics possible	Often result in unpleasant map generation
Data Volume	Data size is compact	Data sets can be quite large
Topology Description	Topology can be completely described	Topology linkages difficult to be described
Computation Expenses	Technology relatively expensive	Technology relatively inexpensive
Update	Dynamic update is harder	Dynamic update is much easier
Complex / Easiness of Structure	Complex structure	Easy Structure
Simulation Difficulty	Due to non-uniform sizes simulation is hard	Data is uniform thus making simulation easy

Table 3 - Comparing Vector and Raster Data (Brooks, 2003)

2.11 DBMS or Shape files

Retrieving data from a Shape file is much faster than querying a DBMS table (Brock Anderson, 2007). In fact, when the GIS project is nothing but a presentation of a huge dataset which is going to be updated very rarely or even never, the best option would be to use Shape files (Tweedie, 2006). Such a belief can also be granted when there is a GiST (Generalized Search Tree Index) acting on the data in the database. Table 4 shows a benchmark performed by Tweedie (2006) indicating the times each query took for MapServer to display the data (Tweedie, 2006). In this scenario PostGIS was used as the sample RDBMS.

Shape file (no index)	Shape file (qix index)	PostGIS 8.1 (no index)	PostGIS 8.1 (GiST index)
2.938s	2.201s	5.297s	4.275s
0.694s	0.294s	2.812s	1.656s
0.601s	0.140s	1.796s	0.987s
0.219s	0.032s	0.914s	0.223s

Table 4 - Shape files and PostGIS Comparison (Tweedie, 2006)

The querying time for Shape files is lower than the same query on the same data stored on PostGIS. However an RDBMS has much more features than a shape file can offer. The most important feature a RDBMS carry with it is the geoprocessing functions. Another feature one finds in a RDBMS which lacks in shape files is the normal features; a DBMS offers over a flat file.

A very common feature would be the availability to define certain record fields, such as keys and indexes which help in search queries. Integrity constraints can be created also on a RDBMS in order to provide validity checks over the data. Also an RDBMS can handle one-to-many relationships, something which lacks in flat files (Databasedev, 2007). Thus features like table locking, data validation with triggers, multiple user handling, and rollback functions are all features that can be performed only using a DBMS. Another very important feature is that using a RDBMS one can have more control over the data since all data will be centralised, even though this can result in a higher centralized risk. Another advantage over flat files enjoyed by RDBMS is the availability to generate reports which can filter data and display chosen fields (Databasedev, 2007).

2.12 Summary

The main areas of a location based service and how a GIS can enhance the service offered were discussed in this chapter. In the Literature Review, found on Chapter 3, a review of companies who ventured in Location Based Services will be discussed and such scenarios discussed from the point of view of profits, technologies used and success factor.

Chapter 3 : Literature Review

3 Literature Review

3.1 Introduction

In any environment where a GIS is set up, changes are reflected on how spatial data is managed and analyzed. Failure to do so will result in a little to no advantage over using non-digital data. As for any other information systems, the amount of attainable goals set for a GIS are the basis to verify if the project succeeded or failed. Thus setting up goals during the planning phase of the system life cycle is vital for the project. Close attention has to be given to the users in the system, the variables, and the aims that the improved system is to acquire. In order to be able to achieve such goals, relevant experience has to be gathered. Such expertise can be gained by reviewing other studies and projects in order to ripe the benefits attained so as to apply them to the proposed system.

This chapter will start off by reviewing research which tackled the areas of study, being, location based services, but also Geographic Information Systems. A closer look will be given to a niche section of GISs, GIS-T (Geographic Information Systems in Transportation). By reviewing other studies, experience can be collected so as to propose a better system. In the following review, one can find also a review of how systems were studied in terms of planning for data transfer, preserving data constraints and integrity. In this chapter, one can also find a review of a number of GIS projects, their development and the problems encountered. It will also review the advantages obtained both technologically and from a business perspective.

The project is proposing a study and plans a new Geographic Information System, which can be accessed by mobile users, hence the idea of a Location Based Service. The service should include options to offer different routes to the user depending on the choice or the type of user. These main objectives need to be kept in mind whilst reading this chapter.

3.2 The Network Model

According to Goodchild (1992), three classes are applied to transport GIS (GIS-T). These classes laid basis to many GIS-T projects. Later on, in another study, Thill (2000) summarised these classes as:

- Field Models: This class deals with representing the continuous variation of an object over space. A scenario of this model is Terrain elevation.
- Discrete Models: This class depends on the geographical entities present in the geographic model. Examples of such a class can be bus stops, train stations, fuel stations.
- Network Models to represent topologically connected linear entities: This class refers to entities which are connected together to form the data network. Such entities need to be referenced and fixed on the surface.

Thill (2000) continues by stating that a network model built around the concepts of arc and node are the most appropriate models used in applications related to GIS-T. This is because single and multi-model infrastructures are: *"vital in enabling and supporting passenger and freight movement".* Some examples of applications which are useful to this study using such a network model are:

- Real-time and off-line routing procedures;
- Emergency vehicle dispatching;
- Web-based traffic information;
- Real-time congestion management and re-routing accordingly.

The above mentioned examples of applications where a GIS-T can be used can easily identify with the main objectives proposed in this project. Thus one can deduce that the objectives of this project reflect many of the characteristics present in a network model. Such a model brings more challenges than a conventional GIS would, mainly since it combines also the area of Transportation Information Systems (TIS), as Vonderohe et al. (1993) suggested. According to Gupta et al. (2003) TIS has the advantage of data integration. Thus data collected prior to the project, both internally and externally can be collated together. This union between a GIS and TIS, commonly referred as GIS-T, contains also challenges. These challenges are related to Data Management, Data Manipulation and Data Analysis. These areas of interests are further investigated in Thill (2000) and are summarised below.

Data can be collected from various parties over various time periods. Such data can be stored for future use. Once the data is used later on, the first set(s) can be referred as Legacy Data. The first challenge faced by a GIS-T would be the need to incorporate fragmented Information Systems owned by various parties. The developers of GIS-T are required to combine all these legacy systems. Legacy databases/Systems can hold a hard to notice amount of wealth which should not be overlooked when a centralised system is being planned (Goodchild & Noronha, 2000).

Similar to the first challenge, interoperability deals with having these disperse data systems interoperating together. This technique needs a heterogeneous approach on naming, classification and topology, present in the GIS-T storage. Thill (2000) outlines a three-step approach for achieving data interoperability successfully:

- Algorithms for map matching;
- Models of error;
- Error propagation in transportation data.

Dueker & Butler (2000), argue that sharing GIS-T data is "*both an important issue and a difficult one".* Difficulty can be attributed to the different ways in which geographic elements are represented. An example is provided in Section 2.8. Also, Dueker & Butler (2000) in their study they mention that there is a lack of agreement between transportation organisations in identifying common objectives for the data. Sperling & Sharp (1999) describe this lack of agreement in different scenarios such as the amalgamation of TIGER data with local street map data. This can create difficulties since the scope of the exercise for which the data is being collected differs, thus, the result and the level of details differs as well.

The different approaches also affect how streets are broken into logical statements which can be prepared for information extracts. For example routing software would accept a straight highway as three line segments, ensuring that the driver starts the highway and does not exit before the end of the line (broken down in a few segments). In accident recognition software, the same highway has to be segmented in smaller and more manageable slices for the accident to be located more effectively. Dueker & Butler (2000) prefers to use the terminology entitled "transport features" since these can be used and include railways, shipping lanes, air routes and any other linear features that function in the Network Model. These issues have generated various academic works some of which are mentioned in this chapter to propose methods by which data interoperability issues are minimised. One technique to do so is the method described by Sester, Anders, & Walker (1998) where a top-down approach is proposed. This approach allows spatial objects to be aggregated into larger transport features. This will enable any application-specific GIS to choose which layers it needs, and just derive the data from one source. For example, a GIS tackling railways, need not use the streets layer, but just the railway one, thus query only the railway provider. Devogele, Parent, & Spaccapietra (1998) provided another technique on how to facilitate this process. The solution presented in their research, accepts a wide range of formats, out of which different scaled databases can be aggregated.

This issue of interoperability is not a onetime problem, since as the need for more data grows; the need to manage data becomes equally important (Ramakrishnan & Gehrke, 2003). Dueker & Butler (2000), present a looser model which allows for each piece of data to be made available for use, right away. This model claims two fundamental principles for sharing data;

- Transport features must be uniquely identified, thus any user / supplier of data need to identify common features;
- Data providers need to include a standardized unique identifier with each feature.

In preparing a GIS for vehicle tracking, planning is necessary to identify the transport feature (street) and also to position the street itself on the network. Despite the agreement within the Intelligent Transportation System (ITS) community to have a "cross-streets" profile, which provides a message with the name of the street, and the exact location of the segment currently available maps are far from desirable when it comes to standardising names. This affects negatively the performance of routing (Noronha, 1999).

3.2.1 Online GIS-T

Online routing, traffic management and congestion detection are all types of applications where online data is necessary. GIS-T is required to pursue the issue of data quality and timeliness. Data retrieval has to be made as simple and lag-less as possible, whilst processing and storage techniques have to be improved. In contrast with what Thill (2000) explained how data manipulation still lacked what is required by the society with regard to spatial data, Downey (2006) claims that these techniques did improve. More recently, Wootton & Spainhour (2007), discuss the fast improvements which took place during these years with regard to data manipulation tools in GIS. Routing algorithms, for example, need to be enhanced so as to attain the required advantage of real-time data.

3.2.2 Large Data Sets

Geographic Information Systems tend to hold large quantities of geo referenced data. Visualisation techniques have come a long way, now able to handle much larger data sets than before. Eight years ago, Thill (2000) mentioned that visualisation techniques did not adapt correctly to the change in data, and thus the society was relying on software which was designed to handle much smaller data sets. In contrast to this picture, Kardos et al. (2007) discuss that visualization techniques have come a long way, and large data sets can be easily handled. Richard et al. (2006) proposed Grid-computing to help achieve better support for large datasets, especially when it comes to visualise data.

3.3 The need for Location Based Services

One of the most important needs for an LBS is to provide an added service to the client. The client, being the end user would need to be the main focus of the service being planned. The activities performed by the user need to be studied in order to make sure that with the least effort from the user, knowledge is gained. "*An activity is a sequence of actions conducted by a human being aimed at achieving a certain objective*" (Nardi, 1996). In mobile situations, such objectives can vary from finding the way to an item in the super market, finding the way to the nearest shop or even more complicated like finding a person amidst a crowd.

The activities mentioned earlier can be classified into groups of activities which any LBS user needs to perform. Meng, Alexander, & Reichenbacher (2004), identified five main concerns for a user within a geospatial context. These points are:-

- "Where am I?" – This is a basic question can which supports all the queries in a spatial context. Even if the user is not particularly interested in the answer to such question, the answer is still important as it provides the system with the **location** of the user. "Where is?" – This question which can be simplified to **searching** in space for an object or a person.
- "How to go?" – This leads the user to **navigate**. Once the starting point of the user is established a route is developed in order for the user to reach his destination. "What is?" – At this point the user **investigates** the identity of the location, object or person in space.
- "What is happening?" – This question can be asked in environments where the need to **check** what is happening arises. Steiniger, Neun, & Edwardes (2006), explain that this question can also take the shape of temporal information, since it takes in account the state of entities within a time frame.

These points can be regarded as the basic spatial functions of GIS projects. These same questions have to be answered by any organisation embarking on a GIS and a LBS system. System designers have to make sure that at any point in the system these spatial queries/ tools are supporting the IS.

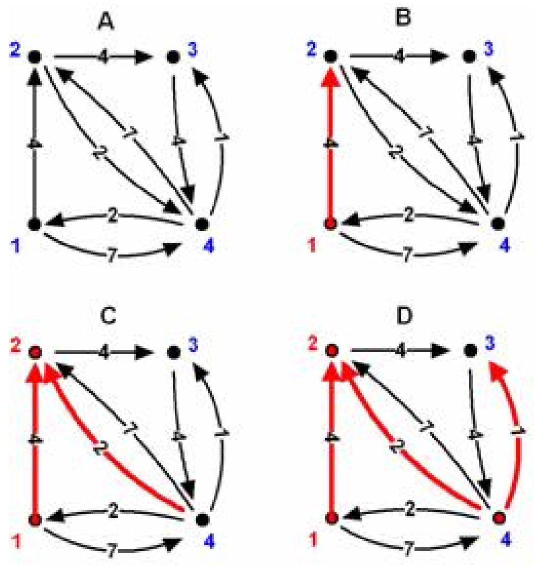
3.4 Finding the best route

Different studies were undertaken to generate routing queries, some of which are going to be reviewed in this section. Mainguenaud (2000) discuss route finding techniques which differ in size and methodology. Whilst certain routing does not need a 2D representation of the network, such as when preparing routes to plan bus stops along a public transport route. By 2D representation it is understood that the plane on which spatial data is represented takes a flat plane, thus no consideration of altitude is given. There is a difference between pre-determined route findings to dynamic routing where the route depends on the changing data on the road network. Bierlaire & Frejinger (2008) claim that a network which lacks integrity (an incomplete road network), can result in wrong routing results. Whilst discussing the resulting variables in an incomplete network, Bierlaire & Frejinger (2008) refer to three pieces of study where the route completeness after following the computed routes was successful. The first study shows that the best results attained by Wolf et al. (1999) only matched 63 per cent of the exact map data. A more recent study performed by Nielsen (2004) showed that 90 per cent of the trip readings (performed by a GPS) collected in Copenhagen region missed some data. Thirdly, Bierlaire & Frejinger (2008) reports that Du & Aultman-Hall (2007) claims that the best algorithm available in the market at the time reached only 94 per cent of positively identified trip ends.

In another research tackling routing practice Smith, Goodchild & Longley (2008) various optimal routing techniques are analysed and discussed. For the scope of this study, only Shortest Path techniques will be discussed.

Some of the first notions of a vehicle routing problem and the use of mathematical algorithm featured in Ramser & Dantzig (1959), where the problem looked at how a delivery truck could reach the service stations supplying them with goods loaded at a terminal in the shortest route possible. Since then, routing has continued to develop, and many studies have been published in this area, some of which are discussed further on in this section. A recent study which based its methods on Ramser & Dantzig (1959)'s research is Mitrović-Minić (1998) and later by Mitrović-Minić & Laporte (2004). These studies conclude that optimization techniques in solving pickup and delivery problems are not promising. The main application of these techniques is when a route is needed between a source vertex and a target one. Thus routing algorithms can be deemed more successful when the starting and ending vertices are known prior to the exercise, rather than trying to minimise paths when visiting vertices. In order to obtain a result, a mechanism to capture the cost of traversing the route is needed. Any routing technique needs to be computed upon a pre-defined network, as established by the Network Model discussed in Section 3.2. Having the network ready, different researches proposed different algorithms, in which several of the outcomes provide a method by which a route starting from a single vertex can reach all the rest of the vertices in the network. These are called single-source SPAs or single-source queries (Smith, Goodchild, & Longley, 2008). A benefit of these queries is that they can be enhanced to allow for secondary constraints, such as Boolean weights or differential costs. Below a discussion of these queries will follow:-

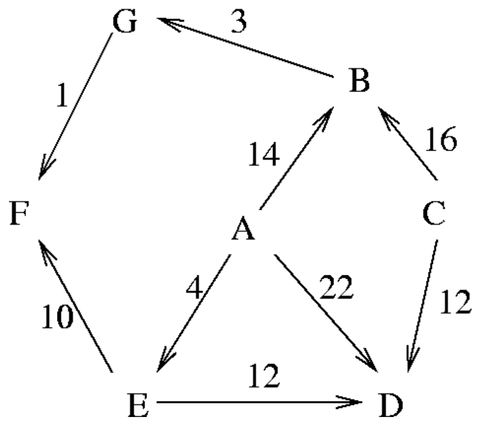
The **Dantzig algorithm** relies on searching for the shortest route using a positive edge weight. Thus route restrictions, as described earlier in the previous section, for example Boolean weights, are possible. This can be seen as a form of discrete dynamic programming (Smith, Goodchild, & Longley, 2008). Figure 3 outlines the algorithm.



1. Get the starting vertex. *Vertex 1*
2. Find the shortest edge (by shortest edge it is understood the determining factor deciding the algorithm. If two edges have the same cost, randomly choose one. *Edge 2 / Cost 4*
3. Add vertex 2 to the result set.
4. Find the shortest edge from either of the vertices (1 or 2 in this example), thus from the result set and from the found edge in step 2. *Vertex 4 / Edge 2*
5. Add the result to the result set
6. Find the shortest edge from the result set to vertex 4, thus following the path 1 – 2 – 4 -3 as per diagram D. The cost is *7*
7. Add the new vertex, (V*ertex 3*) to the result set
8. Stop the process since all vertices reached (*1 – 2 – 4 – 3).*
9. Repeat the process starting from vertex 2, until all four are covered again.
10. Repeat step 9 starting with the next vertex
11. Stop when all vertices have been parsed and the route computed starting from each.
12. Return the shortest path traversing all the vertices.

Figure 3 : Dantzig Algorithm Explained (Smith, Goodchild, & Longley, 2008)

Similar to the previous algorithm, the **Dijkstra algorithm** works by storing the current cost of the shortest path in each parse. Figure 4 explains this algorithm in detail.



1. Find the starting vertex. *Vertex A*
2. Start all vertices so distance from source to target (dist(t)) is infinity
3. Initialise distance from source (dist(s)) to 0
4. Scan all the edges coming out from the starting vertex
    a. For each edge found. *Vertex B = 14, D = 22, E = 4*:
        i. Add the edge's distance to dist(s)
        ii. If dist(s) is less than dist(t)
            - dist(t) = dist(s)
5. Move on to the next vertex
6. Repeat steps 2 to 4 until all the vertices are parsed
7. Output the *length* covering all vertices. *Resulting Output : A,E,F,B,D,G*

Figure 4 : Dijkstra Algorithm (Ryan, 2004)

In this line algorithm, the output of the process is the length. Vertex C was not included since it cannot be reached from A, thus will not be visited by the algorithm. Unlike Dantzig, Dijkstra provides only for the length of the shortest path found. Application enhancements can allow this algorithm to store also the vertices order during the parsing and can output the path in step 6. However this depends on the application of the algorithm (Smith, Goodchild, & Longley, 2008).

Smith, Goodchild, & Longley (2008) define this algorithm as a "*greedy*" one. This is also reflected in the fact that it can be computed in incoherent times. In real world scenarios, this can be attributed to the fact that transport feature segments have very limited vertex connectivity. To counter balance this runtime, one could use the A* algorithm.

The **A* algorithm** can be viewed as the simplest of the three. This algorithm is goal oriented, thus the first route that is found going through all vertices is handed out to the user. Since A* uses a heuristic technique, finds a good solution in the shortest time, sometimes by sacrificing the quality of the output. The first route must still follow some procedures to be found which can be summarised below:-

1. Find the distance of each vertex from the target
2. Add this distance to the distance found by the network to this vertex if it is an open path. Ignore closed paths.
3. Add the open path to the Open List (commonly denoted as G – the cost of the path from source to end)
4. Repeat Steps 2 and 3 until the path is populated and you get the final path (commonly denoted as H – the cost from the starting point to ending point)
5. Using a pre-defined priority method, visit all vertices starting from the target, recording the

Figure 5 : A* Algorithm (Dechter & Pearl, 1985)

In step 5, the "pre-defined priority method" is mentioned which can be summarised to the three mostly used heuristic approaches to identify distance (Dechter & Pearl, 1985):

- Euclidean distance > $Sqrt(Dx^2+Dy^2+Dz^2)$;
- Manhattan distance > $|Dx|+|Dy|+|Dz|$;
- Maximum distance > $Max(|Dx|, |Dy|, |Dz|)$.

Heuristics results and the computational power is required for each to run are difficult to compare (Marchesin, 2003). According to Mitrović-Minić (1998) the most suitable way to test for a new heuristic approach is to compare its results to previously known ones.

Since this algorithm typically visits fewer nodes than the rest of the mentioned algorithms, the resulting computing time should be "*faster – often a lot faster*" (Smith, Goodchild, &

Longley, 2008). Thus this algorithm is more useful in scenarios where an optimistic estimate of the cost is enough. By optimistic it is understood that the true cost of the path, will be at minimum as big as the estimate. Another attribute of A* is that once it finds a path, it does not try to find a shorter one (lower cost), and can be described as admissible (Detcher & Perl, 1985).

3.5 Drawbacks of Automatic Routing

In the previous section, routing techniques have been discussed. The advantages of finding the shortest path between two points and also different ways to compute such algorithms were reviewed. In this section, an analysis of the drawbacks of automatic routing will be discussed. Any project will always have some drawbacks, and a routing project is no less. The main drawback discussed further on is the alienation sensation felt by the driver.

In a study done by Leshed et al. (2008) drivers become more relying on technology and are disorienting themselves whilst driving. As one of the respondents to their survey states, as long as he knows what a quarter mile is, he feels ready to embark on a journey. This results in a lot of drivers not paying attention to the road itself and just relying on the instructions given by their navigation system. Burnett & Lee (2005) go a step further and claim that this technology is resulting in a poor reconstruction of the environment one is driving through. Leshed et al. (2008) mention also that GPS routing, is an example of Borgmann (1984) fundamental criticism. He blames technology for demanding less skill and knowledge from the user, thus resulting in a more laid-back user. However, Leshed et al. (2008), continue by mentioning that some degree of engagement with the environment is still necessary, since only a driver would be able to deal with unexpected happenings on the road. To counter these two claims, Leshed et al. (2008) propose that more integration is built in between GPS devices and the users. In concluding their study they do not investigate a redesign of GPS technology, but that more care is given to the user's experience of driving and his environment.

Besides from disorienting the user from the environment, automatic routing can also result in travel time expectations. If the user relies blindly on his routing device, traffic conditions amongst other factors can change and therefore altering the outcome (Karl & Béchervaise, 2003).

These negative effects on the user can be attributed to automatic routing. Since route finding is becoming more and more easy to use, and user intervention is minimised, the user is becoming more alienated from the environment. Such effects can be to the detriment of the well-being of the drivers.

3.5.1 Dealing with arriving late

Factors influencing travel time include driving habits on the roads (Li & McDonald, 2002), special events (Karl, Charles, & Trayford, 1999) and weather conditions (Chien & Kuchipudi, 2003). Another example is a road block or accident along the route (Hawkins & Stopher, 2004; Lin, Zito, & Taylor, 2005).

Another problematic scenario has been discussed by Ramakrishna et al. (2006) in an attempt to solve the bus approaching query. This query tries to provide the user with the expected time of arrival of a bus, by applying the current state of the road with the travel speeds to destination. Lin, Zito & Taylor (2005) discuss two approaches: regression methods and time series estimations, thus estimating the arrival times of a bus. Anderson & Bell (1997) proposed also what Lin, Zito & Taylor (2005) describe as an amalgamation of two methods; "data fusion". Ishak & Al-Deek (2002)'s research however does not support these finding and considers data fusion as inaccurate.

Despite the results mentioned here, in a study performed by Haynes et al (2006) shows that a moderately strong association between actual driving times and estimated times by a GIS exists. The ratio was of 0.856. As one can see in Figure 6, more than 50 per cent of the estimated journey times were erroneous by just five minutes, 77 per cent by ten minutes, and 90 per cent were within fifteen minutes. Many of the respondents answered that this change was due to traffic congestion and also to the fact that they rounded the driving times when answering the survey.

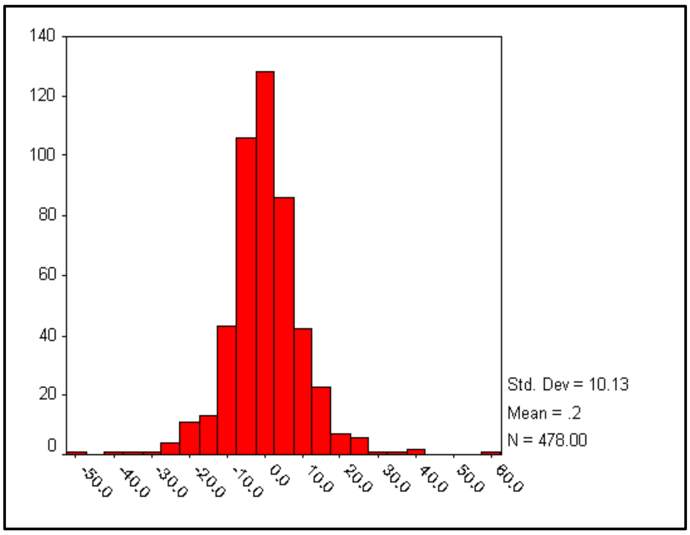


Figure 6 : Difference between reported and estimated car journey time (mins) (Haynes, Jones, Sauerzapf, & Zhao, 2006)

The above studies tackled how one can incorporate temporal queries with spatial data so as to combine routing within a time frame. This section has reviewed studies which tackle time series estimations, thus trying to eliminate the possibility of a driver to arrive late at the destination. In the following section, studies concerning proposal plans for new GIS-T systems follow.

3.6 The Choice of Variables

In this section, the variables affecting the routing are going to be discussed. Different studies have dealt with separate (sometimes including other) variables. Each of these studies will be reviewed.

In studies performed by Cova (1999) , Berry et al. (2005) and Kovel (2000), sensors are used to alert the system that an emergency situation has rose. The knowledge-based support system described in these studies, made use of meta data so as to be able to interpret results coming from the sensors to meaningful concepts. Thus they had to make use of data collected only from description and not from source. Whilst Berry et al. (2005) and Kovel (2000)'s papers detail how a GIS can prevent incidents, Cova (1999) focused more on short term based emergency response, such as flooding or earthquake occurrences, but not so much on how to route response teams.

In another research, Parentela & Sathisan (1995), start their research by quantifying Preparedness which is the amount of time a state is able to deal with an emergency response. Time is well known to be critical for these applications, thus most of their research was outlined using this determining variable. According to Parentela & Sathisan (1995), two ways to deal with this problem are: knowing exactly the exact location of the accident and prepare a routing itinerary from the nearest response point. Response time as recommended by the Federal Highway Administration (FHA) should be 10 minutes (Parentela & Sathisan, 1995). This is the time an emergency station is allowed to response, and then another 45 minutes are allowed to arrive on scene. Thus, the network under study gives a travel time of 35 minutes (45 minutes in transit *minus* 10 minutes maximum delay to receive the call). In every area of study there will be areas where this transit time will take longer, due to the geographical distance from the aid station to the accident.

Despite that 72 per cent of people who work in some type of Emergency Response only 28 per cent have ever used GIS (Parentela & Sathisan, 1995). This study reported that only 33 per cent of the group, showed a desire to learn GIS in order to provide a better service (Spearin, 2002).

Once a review on how surveyed Emergency Response personnel view GIS, and their reluctance to learn how to use it, the view now is shifting on how data can be managed in order to provide a service within the time factor as stipulated earlier on in the document. Response time is a critical factor whilst planning and setting up of an Emergency Response team, thus a review on how this factor was studied is felt needed.

Another study was carried out by Derekenaris et al. (2001). This study tackled the Emergency Routing management problem by looking at time as a critical factor. Response time is improved in emergency situations. However they recall that finding the best route in a dense road network is not an easy task, and can result in time-consuming algorithms. Route planning performed by a GIS and its capabilities to handle spatial features, can be used as a Decision Support System (Crossland, Wyne, & Perkins, 1995). The same spatial features discussed in Crossland et al. (1995) can be used also in other GISs to allow the management of other vehicles (Hafberg, 1995).

The system proposed by Derekenaris et al. (2001), as proposed in Figure 7, makes use of the centralised database, as suggested by Laurini (2000), Cheng et al. (2004) and Chang et al. (2000). Data collected will be passed to the Operations Centre which is where the actual routing takes place. Laurini (2000) discusses how this model is easy to build however can be quite bothersome if there is a downtime in their central system. Another method used in this study is that of geocoding all telephone numbers (fixed lines) in order to be able to route an ambulance to a house call if the request for aid cannot be heard properly. This is similar to what the Federal Communications Act (FCA) in USA and the European Commission are proposing with the likes of e911 and e112 systems. Derekenaris et al. (2001)'s system was not set up to deal with mobile phones.

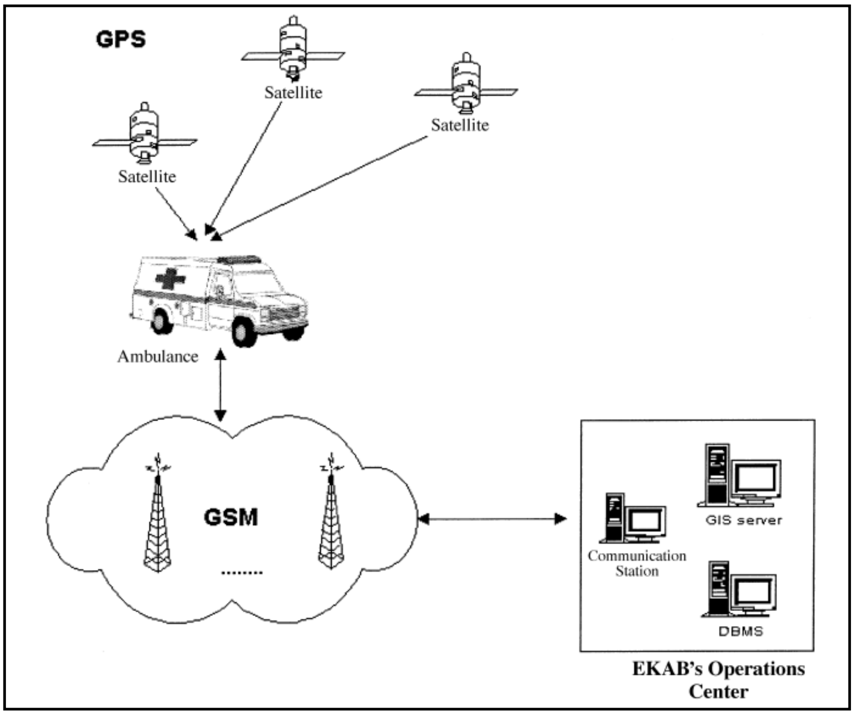


Figure 7 : EKABs System as proposed by Derekenaris et al. (2001)

Derekenaris et al. (2001) are proposing to deal with routing by taking the least minimum time to arrive on scene whilst integrating the actual data on the network. Data is collected from road and other sensors loaded on the ambulances thus making use of the Floating Vehicle approach suggested by Gates, Burr & Simmons (2002). The algorithm chosen for their dynamic routing was Dijkstra, due to its ability to propose the shortest path.

The above review concluded how previous studies tackled the management and routing of ambulances within timeframes (remembering the limits as proposed by the Federal Highway Agency), which concept can easily be used in all emergency response teams. Now that routing algorithms were reviewed, and also their usability in the emergency response area, the view will turn on how to compute a route plan.

What needs to be mentioned here is that only few studies were conducted tackling how the user can decide which decision to take. Also, fuel consumption features very little as a factor in these studies. Lin (2002) conducted a research using fuel use by applying the expertise of drivers, mainly taxi drivers to check how these prepare their routes within the city. The results proposed by Lin (2002) are quite opposite to what Leshed et al. (2008) proposed. In many of the responses, it was evident that drivers took into consideration time of day, speed, fuel consumption and congestion. Many of the respondents to Lin's survey resulted that the routes chosen were very similar, when given the same time of the day. These results have led Chang, Wu & Ho (2000) to develop a network system where routes were planned and fuel consumption, speed, and real-time traffic information featured. The system proposed by Chang, Wu & Ho (2000) is presented in Figure 8. In this system, according to Lin (2002), data passed on from the Location Based Service, will feature existing data from various sources, and combine it with pre-gathered information from previous knowledge developed in the system. Chang, Wu & Ho (2000) conclude their research by proposing a new technology which integrates road experience from the driver and shortest-path techniques in improving routing algorithms.

Emissions are discussed by Ochieng et al. (2002). The very high level report deals with enabling spatio-temporally data being collected from vehicles and transmitted to a centralised database. Software processes the information on traffic and attempts to disperse high concentrations of traffic to reduce emissions.

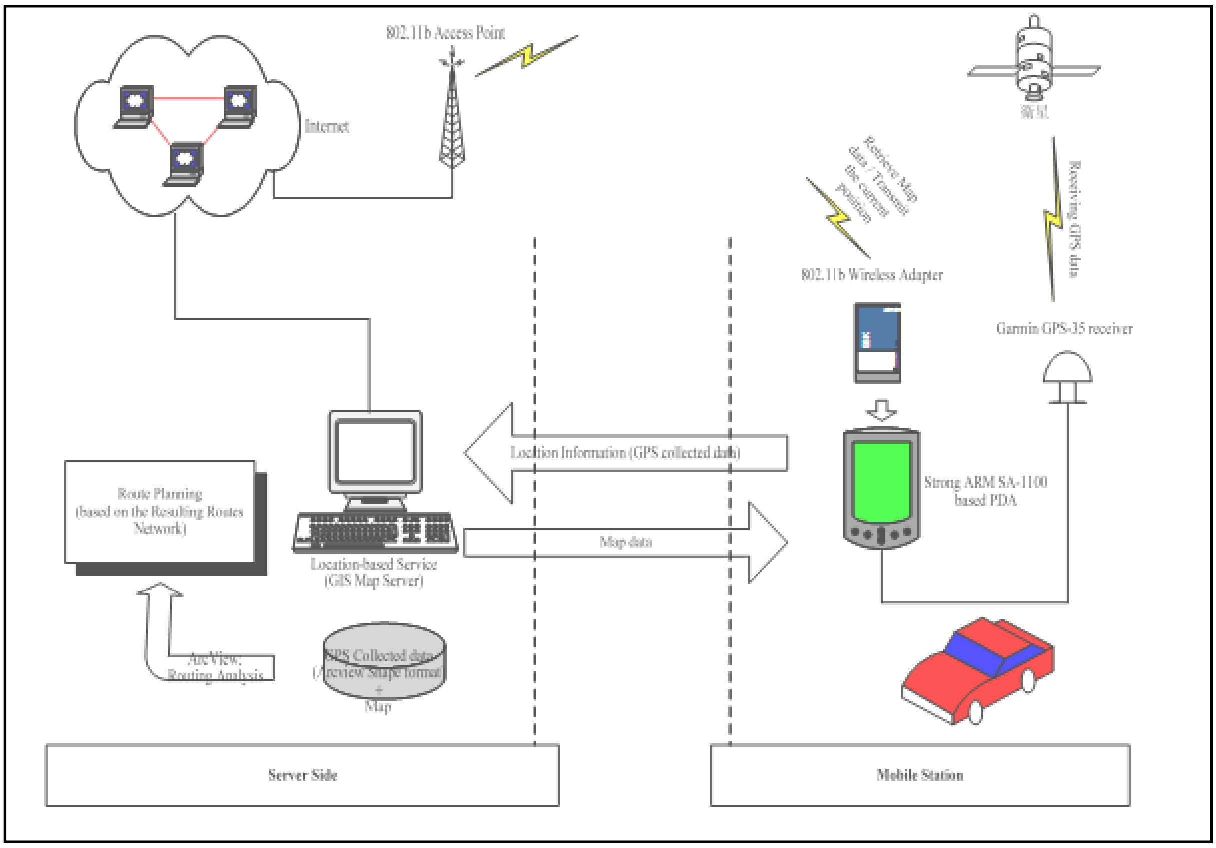


Figure 8 : System Proposed by Chang, Wu & Ho (2000)

It is believed that drivers do take into consideration the time of day (Leshed, Velden, Rieger, Kot, & Sengers, 2008) when planning a route, but this is not always the case with software aided routes. Studies tackling dynamic routing will be reviewed in order to obtain the necessary knowledge for the planning of this project's system.

Conceição et al. (2007) propose a system that dynamically routes the vehicles according to live data from the network. This system, similar to the one used by Chang et al. 2000 makes use of a mash up created by Car-to-Car networks[7]. This Car to Car approach allows vehicles to inform the back end system with any data found whilst traversing the network, either automatic or by input from the user. This geospatial information allows better information share between all participants. Traffic conditions can be injected into the system, thus enhancing the system with more knowledge and therefore provide better decision support. In this model proposed by Conceição et al. (2007), there is no need of a back-end, since the

[7] Car-To-Car Communication Consortium: http://www.car-to-car.org. This non-profit Consortium was initiated by European Vehicle manufacturers and tries to improve road traffic safety and efficiency by means of inter-vehicular communication.

data is being collected and reproduced to the networked cars straight away. Thus, routing will be provided using the In-Vehicle devices. The project proposed in this paper, named DIVERT makes use of Open Source licensing allowing contributors to help in the project. Their simulator takes into account high visualisation and inter-vehicle communication as can be seen in Figure 9.

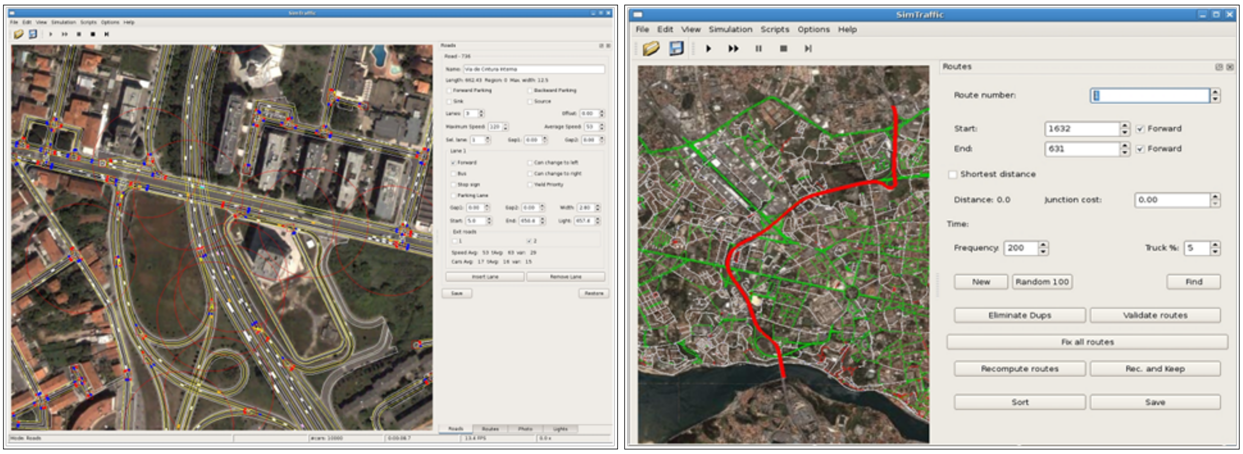


Figure 9 : Screenshots from DIVERT. (Left) Road Editing Interface. (Right) Setting up the Pre-defined route Interface. (Conceição, Damas, Ferreira, & Barros, 2007)

According to the case study presented in the FOSS4G 2007, Conceição et al. (2007) describe their next step in their project thus to take into consideration 3D modelling. This will allow the user to adjust acceleration due to steepness, whilst also, improving wireless communication through ensuring line of sight. In concluding their study, they propose that their next research would try to consider colliding vehicles on the simulator. The vehicles in the simulator, have their path chosen randomly from a set of pre-defined routes as can be seen Figure 9 (b). The pre-defined paths are set up using the Shortest Path algorithm, based on time; taking into consideration also previously collected average segment speeds.

Donlon & Forbus (1999) also conducted a study dealing with Dynamic Routing. This research differs from the rest of the studies discussed till now since it tackles routing in areas where there are no pre-defined roads. This report discusses how military vehicles can be routed in hostile environments. According to the study, this project is a novel use for GIS, in the sense that queries before were of a qualitative nature, whilst this study looks at proposing quantitative approaches. In essence, the project is not that different, since trafficability is still a dominant factor: "Trafficability is a measure of the capability for vehicular movement through some region" (Donlon & Forbus, 1999). Thus what the authors want to conclude is that the relationship between a moving entity and the area where it is allowed to move is still existent. The only differing option is that the area per se is not pre-defined. From this research one can deduct that routing can be dealt in similar ways, no matter the area and the traffic. Other studies however still look at the provision of access.

The inclusion of fuel consumption in this type of routing heuristic is less explored. This can also be seen from the limited amount of literature one finds on this topic. A study carried out on this area was published by Ericsson et al (2006). They start their research with a quick overview, explaining why it is important to start taking into consideration fuel consumption. Besides the increasing fuel prices, one should keep in mind emissions as well. According to figures published by the European Commission (2006), transport accounts for more than 30 per cent of total CO_2 emissions in the EU, a 9 per cent increase over three years (EEA, 2003). Such an increase in emissions due to transport is now reaching 20 per cent of total gas emissions (Department of Transport, UK, 2007), pushing the figure to 26 per cent higher than those reached in the 90's (EU-15, 2006), despite their constant decrease of 12.4 per cent per annum in the late 90's (Jones, 2006). Such figures clearly shows that to avoid further damage to the environment, better fuel management is needed.

In the system proposed by Ericsson et al. (2006) the technique of a floating vehicle is used again, in order to collate data and use it for dynamic routing. In the equation, care was taken to avoid junctions which are controlled by traffic lights, which are hot spots for fuel consumption (Brundell-Freij & Ericsson, 2005). Also in an attempt to make sure that the accelerator speed is not pressed uselessly calming music was provided to the driver (Rosqvist, 2003). From the sample, 109 trips were completed, out of which 50 journeys showed a decrease in the fuel consumption. The 50 journeys recorded were made of 35 off-peak and 15 peak trips. Despite Li & McDonald (2002) claim that in some years every vehicle will have an IT platform to act as probe vehicle, the survey proposed by Ericsson et al. (2006) lacks such probes. This is confirmed also by Davidsson et al. (2002) where it is proposed that at least 1 vehicle per kilometre, is covered. Ericsson et al. (2006) had only 0.4 vehicles per kilometre. The results, according to Ericsson et al. (2006) can be further seen on a larger data network, where more disturbances are caused in the network. Here a disturbance is reflected in a car stopping for more than 80 seconds, and therefore slowing down the whole network capacity. This disturbance has a 26 per cent probability to be captured by a probe vehicle under study. The conclusion from this study, is that similar to what Johansson (1999) found out; fuel consumption can be reduced by 11 per cent. This however is a floating value, since it was observed that over a long time, drivers tend to get used to the road, and do not respect precisely the pre-defined route. Another important finding from this study is that no significant fuel reduction was found, when the proposed route was based on the fastest route, as opposed to the choice over shortest route. Despite these findings, Smith (2006), shows that prediction techniques, in both traffic models and also fuel emission depends on the modelling environment. This means that the effects may vary in the environment where the modelling is being practiced.

Ericsson et al. (2005) concluded that fuel consumption depended also on the driver. Changing of gears, breaking power, acceleration and handling of car all played their role in fuel consumption. A similar research proposed by van der Voort et al. (2001) proposed a system where the driver was advised when to change gear and do certain manoeuvres so as to minimise the consumption of fuel. Greenroad (2007) found a correlation between aggressive driving and peaceful driving. Their research has also found that aggressive drivers need to visit a fuel station every 3 days on average, whilst peaceful drivers every 5.1 days. Another interesting finding is that aggressive driving results in an increase of about 10 per cent of fuel consumption when compared to driving safely.

Variables	Aggressive Drivers	Peaceful Drivers
Sample Number	663	531
Driving Hours between fuel station visits	7.4	8.8
Risk Index	56.3	14.1
Miles per gallon	29.1	31.3
Days between fuel station visits	3	5.1

Table 5 - Risky driving vs. Safe driving (Greenroad, 2007)

Thirty four per cent of the total road transport emissions in the EU are due to road freight countries (EAA, 2006). Though the rates were not as high as in the end of last century, the figures were already increasing. Ma et al. (1999) have researched this area and proposed theoretical ideas to help the environment. What they propose is that it is a rational alternative to start developing "environmentally-friendly" transport modes, which emit low rates of fumes. In their study they proposed to route transport vehicles as far from urban areas as possible to help the level of emissions inside a city remains to a minimum.

3.7 Case Studies of organisations using Open Source

Since this study uses Open Source Software it was felt appropriate to review case studies of organisations involved in the use of Open Source.

Infoterra

Back in the 1980s, the UK national Remote Sensing Centre started seeking partnerships with space agencies and expanded aerial data collection. Late in the 90s, this Centre was privatized and renamed "Infoterra"[8]. The main task of this organisation is to provide remote sensed images and aerial photography.

The main concern for Infoterra was always how to manage the huge collection of data for its customers, especially when the Internet was introduced. This step required Infoterra to provide more access to its online data, thus allowing direct access to data for its customers. The UK Ordnance Survey exercised in 2001 was entrusted to Infoterra. To conduct the survey data was handled using PostgreSQL. The new data acquired from this survey, at the end of the contract amounted to around 1 million metadata records and no operational issues were ever reported. These performance results built a level of confidence within Infoterra, which in turn decided to move larger datasets on the Open Source RDBMS. Infoterra decided to merge all their datasets into one mine, labelled as the "GeoStore".

Infoterra's own GeoStore is a combination of PostgreSQL with the spatial extension (PostGIS) as the backend, and UMN MapServer acting as the front end for the task. This Open Source collaboration is described as the most "cost-effective" way Infoterra could find (Elliot, 2006). Infoterra's high end online data base servers hold not less than 30 terabytes of aerial, satellite, and vector data, stratified in 600 million rows. Another 1000 terabytes of offline data is kept stored on tape storage for mining purposes. Infoterra's dataset can be

[8] Infoterra Website : http://www.infoterra-global.com

regarded as one of the largest commercial GIS in Europe (Infoterra Ltd., 2008), thus a main concern is to build special tools to handle such volumes. Despites these volumes, the loading time of their GML data to PostGIS takes just 12 hours, thus loading 14,000 features per second (Elliot, 2006). Operating its online "GeoStore", Infoterra is one of the largest Geospatial online markets on the net. However their business is extended to offline business, satisfying customer needs on other media, be it tapes, CDROM, DVDs and so on.

UC Davis Soil Resource Laboratory

The next scenario looks at an American research laboratory[9] focused on soil science. Back in 2004, the researchers started on a project to make data available to the public. Most of the raw data was collected by the US Department of Agriculture most of which at a 1:24000 scale (Beaudette, 2006). At the start of the project, the soil survey for California, Nevada and Arizona only consisted of almost half a million polygons, and their associated attribute tables. From the beginning of the project, it was always the interest of the lab to use as much Open Source software as possible (Beaudette, 2006).

Their first attempt to solve their concern was using the UMN MapServer as their front end and MySQL as the backend DBMS. As discussed earlier, MySQL still lacks full geospatial management, despite handling simple features, thus this resulted in a limitation to the project. MySQL being limited in its geospatial objects' handling, was used to store the attributes tables, whilst the spatial data was stored separately into layers. To minimise data loading times, it was decided to create new layer files for every survey area. Despite the combination of the back/front end gave the expected results (thus presenting data to the user), it had certain drawbacks that quickly triggered the need to seek an alternative.

The main drawback identified was that since the data was available only on a per survey area, moving outside the boundaries of a data collection, one could not pass to another data set. This limitation gave the impression that data ended with the survey boundaries. Another problem was encountered in the way to manage the 120 separate data sets each with its own survey area. In 2005, TIGER data was added and this added the challenge to manage survey areas.

[9] UC Davis Soil Resource Laboratory Website: http://casoilresource.lawr.ucdavis.edu/drupal

In 2006, the lab decided to collate all data into one source, and changed their option to use MySQL and Shape Files to using the Open Source competitor PostgreSQL with its own spatial extender: PostGIS. An important tool in this process of inputting the Shape Files to PostGIS objects was the ogr2ogr, another open source project. This tool forms part of the GDAL library[10] and the FW-Toolset, both Open Source software started by Frank Warmerdam.

Results of the new collaboration between MapServer and PostGIS were quickly noted. Having just one table for all the 120 surveys allowed much easier management of data. Another advantage over the previous set up was the capability of using the "simplify" PostGIS function. Such function reduces precision of certain layers, thus resulted in a substantial improvement in querying data (Beaudette, 2006).

Following the merger of the data sources, the target shifted from managing the data to mining the data. The GIS now can be approached via SQL statements, thus, no survey area has to be treated in isolation with the rest of the data set. According to Beaudette (2006), this move resulted in the "*largest improvement in the last six months*".

Dutch Directorate for Public Works and Water Management

Back in 2003, the Dutch Directorate for Public Works and Water Management started an Open Source project, named 'Geoservices'[11] to centrally manage all the data collected and make it available on the Internet. A problem faced before the launch of this project was how to combine all the data being from different sources into one data set.

The main concern was to set up a system based on Open Standards to allow for interoperability and use of Open Source Software. This was allowed to ensure independence from suppliers. OGC Web Services Architecture was used as the foundation for their development. These standards allowed the organisation to start tackling two of the main problems. The diversity of available web based GIS Software, including in the Open Source

[10] The GDAL Library (http://www.gdal.org) is a transistor library for raster geospatial data. This library is released under Open Source license.
[11] Geoservices Website: http://www.geoservices.com

sector and the duplication of geo-data from different applications resulted in very limited to none interoperability (Folmer, 2005). The directorate commissioned the job to sub-contractors, and the main task to build the Geoservices architecture was entrusted to Geodan[12].

The need for Open Standards was felt. An effort to have all the different data sources respecting the Open Geospatial Consortium (OGC) was made. The OGC Standards incorporated in this project are: Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS), Styled Layer Descriptor (SLD), Web Map Context (WMC) and Geography Mark-up Language (GML). In addition, the Metadata in this project was standardised. This project decided to adapt ISO 19115 for metadata describing the geospatial objects and ISO 19119 for metadata describing the services.

Now that the organisation had both its data and metadata standardised, reviews on the software to data management started. Similar to both previous case studies, UMN MapServer was used as the Web-Application GIS. For the clients inside the department, the Chamelon environment was built upon MapServer. Other components that were used to set up this platform are Apache, PHP, Tomcat and Linux; all Open Source projects.

From this project, several lessons were learnt by the directorate (Folmer, 2005). As one can see from the success of this project, using Open standards overcome many interoperability problems. Having all the sources abiding by the OGC standards, the organisation made sure that data can change formats but not value (Folmer, 2005). Using Open Standards, little to no time was dedicated to the validation of data, thus allowing more time to be used on productive purposes such as knowledge gathering. According to Folmer (2005), some data was hard to standardize, thus certain “workarounds” were needed. It was necessary that the management was informed about all the cases where a workaround was needed, so as to incorporate them into the standards. According to Folmer (2005), another argument why this project was successful was thanks to the Open Source Community. The developers and the user base in the OS Community are eager to share information and many are willing to

[12] Geodan (http://www.geodan.com) is a company with 20 years of experience in the field of geo-information. They are consultants and project managers on GIS software, Geo-Services on the Internet and LBS products.

help in bug fixes or propose workarounds. The Open Source Community allows software to be enhanced. From a business point of view, the directorate benefited also from a diminished dependency on suppliers.

AutoGuard S.A

AutoGuard SA[13] (previously AutoGuard & Insurance Sp. z o.o.) is a Polish company that started operating in 2000. The main services of AutoGuard offers are telematic services, using novel technologies in fields of wireless communication. This organisation makes use of a mixture of Open Source software and practices with Closed Source. The clientele base is mainly organisations from the East, and varies from private car owners to fleet management companies. One of the main services offered by this organisation is routing software for vehicles. Other products AutoGuard produce are special terminals and mobile GPS tracking devices.

Marcinkiewicz (2007) explains how the company monitors more than ten thousand cars in Central and Eastern Europe. The data released by the organisation in 2007, amounted to 400 GB, and more than 27 million of vertices. The largest data set the company owns reaches up to 10 million edges, displaying Germany's road map.

Before making use of Open Source products, AutoGuard used to make use of proprietary software developed by a Polish vendor, and Web Applications. The service rendered by this software was described as "slow and costly" (Marcinkiewicz, 2007). It was decided that Open Source software should be tested and the results compared with those of the proprietary software. To allow faster data retrieval, data is cached on regular intervals, and the organisation allows a 0.5 Terabyte of space for cached data. However, most of the low scale images (1:4000 – 1:120,000), are generated on the fly, since tests showed that retrieval takes longer time than generation (Marcinkiewicz, 2007). According to Marcinkiewicz (2007) experience, in the case of low scale images, the demand for edges lies between the ten thousand and hundred thousand figures.

[13] Autogard Website: http://www.kompass.com/en/CZ001863

Similar to the earlier projects, the choice for a Web Application fell upon the UMN MapServer project. As a backend for the data, PostgreSQL with its spatial extension PostGIS was used to process the data sets. The results were exported from PostGIS to MapServer using a closed binary data format written in house. Enhancing the service offered by MapServer, KaMap was used. KaMap is an Open Source JavaScript API very similar to Open Layers used in this project, providing the digitalised maps to be panned, zoomed and adding interactive features as much as possible. Back in the office, QuantumGIS (QGis) was used as the desktop application. Datasets for a testing session can easily load a layer file with 400 MB of geodata and another 400 MB of metadata (DBF). Marcinkiewicz (2007) argues that QGis was by far out-performing the proprietary software used earlier in their organisation, since it can be easily extended using Python and C++.

The move from proprietary software to Open Source was an inevitable move. Despite the amount of licensing being saved, Open Source software allows the organisation to be independent from what the supplier offers, and be able to change software much easier without infringing any rights. More recently tests have started on MapNik, another Open Source Web Application software, which is being tested against the performance of UMN MapServer. MapNik also supports OGC standards, so changing from MapServer to MapNik in case the latter is found to perform better will be straightforward (Alves & Davis, 2006).

High Country Software Ltd.

Back in 1996, two New Zealander brothers, founded a company with the aim to set up a nearby engine (a software which finds nearby locations to a point) and provide related services. The engine was later code named: Mobilemaps[14]. The main scope of such an engine was to establish an open, global network for finding and presenting nearby information over the Web. An important role of this engine was to present relevant Web pages depending on the local position of the searched item (Abrahamson, 2003). For example, searching for "London bar hotel" will result in a map of London with links to Bars and Hotels in London. The combination of spatial data and keyword searching made Mobilemaps different from many search engines which normally had no knowledge of location and ability to Web search. Using 'NearbyAds' technology, Mobilemaps was

[14] Mobilemaps website: http://mobilemaps.com

enhanced with an advertising plug-in that allowed companies to buy advertising space in particular locations.

Before the two brothers set up the company, in 1995, they developed a Windows software which presented a three-dimensional map viewer. According to Abrahamson (2003), Open Source operating systems were the cheapest web hosts. Their Windows software quickly started encountering problems and a mutual decision was made to start developing on Open Source system as well.

The plan to start developing on Open Source system was to enhance the Windows client and start developing a Location Search for the Web. Open Source removed any connections to Microsoft's development, thus any changes to libraries published by their pre-supplier did not reflect any more on their business plan. Another decision which the brothers took was that the end product would be released also under Open Source. Their business plan was drafted to generate income only from targeted advertising and little to none from the actual sale of software.

Since the project was to be developed using Open Source standards, cross platform compatibility was easy to conduct. Choosing Apache Web Server was an easy choice especially when even back then it still had a bigger share than all the rest combined together (Abrahamson, 2003). Other Open Source software used in this project was: UMN MapServer to act as their web map server and GRASS GIS to handle layer files.

The developing methodology used for this project was an iterative development, so they built a MySQL DBMS for their first tests and were ready to modify later on during the development. At the time, PostgreSQL was already venturing in the Geospatial area, but the brothers decided to keep the DBMS usage to relational attribute data only.

The main technological question arose when plans to start developing the software were drawn. How could they mimic the "relevance scoring" performed by the likes of Google, and other search engines? In text base searches, pre-sorting results is a technique to fix such problem, but when it comes to geo-location, the query is much harder, due to its constantly altering attribute (Abrahamson, 2003). In their initial prototyping stage, to eliminate such a situation, the results limited to a defined area, but they soon realised that as the data set grew, the defined area had to keep shrinking to maintain decent speeds (Abrahamson, 2003). The brothers were not pleased with the performance their engine was producing so they invested some research on the solution. The technique used to solve this problem was the “Peano algorithm”. This technique creates an index by running in zigzag pattern around the data. They dealt with their problem by creating two indexes on top of each other with a slight offset so as to capture as much indices as possible and collect proximity data (Abrahamson, 2003).

Another key hurdle in the process was geo-coding. The software needed to be able to translate street addresses into a geographical location, as part of the search parameters. Since most of the clientele of Mobile Maps is based in United States and United Kingdom this problem was quite a trivial one, due to the use of zip code, which the Web Spider technology could crawl through. Data crawled by the web spider technology can be shared between the clients once it is collected (Abrahamson, 2003).

After three years of development and testing Mobilemaps was released to the general public in its first non-beta build. As explained earlier, the main target clientele of Mobilemaps are advertising companies that pay for each time their advert is clicked. By developing software using Open Source models, both as building material for their own software and as allowing Mobilemaps to be released under Open Source licence, High Country Software Ltd. has benefited in different ways. Besides from saving money on licensing, the company now is free both from any obligations on their supplier and limitations due to cross platform operations. According to Abrahamson (2003), another advantage enjoyed is that software is being finished in lesser times than before, besides having the community issuing needed patches in a matter of days.

3.8 Case Studies

Previous case studies provided a number of scenarios, where Open Source products were used. This section deals with case studies when to use Open Source Software in GIS.

Bonaccorsi & Rossi (2005) conducted a research on 146 Italian firms in order to establish why they chose Open Source initiatives within the company. Bonaccorsi and Rossi (2005) were trying to find alternative answers to the monetary gains obtained. Most companies offered their free Open Source collaboration in order to gain a reputation within the OS peers. The more a community is geared towards a project initiated by the same company, the more trust the company builds, which in turn has an effect on its consultancy efforts. Such a claim can be supported by one of the main developers of PostGIS, when in a personalised web-blog he discusses why Refractions Research embarked on their spatial extension to PostgreSQL. According to Ramsey (2005), PostGIS enables the company to attain enough trust to attract clients wanting consultancy on products, even competing with other companies consulting using Open Source models too, for the simple reason that their advice is coming straight from the source. Another advantage Ramsey (2005) views important for his company is that the clients increase their level of trust with the company becoming more reliable.

Eric S. Raymond (1999) continues to further deepen the subject and stratify different types of business models a company in Open Source venture in. The first model proposed is the “Loss-Leader / Market Positioner”. In this model the organisation creates open-source software to create a market position for proprietary software that generates enough money for both projects. An example found in this research is when a company creates software and charges money for a server licence, like for example Netscape did to stop Microsoft with their Internet Explorer taking over the HTML standards.

Another model discussed in the study by Raymond (1999) is the “Widget Frosting”. This model is more focusing on hardware rather than software. In such a scenario it is a must for a company to open its source, so as to allow third party companies to provide the best support they can with the hardware. Raymond (1999) continues to mention the feature of future-proofing, where the open-source can still allow clientele to patch the software even if the organisation officially stops.

The "Give Away the Recipe, Open A Restaurant" model is another model discussed by Raymond (1999). The only difference from the first model discussed, is that instead of opening the position for closed software, the model opens for services. This model can be seen being operated in the Server Licences one can obtain for the likes of Red Hat and MySQL. The software in such cases still remains Open Source; however the client would be buying the support rather than anything.

The fourth model proposed by (Raymond, 1999) is to generate revenue by selling related things to the Open Source software. This business model can lie in a broad area, starting from selling marketing items such as T-shirts to other software-related procedures. Companies can issue professional-documentation against a market price, or plug-ins to further enhance the software only as closed source. Another idea can be to provide customizable patches for a client on a closed source basis.

Another model discussed by Raymond (1999) is the "Free the Future, Sell the Present" model. Adapting this business model, an organisation release the software already compiled and have the source as a closed source. However an accompanying license can state that the source can be freely hack-able after a stipulated time frame, or if the vender folds. This license would allow the clients to know that the product is customizable to their needs, because they already have the source. An organisation that implemented this type of model is Aladdin Enterprises who released their Ghostscript program under a similar license.

"Free the Software, Sell the Brand" is another business model speculated in this paper. Open Sourcing the software a brand can issue certifications that other technologies wearing the same brand are compatible with their software. Sun Microsystems are handling their Java and Jini products utilising this module.

The last module offered by Raymond (1999) is the "Free the Software, Sell the Content". This model presumes that the data in your software is valuable and can generate money for itself. By using this model, a company would release the software for the Open Source community and sell the content. The more the software expands the market, thanks to the third party developers, the more the market expands thus more possible clientele for the content. This model was used by AOL when the company Open Sourced its software and kept generating the money from its content.

However Camara & Onsrud (2004) states that it is "very difficult for a company to develop a world-class FOSS4G" software and base all their income and subsequently profits on consulting fees only.

Back on the research findings by Bonaccorsi and Rossi (2005), the main reasons why an organisation chooses Open Source methodology can be split into two, Extrinsic and Intrinsic motivations. Extrinsic motivations include reduced licensing costs, and reducing testing costs by relying on tests performed by the Open Source community (Nasr, 2007). The intrinsic motivations, according to Nasr (2007), are mainly due to standards that most Open Source software use. Relying on standards, puts at ease the minds of many businesses since their software can easily be moved on different platforms without any trouble. Also businesses want to make sure that their data is as much as possible not bound to one platform, and feel the need to not being dependant on their supplier / data manager (Nasr, 2007). Bonaccorsi & Rossi (2005) shows that only 18.5 per cent of the interviewed companies use Open Standards. The rest, in some degree or another, either, do not inject anything back to the community, or else are simply inconsistent in their activities, and disrespect the licensing in a way or another.

A deeper detailed look upon how companies deal with Open Source GIS mentality follows. GIS vendors tend to prefer to use Open Standards for their projects so as to be able to conform to any data out on the market. Looking back at the section where GIS DBMSs are heading to, one can find out that most of them are trying to confirm to the OGC Standards. Such standards changed a lot how GIS vendors prepared their software, from the proprietary format published by the long established ESRI. In 1994, the Open Geospatial Consortium was set up, mainly on the basis to challenge these closed formats, and provide a standard widely available and attainable format to all. Here it is worth mentioning that the standards followed by OGC were first laid down by the Open Source GIS; GRASS back in the 1980's. Open Source community quickly accepted the standards proposed by the OGC and started developing software using such. This process helped also refining the standards. A big leap forward took place when the Simple Feature Access was added to the standards, and quickly started being used to other GISs, including proprietary ones. Another important research conducted by Horanont, Tripathi & Raghavan (2002), published some results that shed a different light on how proprietary software viewed Open Source GIS applications: UMN MapServer (as the most used open source Web-data publisher) was compared with MapXtreme by Map Info Corporation and ArcIMS by ESRI. Since MapServer is built around the OGC Standards and makes use of the OGR Library, it needs no conversion tools so as to display data on the web, unlike the other two. This gave an advantage to organisations to allow them to make use of different data sources, without the need to inter-operate the data, if all data follows the standards. Also another notable output from the research, is the availability Open Source software gives to the developer to extend functionality with a broad range of programming languages. MapServer's own MapScript can be extended using different language unlike the other two proprietary ones.

Statistics always improve the picture given to any research. A questionnaire performed by Nasr (2007) gives some insight into how organisations, in the public and private sector view Open Source GIS. The organisations questioned were invited in an online survey. From the findings one can easily point out that in organisations it is the software developers who know what are the concepts of Open Source, and little to none knowledge is expected from decision makers. Nasr (2007) claims that more than 80 per cent of the organisations choose Open Source because of the absenteeism of licensing fees otherwise.

Another interesting finding is how most of the developers prefer Open Source to other options due to the fact that they can easily modify the source code, which values reach the 70 per cent figures. These two figures show that lack of licensing fees do play a very important role why Open Source is chosen. In the GIS environment, according to Decision makers, licensing fees come second, and the first spot is taken by: "Support of a variety of web clients". To Decision makers, the scalability of the product is ranked at the fifth place in importance of decision. Also the findings show that decision makers, in a sentiment indicator, seem to have a negative image of Open Source software, due to what they see as inappropriate support. Since decision makers are the ones who decide what software is bought and developed, such ideas handle the use of Open Source software, and this ripple effect on why there is a “poor adaptation of open source GIS tools in mainstream businesses” (Nasr, 2007).

Both decision makers and software developers regard the market share of the GIS to choose as the most un-important factor. Another interesting finding from Nasr (2007) is that users usually have very high expectations from a GIS, much more than the developers and decision makers do. WebGIS users almost balance each other as to what are the main requirements expected from the system. There is a very slight difference between what the users search in a WebGIS, basically between the availability of doing spatial queries and routing analysis between two points. Both figures are around the 50 per cent mark. However Neis & Dietze (2008) regard routing as more important for WebGIS applications. Further questions in Nasr’s (2007) document show that almost all the developers answered that the functional requirements are changed during the building of the project, this being due to a result that the client start actually seeing what the application can offer.

OGC Standards play an important role in why a GIS is chosen over a competitor. In a question asking what roles do standards play in the decision, more than 72 per cent of the answers shared the Agree / Strongly Agree mark (Nasr, 2007). Allowing software developers the choice to use their preferred language, has put Open Source GIS software top of the range in the question if the languages supported do play a role in deciding which application to use. Another 73 percent shared the Agree / Strongly Agree answer in this question too.

3.9 Looking at the future

After a deep insight on how the technology adapted to the needs of the industry, an insight on the future is presented in this section. One of the main objectives of this research is to provide a dynamic network adapting to traffic conditions. Such an aim makes reviewing plans of what the future might reserve a must. Various studies were conducted, such as the report issued by the UK Dept. Trade and Industry (2006) where the future of transportation systems, and how GIS tackle traffic were discussed.

The UK Dept. Trade and Industry (2006) issued a report which presents the future till the year 2055. Despite being a study focused on the United Kingdom, general information can still be quoted on how the challenges and opportunities can be bridged together so as to have intelligent infrastructure. The future always reserves uncertainties, and this document takes two variables as the main concerns. The first one being the development of a low-environmental-impact transport system, and the second is the acceptance of intelligent infrastructure. The increasing concern over environmental impact, and climate change might have serious implications on the future in the next 50 years.

The document outlines four possible scenarios which the industry might develop. These scenarios are not mutually exclusive and it could possibly be a mix of one or more of these scenarios. Such scenarios are labelled: Perpetual Motion, Urban Colonies, Tribal Trading, Good Intentions (Dept. Trade and Industry, 2006).

The **Perpetual Motion model** assumes a society driven by constant information, competition and consumption. Also it is important to note that in this model there are no energy deficiencies and by 2055 society would have found a cheap, clean, preferably low cost energy supply. This scenario assumes enough technological information which will lead to reduction in emissions. The concern of this scenario is that mobility will still be high and congestion present in many areas, thus the need for routing software would be greater than ever.

The **Urban Colonies scenario** (Dept. Trade and Industry, 2006), tackles the British society as a model where technology is primarily focused on minimising environmental impacts, thus making use of sustainable buildings, and new urban planning. Transport will have to adapt to be energy clean, and huge rise of Public transport which runs on electric and low-energy. In this scenario the IT supporting the transport systems, is very limited, and is kept to a minimum.

The **Tribal Trading model** starts from a post-scenario situation (Dept. Trade and Industry, 2006). It shows a world which just suffered a global recession which left millions unemployed and has just come out from an energy shock. This environment continues with a view that local transport is typically by bike or horse, and public transport is seen as a luxury. Building intelligent infrastructures using technology is by far out from the agenda of the governments, and the main concern of all is how to improve the situation from this post-trauma scenario.

The **Good Intentions scenario** (Dept. Trade and Industry, 2006) describes very environment-aware governments, without any new alternative energy sources. Emission controls are high on the agenda, where intelligent cars monitor the environmental cost of a journey. A tracking of all private cars is done by the government so as to be able to tax for carbon emission. The need of Geographic Information Systems is felt from everyone which incorporated with location based services allow the car onboard systems to adjust speed to minimise emission. An efficient public transport system is widely used.

As one can see from these four possible scenarios, but with the exception of Tribal Trading, the importance of (Geographic) Information Systems is felt throughout. Location Based Services are introduced in Government's policy to help both society and governments themselves. There is still however very little knowledge on how society will be affected with regard to privacy. This issue has featured extensively in many discussions over LBS. There might be concern over adaptive routing being used by third parties for advertising purposes, or routing adjusted to benefit only a stratum of the society. According to Altman (2008) privacy control should always remain at the discretion of the user. This is done by having the user fully aware with who s/he is sharing the data and also when. Such control should come in permission based where the interface should be easy for the users to understand and manipulate.

3.10 Identifying the choices

In this section a brief introduction regarding Open Source GIS software, will be conducted. The introduction starts off with a review about the Geographic Information System and continues on with specifying details about other useful.

3.10.1 Geographic Information Systems

After comparing the main DBMS's available in the market (Table 2), it was decided to shortlist two for comparison, one coming from the Open Source field and one from the Closed Source; PostGIS (which is a DMBS enabled with Spatial Objects) and ArcGIS. Several reports have been reviewed with respect to these two DBMSs and the findings are being summarised later on. PostGIS is an extension to the popular Object Oriented DBMS PostgreSQL. PostgreSQL supports native interfaces for ODBC, JDBC, .Net, C, C++, PHP, Perl, TCL, ECPG, Python, and Ruby. Besides this native interfacing, PostgreSQL can be extended using language specific procedures. PostGIS can also interface with Geographic Resources Analysis Support System, better known as GRASS. ArcGIS, which is proprietary software of ESRI, is an integrated collection of GIS software products for building a complete GIS.

Such projects were involved in handling the data from the original source, thus Vector Files to the objects stored in PostGIS. Each of this software used is going to be briefly discussed further below in this section, identifying the role each one played with data manipulation.

GRASS GIS

GRASS GIS[15] is an Open Source project which has the ability to process 2D/3D vectors and support for vector network analysis. GRASS GIS 6 now enables data to be stored in PostGIS, so it became the natural choice for my implementation. Similar to PostGIS, GRASS GIS works on a wide range of cross platforms (GRASS Development Team, 2008). GRASS is currently used in academic, commercial and many other environmental consulting companies (Smith, Goodchild, & Longley, 2008). The GRASS GIS comes also with its own user-interface and also a text-interface version.

pgRouting

Another Open Source extension to PostGIS enabling routing techniques is pgRouting. pgRouting[16] comes with three different routing techniques;

- Shortest Path Dijkstra -- routing without heuristics

15 GRASS GIS Website: http://grass.itc.it/

16 pgRouting Website: http://pgrouting.postlbs.org

- Shortest Path A* -- routing for large data sets with heuristics
- Shortest Path Shooting Star -- routing with turn restrictions

By the term 'heuristic' it is understood that the techniques used to determine the road plan are readily made algorithms, which despite being loosely applicable can be used to solve problems (Pearl, 1984). The main functions of pgRouting will be used so as to enable the users to plan the route depending on the desired option thus taking into consideration elevation and turn restrictions.

3.10.2 Desktop GIS Tool

Quantum GIS commonly referred as QGis[17] is another Open Source project. QGis can act as a frontend for GRASS GIS; in fact it comes with a default plug-in to incorporate GRASS. The plug-ins available for download allow the user to manage GRASS data better and in an easier way, and by using OGR tools data can be shifted between different datasets, and data forms, be it vector, raster, PostGIS layers and others (QGis, 2008). Continuous support and development, combined with cross platform operation has led this application to be one of the leading front end Open Source applications (QGis, 2008). Meetings are conducted online via their IRC channel on FreeNode.org and the team really does appreciate users input for their next release.

3.10.3 GIS Web Server

MapServer[18] is another Open Source project developed by the University of Minnesota, having early contributions by NASA. The main use of MapServer is to act as a data renderer thus making data stored on the server available to the general public via Web Interface (Lime, 2008). MapServer can be enhanced for more functions using MapScript which is a scripting language availing the users to script in their favourite programming / scripting languages such as PHP, Python, Perl, Ruby, Java, and C#. MapServer supports different data standards, naming the main three standards: OGR, OGC and GDAL (Lime, 2008).

[17] QGis Website: http://qgis.org
[18] MapServer Website: http://mapserver.gis.umn.edu

OpenLayers

OpenLayers[19] excels at displaying map data in JavaScript supporting Web Browsers. OpenLayers is a user side script, thus, there are no cross platforms restrictions. OpenLayers code is released under the BSD Licence and is also Open Source (OpenLayers, 2008). Making use of API's, OpenLayers consent the developer to load any mapping service possible, not just those loaded by MapServer.

TileCache

TileCache[20] provides a caching mechanism using Python CGI scripts. Using TileCache, the maps generated will be split into tiles and cached on the server's hard disk, thus reducing the computation time by a high fraction (TileCache, 2008). OpenLayers can then be used to call the tiles and serve them to the client at a much faster response time. TileCache is another Open Source project released under the BSD license. When the layers are generated from static Vector Layers, not on-the-fly, the result is not of a big difference however Schmidt (2007) claims that a small improvement is still seen.

3.11 Summary

The literature review gave an insight on Open Source development combined with Open Standards methodology. The case-studies provided the necessary support for the development of the project, in using OS. It is however unjustified to claim such praise to Open Source when counter arguments have also been published. One could also get an idea on how different organisations, be it in the private or public sector, took on an Open Source project as their main software for running the business. The research and case-studies in this chapter will help the rest of the project by supporting the decisions on which software would be chosen to compute the problem. Also methodologies will be studied so as to come up with a design methodology for the setting up of the project. Lee & Lee (2006) concluded that despite the Open Source GIS Software they reviewed was very mature; such software did not offer all the functionality which proprietary software can offer. Over the years proprietary software has matured and has a number of very advanced features which are

[19] OpenLayers Website: http://openlayers.org
[20] TileCache Website: http://tilecache.org

unmatched by the Open Source community. Also, Lee & Lee (2006) found that Open Source GIS map production tools are more challenging than proprietary ones.

This chapter reviewed also how studies were carried to research scenarios where a new GIS-T was being planned. Studies have showed that when planning a system, the scope of the project has to be identified by all the stakeholders of the system. Two types of users were identified, and studied, being Emergency Response Teams and the end drivers. It was discussed that driving habits reflected on fuel consumption (Ericsson, Larsson, & Brundell-Freij, 2005). Another study showed that automatic routing can help decrease fuel consumption (Smit, 2006). Routing algorithms were studied also, focusing on shortest path techniques, identifying the similarities and differences.

Following the literature review, the methodology chapter follows with a plan of how this project should be conducted. A discussion on what are the key factors of the system, who are the prospective users and the main objectives of the system, will follow.

Chapter 4 : Methodology

4 Methodology

4.1 Introduction

Taking into account the literature and other similar applications, a specific methodology for this project is outlined. This chapter starts by reviewing the project requirements, then discuss and propose the methodology used followed by discussing the variables in the system. Once the inputs are discussed, this chapter discuss how the system is proposed to be built, how the algorithms were computed and finishes by introducing the testing mechanisms.

4.2 Functional List

Recalling what was written in the introduction chapter, the project objectives are:

- To design and build a Geographic Information System that will act as the backend for the system offering these services. One must note that computation limitations exist.
- To allow live data being fed directly to the system. This objective will be possible by enabling the network to ping different sources, be it road network monitors and weather stations, to allow for new data feeds. This will make sure that route planning is always computed on the latest data available.
- To build a Location Based Service for mobile customers which act as an intelligent route planner. Such a route planner allows the user to specify which criteria is the most attractive to be used depending on the main variables in the system (time, consumption and distance) and build the Information System around these variables.
- To identify the main variables in the system (time, consumption and distance) and build the Information System around these variables.
- To identify the main objects in the system, thus which vehicles form part of the system and how these are grouped.
- To design and build a Web-Interface front end for the users to communicate with the backend. Such a Web-Interface should be accessible from mobile devices for the route planning exercise. Another front-end using Open Source readymade software will be used to allow Street Management in case congestion is reported on a street. One must note here

that Congestion reports will be thought of as being reported by third parties and the Street Management interface should update the current status of the road.

- Once the project is done, it is aimed also to propose suggestions for useful additions and improvements.

4.3 The Proposed System

Before comparing the project's methodology to other proposed projects in the past, a very quick overview is given on the expectations from the system. A very simplistic approach is given to outline the key terms in the project, so as to serve as a reference point further on in the document. As Figure 10 shows, there are two different linked Internet users, those who will be communicating with the system over a wireless internet connection, and those which will be desktop users. The system will be split in a three-tier architecture, having a Presentation Layer, Business Layer and Data Layer as developed by Cheng et al. (2004) (Figure 11).

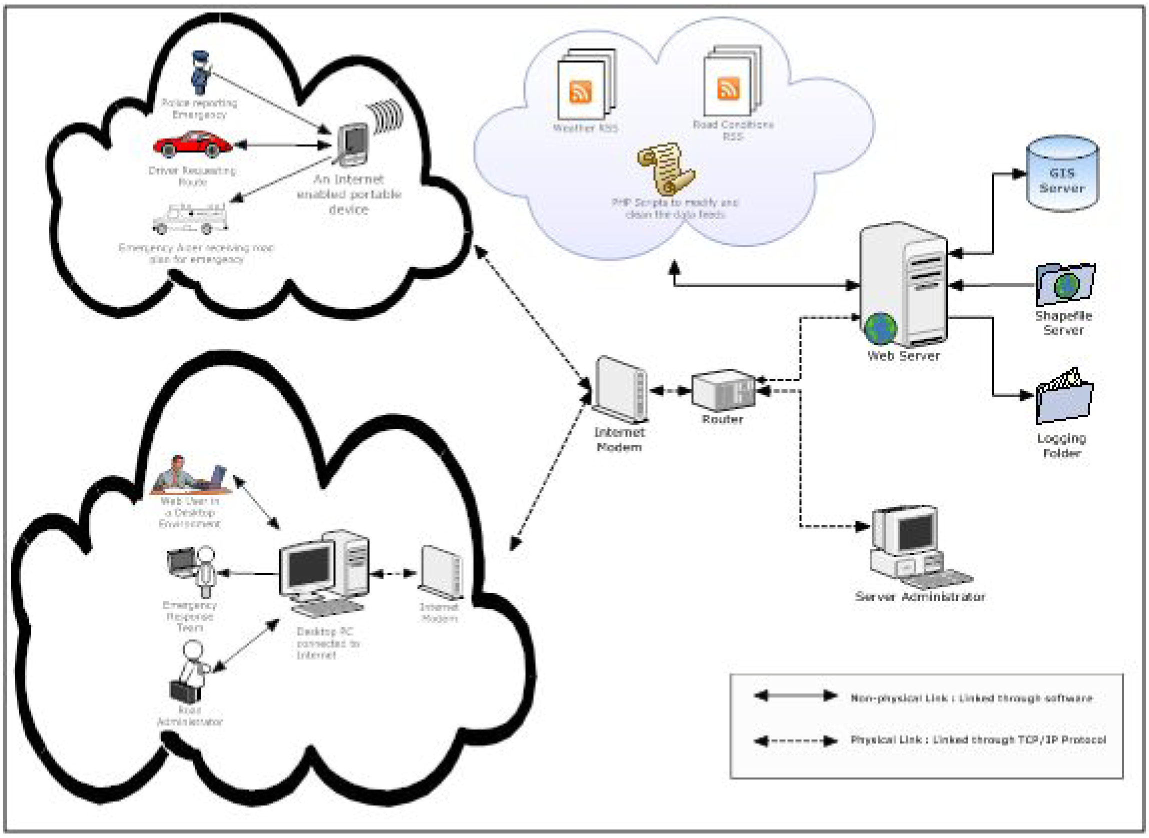


Figure 10 : System Overview

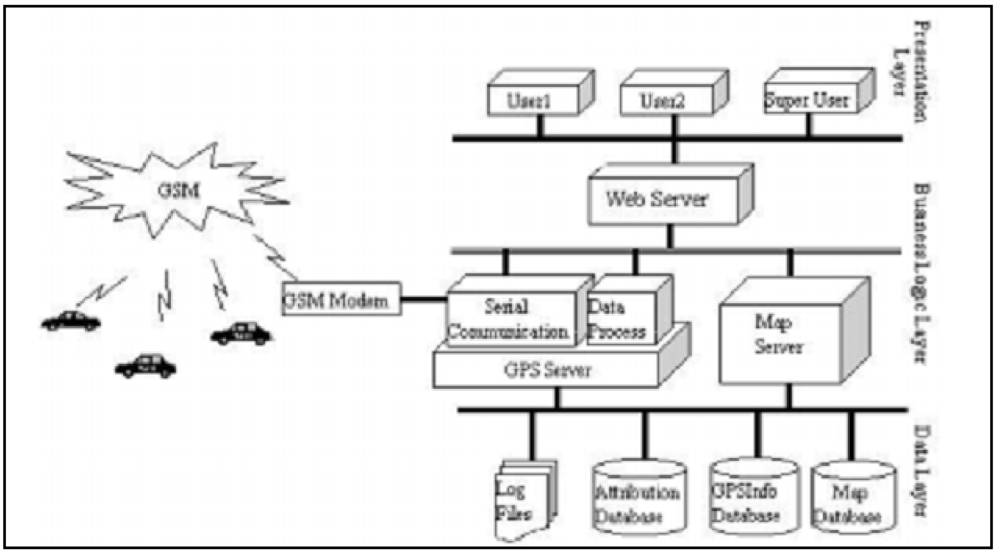


Figure 11 : Three-Tier Architecture (Cheng, Yang, Shao, Liu, & Gao, 2004)

Different options will be presented to the user by the system depending on the logging environment, and needless to say every user type will have own credentials on the system. Figure 10 shows that access to the Database Server (storing both spatial and non-spatial data) is limited to the Web Server. This means that direct Database access will be limited only to the DBMS Administrator (for obvious reasons) whilst keeping all the other access, through scripts residing on the Web Server. The web server will also hold scripts which will be called at agreed times so as to be able to update the GIS as needed. Once again, access to update the road network from the CRON jobbed lists will be attained only through the Web Server, acting as a Gateway. In this overview, functionality is based on a top-down approach, just outlining what the system needs to do, with little to none detail on how it is proposing to do it, and up to what extent.

4.4 Reviewing the Methodologies

In this section, relevant frameworks and methodologies will be reviewed. Also we will be discussing their usage and their similarities with the system proposed in this research. Up till 2004, many studies looked at cost-analysis, with few tackling the real technological advantage proposed by an automated vehicle routing system (Hayashi & Morisugi, 2000; Vreeker, Nijkamp, & Ter Welle, 2002). Li & Liu (2004) propose a framework that focus on social and environmental impacts and take into consideration the adaptability of a system to a change in travel behaviour. In this study, Li & Liu (2004) discuss the different scenarios a system passes through depending on the time of the day, when different levels of throughput are present. The proposed plan was to route as much as possible heavy traffic outside from the urban areas to try to minimise such a negative impact.

Routing of Public Transport is a relatively new technology, according to Li & Liu (2004), which makes it quite hard to compare results of a framework with another. They also outline the unknown factor of long-term effects of a methodology. With unknown factor it is meant that a methodology cannot be trusted to give satisfactory results if the time period is a loose variable. It is important to audit system results following implementation. Such comparison can also be done with countries such as Japan described as "*the only country with the most advanced intelligent transport system in actual operation in the world*" (Iguchi, 2002).

Following suggestions on how to compare results of a methodology using simulation tools, presented by Chen & Stauss (1997), Ambrosino et. al (1999) presented a tool to manage energy consumption. Following this review, one has to mention also the review prepared by Ochieng & Sauer (2002), where they foresee a high increase in use of these technologies. In their article, they state, that given the high increase in the use of navigating equipment in vehicles, these methodologies will be comparable soon. This theorem has been confirmed in a study by Jackson et al (2007) where they commented on the high increase of navigating equipment and appropriate increase in the different methodologies.

Thong & Wong (1997) and Dueker & Butler (2000) have proposed a GIS-T model, where data can be shared between transportation offices. These models can also be regarded as good companions to travel demand models (Miller, 1996). Huang & Lam (2002) and Li & Kawakami (2000) go a step further, and outline that in such an approach it is imperative to include travel behaviour. One should also mention that the more knowledge gained from the users' behaviour, the more does a system reflect the true nature of the environment it is reflecting.

4.5 Background Overview

In this section, an overview of how the project will be carried out will be discussed. A number of steps have been developed to tackle all areas of the project. The main objective of this case study was to test different algorithms in routing and how the network, and route plans adapt to changes. Such changes are:

- Weather conditions: The road network need to adapt to the various options of weather;
- Road Conditions: These can be split into two:
    - Planned Road Blockings: Such blockings are due to road works or any other construction works in the said road segments;
    - Temporary Road Blockings: These include but are not limited to heavy traffic and traffic accidents;
- Vehicle used (engine and type of vehicle): Routing should adapt to the vehicle being used for the trip;
- Road gradient: Routing should also adapt to the gradient of the roads used.

Also a note on the dynamics of data is given, thus allowing the network to adapt to various changes in the data.

4.5.1 Data Acquisition Barriers

One of the main challenges in developing a routing application is to acquire good quality road data. Many of the data available on the free market is limited in detail, and proprietary data was too expensive to collect. However, some agencies are trying to provide useful free data (e.g. OpenStreetMap). In Malta, the road network layer available at the National Mapping Agency, within the Malta Environment and Planning Authority dates back to the early 90s with very few updates. In mid 2008, Google Maps[21] introduced Malta's road network (HiVoltage, 2008). Unfortunately this meant that it was not possible to use this data since it was already too late to change all the project.

[21] Google Maps: http://maps.google.com

Another problem is the need to modify data to the user needs. Data needs to be adapted to the specific application. Since in this project, gradient is needed, so as to disallow small vehicles to climb hills, topographic elevation is a must.

Another concern needed for the data is the turn restrictions. By turn restrictions it is understood restrictions on traffic flow, be it one way streets and also speed limits. Such restrictions are important in order to have a functioning routing system, since the GIS needs to give as much as possible the situation of the real world.

Meta data is also a concern, since it is this data that allows the user to make an informed decision regarding the quality of the data. Different descriptions can be given to data which makes it quite important to manipulate. This type of data can also be significant since it explains the application of when it was created and for what purpose.

Another concern with data is different sources. Data differs from one supplier to another. Data capturing, recording and labelling is subject to the organisation collecting it. Even if one conforms to the standards, one would still need to manipulate the layers so as to make them confirm the same projection and datum. All these transformations can modify the data, with a possible effect on its integration and accuracy.

4.6 Secondary Information

Extensive literature has been researched to establish a basis for this study. This information was achieved thru various sources, including journals, books and internet sites, especially dealing with Location Based Services during the last twenty years. Results of this review were presented in Chapter 3.

4.7 Planning the System

Once the planning was finished, and the Software Requirements document (Appendix I) completed, the proper development stage was initiated. After reviewing available methodologies to use, it was decided to use the DSDM methodology. Similar projects where

DSDM was used at AA (2002) and for cGML (2004). Time and money were outlined as the main constraints in the research. A quick introduction to the methodology follows.

4.7.1 What is DSDM? How is it used?

"Dynamic Systems Development Method" or as commonly referred DSDM, is a Rapid Application Development methodology, thus it is focused on providing modules ready for use in a timed constraint. Having user involvement as the main aim of the study, an iterative approach is chosen. Thus, software modules will be published on incremental stages, with the user keeping modifying the prototypes, until the final product is reached. DSDM was chosen since it is a framework that focuses on tight schedules and low budgets. DSDM is divided into three nominal phases, being the Pre-Project, Project Life Cycle and Post-Project (Figure 12). The Project Life Cycle is subdivided into five stages, which are iterated until the project is fit to pass to the next stage. These are:

- Feasibility Study
- Business Study
- Functional Model Iteration
- Design and Build Iteration
- Implementation Iteration

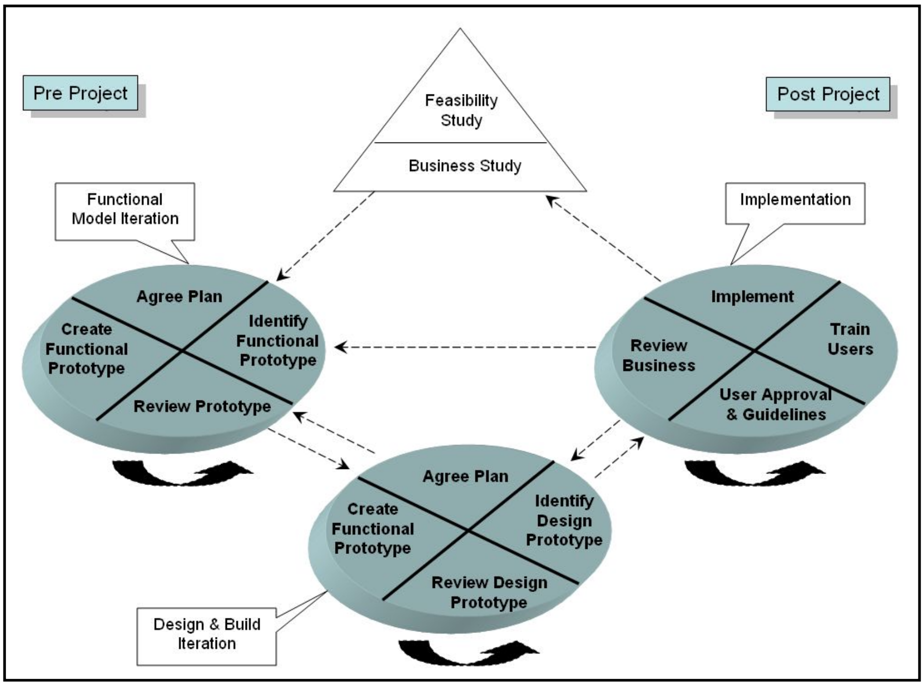


Figure 12 : DSDM Architecture (JupiterOnlineMedia, 2008)

The discussion now will take each of these steps and identify which steps were conducted in the process of building up the software and finalising the project.

In the **Pre-Project phase**, an outline of the specifications was prepared. Once the project was approved, a literature review process was conducted in order to study knowledge from previous studies. Once the specifications of the project were accepted, it was taught that the project can pass to the following stage.

As outlined earlier on in the Chapter, the **Project Life Cycle** is an iterative process, making sure that the users and developer of the project understand each other, so as to have a project used by the users, and tackles what the sponsor, basically the organisation which is paying for the project, is paying for. Since this project is an academic one, there is a thin line between the users and the sponsor; thus the developer will play both roles.

The **Feasibility and Business Study** stages can be combined in one for this project. Starting from the Literature Review, up to the Methodology Chapter, the Feasibility study and Business Study were outlined, forming the main idea behind the project for the thesis. Despite having no sponsor, the budget and time constraints were identified.

Prioritising Requirements

Must Should Could Would or as better known MoSCoW is a way of prioritising requirements in order to make sure that the core requirements are handed before others. “Must” requirements are the main requirements for the system to be productive, thus failure to present one of these requirements can be seen as failure to produce a working project. “Should” requirements can be seen as an important factor to have within the software, but do not affect the core job of the development. “Could” requirements are not so important, and are subject to budgets. Thus, if time is not pushing against development, these requirements could be computed. Failure to prepare these requirements does not affect the running of the system, but its usability. “Would” requirements are the least important of the four. These requirements will be added into the system if budget constraints are not pushing. Therefore, the system would operate fine without these requirements, but are seen as an added value if they are coded in. Table 6 shows how the requirements were labelled.

In the **Functional Model Iteration**, the requirements prepared in Appendix I start to be tackled. As explained earlier on in the document, this is an iterative approach, thus the use of prototyping is a must. The way this stage was handled was by making use of building small modules for the project, and testing them until a certain degree of satisfaction was achieved. An important step in this stage was the schedule fixing. This phase was important since it kept the process going and pressure to finalise the main product was kept to a minimum. The following sections can be seen as the main components in building the project. In every section, one can note the numerous times the word "test" is mentioned. This is due to the fact that testing is an important factor in this methodology. Each component of the system is heavily tested on its own, until everything is certified to work together, which can start being combined in the Design and Build.

Priority	Description
Must	To build up a Location Based Service for mobile customers which act as an intelligent route planner.
Must	To plan a route should allow the user to specify which planning has to be used
Must	To design and build a Geographic Information System that will act as the backend for the system offering these services.
Must	To design and build a Web-Interface front end for the users to communicate with the backend.
Must	To provide an updated route every fixed time intervals, thus allowing the user to have as minimal risk as possible to reach a street in which there is congestion.
Must	To allow queries to be planned against live data and thus minimizing risks of finding traffic jams.
Must	The Web-Interface needs to be compliant to W3C Standards. It must be correctly viewed on Gecko Engine.
Must	All software used in this project need to be Open Source and confirm to Open Standards.
Should	Such a Web-Interface should be accessible from mobile devices for the route planning section.
Should	To prepare different types of Users, each with a separate login and a login class.

Should	End User Documentation should be easy to understand and enriched with diagrams.
Should	Syndicate data from a Traffic and Weather update site.
Could	Make the Web Interface compliant to more than one engine.
Could	Introduce more types of User roles
Would	Use Maltese data instead of American one
Would	Instead of synthetic use real data, especially Traffic conditions
Would	Use GPS locators instead of a random point on the map
Would	Use more vehicle details to present a better scenario

Table 6 - MoSCoW Requirements

Once the usability and workability of the separate functions in the system is confirmed, the functional models pass to the **Design & Build phase**. In this stage, the functions created earlier on, start getting compiled together, and rigorous testing is done. During the testing phase, sections will not fully work with the rest of the system, and that is where the iteration takes place. The faulty components will be sent back to the First stage, for further development and testing, and retested again in the whole stage. The prototype remain getting back and forth until the key players in the system are happy with the outcome, which will then pass to the Implementation stage. Another important section in this stage is the creation of the non-functional requirements, which are reviewed in the Software Requirements Appendix I.

Given that the system is in the **Implementation stage**, it is felt necessary that certain functionality is discussed and typical case examples are documented in this section. Once the system is operational, and testing has been performed, an Implementation stage is normally needed. The User Manual, (Appendix II), needs to be illustrated with screenshots, and great care should be given to the least of the details, whilst making sure that it is easily understood by non-technical people. Similar to the Software Requirements, this document needs to be understood by users which lack technical knowledge.

4.8 Problem Definition

Location Based Services, supported by Geographic Information System are quite complex. During the input stage, care is needed in order not to obscure the data. During the processing stage care is needed so as to ensure that the utmost of the knowledge is gathered (Dueker & Butler, 2000) and finally to project the output in an understandable way to the user. The problem definition stage needs to outline what the project is expected to do. In this project, such a step was taken in a more official stage, with the preparation of the Software Requirements Document.

System Requirements were developed in order to identify and describe the system proposed for this study outline the main points which the system would need to abide by. The Software Requirements document can be regarded as an agreement between the different actors in the system. In Appendix I, one can view further details to what is going to be outlined below:

- Overall Description
  In this section, a list of features which are to be included in the project was prepared, outlining each of the variables which are to be tackled by the software. Since this system is a user driven one, a great deal of the discussion was around users, ranging for different user classes and characteristics. Differences in usage patterns and what type of growth is expected in the user base formed also a part of the discussion. The operating environment of the software featured also in the discussion with a focus on the constraints needed for the software.
- System Features
  In this section, the features of the system were discussed. Details on what the system should do with regard to routing were outlined. A further detailed discussion about the dataset followed, specifying the constraints and the steps involved to manipulate data. The User interface was also discussed detailing the differences between the desktop and the mobile environments, specifying the differences and similarities between the both.
- Other Non-Functional Requirements
  In this section a discussion about non-functional requirements was conducted. Requirements such as performance and also security requirements featured in this discussion.

Following on the software requirements presented, the methodology used for this project will be divided into five different sections. Each of the sections outlines the main steps taken in this research to present a framework which enables the user to be presented with the desirable route.

1) Choosing the Variables
2) Defining the Users
3) Choosing the Route
4) Routing the Emergency Vehicle
5) Obtaining Standardisation

This study has drawn upon a series of consultations (by telephone, email and IRC) conducted with different key players in the industry, both technical and practical personnel. Most of the interviews were conducted after reading works written by the said interviewees.

In the above list, standardisation is a generic pillar. Throughout the design and implementation of the project, standardisation, in particular Open Standards, is identified as critical, whether it is web interface, coding practices, or even choice of formats to present data. Open Source practice is a good showcase of making use of Open Standards. Some authors, such as Schwartz (2003) see source and standards as two distinct things, with Open Standards being more important. Others such as Werbach (2002) discuss that Open Source is built upon Open Standards and it is much harder to find the distinction between the two. Saint-André (2003) continues to discuss that Open Standards should reflect also on the Closed Source Markets, since many are the organisation that despite having legacy Closed Source practices, are moving on to the agreed Open Standards so as to facilitate development.

4.9 Choosing the Variables

From the start of the project, the main aim of this research was to deal with dynamic routing, thus routing which is based on a set of variables. This makes the choice of variables a key factor in the planning of the system. Based on the literature review, a number of variables affecting dynamic choice of the route, but which are not particularly used in the industry and research have been identified. Despite the obvious choice of some variables others were not so obvious. This section does not only base the discussion on what types of variables were chosen but also on how the data associated with such variables was collected and processed. In this section the discussion will be split into two, a discussion about the adaptation of the variables, how data was collected, what techniques were used to standardise data and a review on how these variables would play a key role in the decision making process.

4.9.1 Traffic Condition

The first thing that comes to mind when one hears about live data, is the availability of road conditions, thus the ease to use a road. Searching the dataset was a key role in the planning of the project. A research for a dataset that featured road gradient, different types of streets and data was conducted. Another key factor in the data set was traffic directions, that is, the identification of One-Ways and Bi-Directional segments. The data used was that of Wake County in USA which is licensed under the TIGER licence. As explained earlier, in order to have a proper live network, the road network needs to be altered so as to symbolise problems on the network, such as traffic accidents, road maintenance and other disturbances. This manipulation was possible through feeds from Weather reporting tools and Traffic Conditions feeds. For Weather Reporting, the website *wunderground.com* was used. This website provides an RSS feed updated regularly with the different weather in the different parts of the state under study. The reports from wunderground.com are not very specific, so it is left to a randomizing engine to decide the effect on weather over the streets as can be seen from Table 7. The traffic conditions data was achieved from various sources but never met the expectations. Thus a randomizing engine was created to simulate traffic blockings on the roads.

Description	Data used		
Data Gathered	Temperature	Humidity	Pressure
	Pressure Change	Condition	Wind Direction
		Wind Speed	
Data Missing	Chance of Rain	Rain Level	

Table 7 - Data found in the Weather RSS feed

4.9.2 Weather Conditions

Weather conditions form an important part of this study. The key role of weather conditions was to establish viability / access to a particular road e.g. during floods. This thought was inspired by the situation in Malta, where floods cause a number of roads to be inaccessible (MaltaStorms.com, 2008). The system had to be designed in such a way as to ensure vehicles would avoid these road segments in such adverse weather conditions.

4.10 Defining the Users

A top down approach was used to identify the user's needs. This was done using the UML approach, in a form of a Class Diagram, and showing inheritance of properties. Any user in the system must be classified and a specialised (web) interface for the user type was prepared. Access to the database, was identified through two means: Mobile WAP and Desktop Web. The users are therefore classified as:

- Desktop Users – Accessing the system using a Desktop browser, thus from the office or home.
    - Main User – This user type is the default user in the system. Basically this type of user needs access to add and manage vehicles on which to perform routing. The user needs also access to enable the default vehicle to use the routing exercise. Another access needed would be to perform a preparatory route plan, so as to print it for directions.
    - Emergency Response Team – This user type is the emergency team waiting for an incoming aid call. This user will have read access to the system, seeing how the route planner has proposed the route to the location in need of assistance.
    - Administrator – There are two type of Administrators, the Road Network and the Database administrator. For the scope of this discussion only the Road Network Administrator will be discussed, since the DB Admin needs no rights from the system, but is allowed free access to the data.
        - Road Network Administrator – The Road Admin needs access to alter the current road network. He can over write traffic blocks / jams, whilst also modify the banning of a segment from the road network.

Another access needed by the Road Admin is the ability to disable a road segment.

- Mobile Users – This type of users are the main users which will benefit from of Location Based Services. These users will be moving around on the road network thus acquiring location is important.
  - Main User – This environment allows the user only to plan a route and follow the instructions given. In this environment the user has no rights to alter details on the vehicle.
  - Welfare Staff – Police officers will have the ability to report an accident within 50 meters of their current location, which will require assistance from an ambulance or a fire engine.

The details described above can be seen in Figure 13, where a Class Diagram is presented (based on the UML methodology).

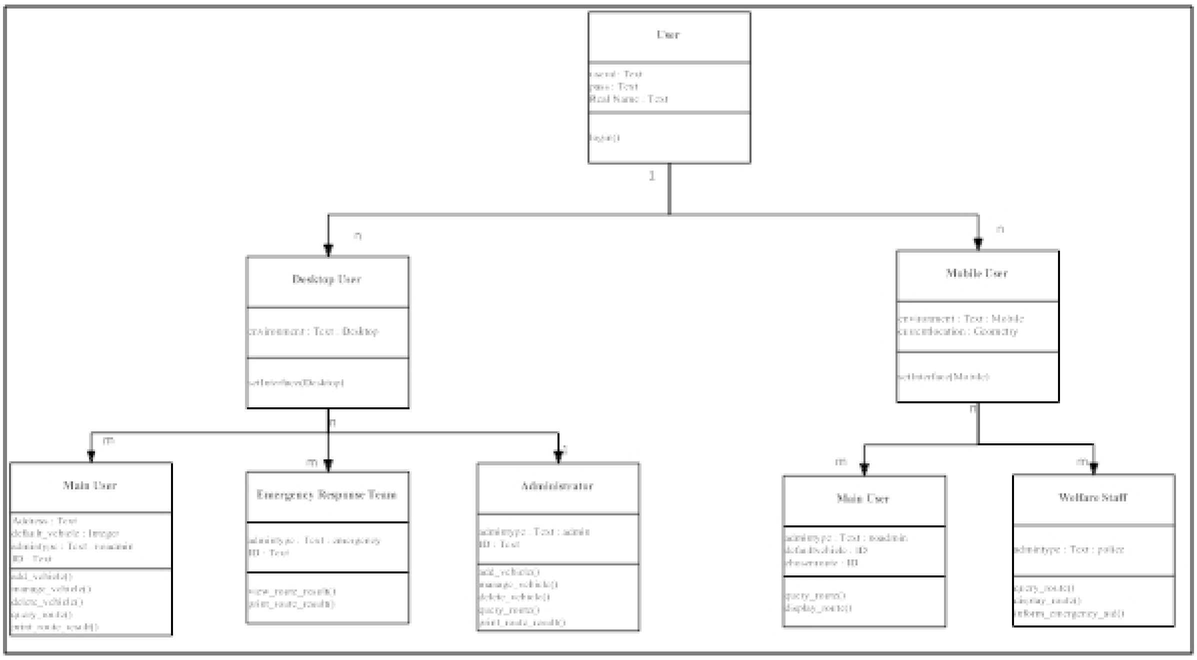


Figure 13 : Users Class Diagram

4.10.1 Routing Options

Different options are provided in preparation and identification of a route. For the scope of this project three options were developed. It was decided that a mixture of well known routing algorithms and an innovative approach were to be taken. These options are discussed below:

The **Shortest Path technique** is an algorithm, which takes care of turning restrictions, and making use of one way restrictions. This technique will make use only of available roads, thus any road which is disabled for the time being, will not be considered for routing.

The **Shortest Time algorithm**, is used when the driver needs to arrive at the destination point, in the shortest time possible, therefore the current speed limit is taken into account. Similar to the Shortest Path, this technique make sure that any road blocked will not be included in the route plan.

The **Fuel Friendly algorithm** takes into account fuel use and is the innovative aspect of this project. This technique makes use of pre-collected data about the road gradient by having segment information. This technique is computed on the assumption that bigger engines can cope with steeper slopes, at the same time it attempts to route smaller vehicles to a pre-defined engine to slope ratio on the street. Similar to the previous two algorithms, only segments which are accessible are given by the route planner. This algorithm takes into account also weather conditions, by avoiding any roads which are flooded with rain. This information is collated by the system (see Section 5.8).

Emergency Response Routing

The routing in this scenario is a bit different. Each road segment needs to have two types of blockings, one for normal traffic and another one if it is totally inaccessible for any vehicle. For example, if there is a traffic accident, normal vehicles will start queuing to pass, however an emergency response vehicle with beacon lights on, will be allowed to reach the scene. Thus, a partition between the types of blocking was needed. Routing for emergency response is based on the Shortest Path algorithm.

4.10.2 Identifying the Nearest Aid Station

The road network is populated with 71 fire stations and 13 hospitals. Routing in case of an accident starts from the nearest aid station on the network itself. Despite knowing that a simple distance query would give much faster results, it was felt that linear / spherical distance on the map differs from distance on the road network.

In Figure 14 the straight line distance is not the same from nearest hospital (1) to accident location (green dot). Therefore location (2) is much closer. This methodology is inspired after the work of Thirumalaivasan & Guruswamy (2005) where they used buffer zones to minimise the search area for their nearest ambulance and hospital, since in their study they were searching for both, rather than one as being done in this study. In their conclusion, they mention that this methodology should work better if the road network is a live one, which fits within this project's scope. A similar methodology was described in the paper by Schuurman et al (2006) where they specify that the most common approach to solve such a problem is to use Voronoi polygons, which are points representing the desired locations. A Euclidean distance forms a polygon around the desired point, in this case the accident point, so as to encapsulate an emergency aid station. A Voronoi polygon will be formed once there is an accident and the nearest aid station will be chosen depending on the pre-computed buffers. Schuurman et al (2006) mention also that this method has a shortcoming in that it does not account for changing elevation or differing road conditions.

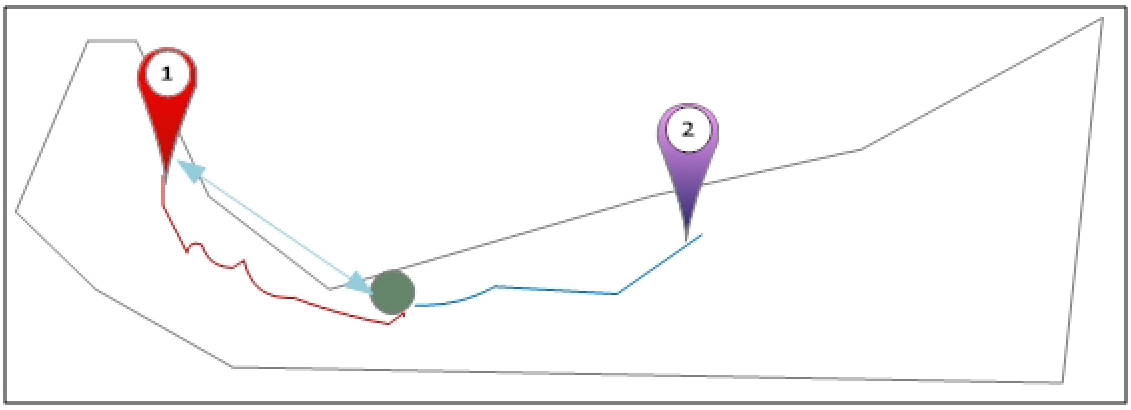


Figure 14 : Shortest Distance on Map not always the shortest Distance

4.11 Obtaining Standardisation

Another concern of this project is standardisation. In an ever increasing world of technologies, it is important to make sure that the project is compatible with most software, however, strict confirmations have to be met. It was decided that all software need to be as a minimum compatible with the Gecko Engine (which is the Open Source web browser engine used by Mozilla Firefox). Thus all the testing need to be performed on a browser using such an engine (Mozilla Firefox 3 was used). As outlined earlier, this does not mean that no care should be given to other engines. Another specification which the software should adhere to is the W3C standards. Every page delivered by the system should be compliant to HTML 4.0 Strict. This would allow the software to be easily portable on other web browsers, and also make it easier to be deployed on mobile devices. In combination with the previous compliance, it is a must that the Style sheet will be verified also against the W3C Standards.

4.12 Summary

In identifying the methodology and systems requirements for this project, this chapter provided the basis for evaluation and testing. Chapter 5 will present the results of the system.

Chapter 5 : Development

5 Development

5.1 Introduction

As outlined earlier on in the document, this project is intended to be developed using Open Source software. This study tries to show that Open Source includes also further help by the peers working in the same environment. Also, another advantage of working in Open Source, is that the work which will be proposed in this project, after submitted and accepted, will be presented back to the community, for further development and research. This idea of injecting knowledge gathered from an Open Source project, helps the community getting richer in information, rather than all the organisations moving on their own scope.

5.2 Choice of Road Network

After an extensive research for the desired data, covering altitude and turn restrictions, attention was brought to the dataset provided with the GRASS data book (Neteler & Mitasova, 2007). This suggestion was given to me by one of the authors of the book itself on the mailing list of GRASS. After reviewing the data, the GIS Analyst of the data collection team, (Hunt, Streets Data Layer, 2008)was contacted so as to clarify certain details in the Meta Data.

It was noted that the Data Set has been modified from Feet to Meters, thus creating some minor differences in the length of the roads. This problem occurred when the Wake County developers migrated from ArcGIS 7.x to 8.x (Hunt, 2008). Another result on the data change was that the data was transformed from WGS84 to NAD83, thus having a small (almost unnoticeable) degree of error in lengths.

The altitude in the Raster files was not useful for applying it to street segments, thus a longer approach was needed. Figure 15 shows the steps taken for manually creating the altitude of the streets. As one can see from the UML Activity Diagram, the commands used are from the GRASS GIS. The Activity Diagram shows the flow of data from having two separate files, one Raster file and one Vector file with distinct data. Both files were in the same projection so there was no need to modify it. Once the Sea level was computed, set to 0 after the consultation with Hunt (2008), the files were connected to output a third Vector File. This Shape File, was checked for altitude changes, and after successfully ensuring data has been exported into a 3 Dimensional plane, the Shape File was exported to PostGIS using OGR tools. Once the data was stored in the Database, another test on altitude using SQL was done so as to ensure data integrity.

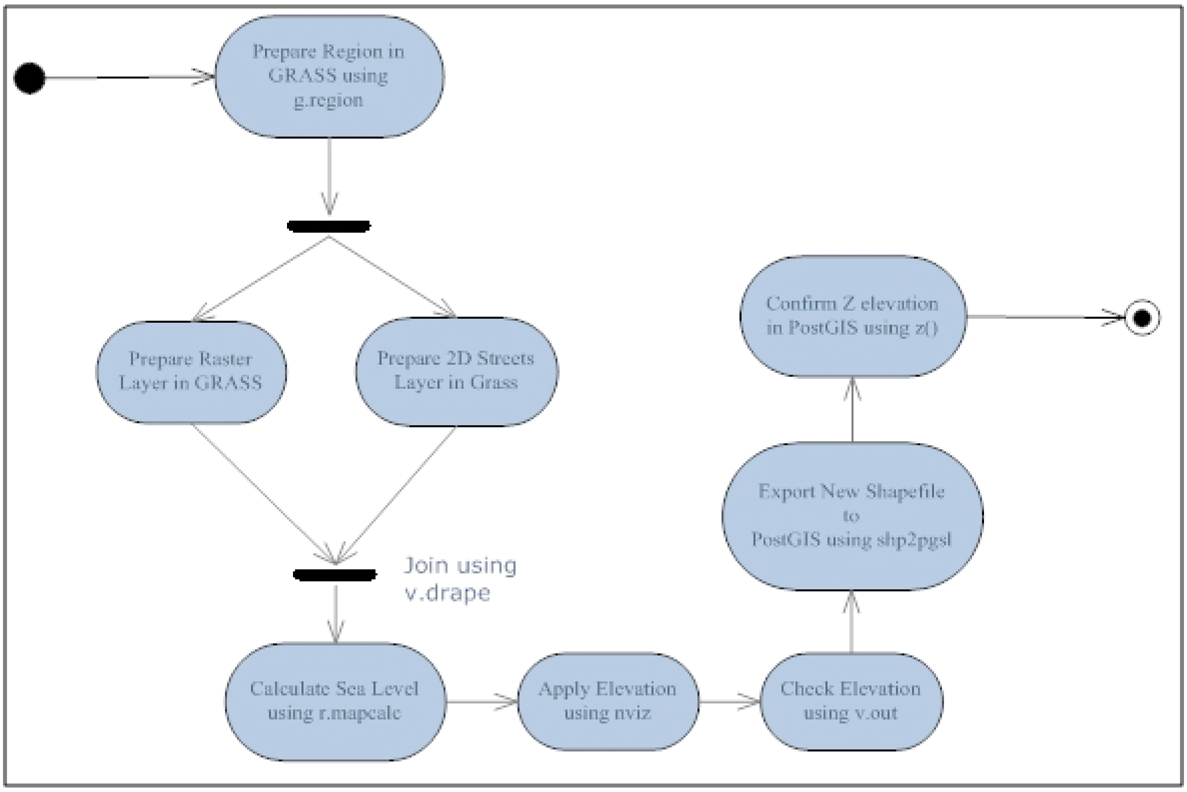


Figure 15 : Activity Diagram for Creating Altitude

Once the data was collected, and stored in the database, certain tests were applied so as to make sure that the network was able to handle turn restrictions and also to make sure that every road segment had its own altitude level. During the cleaning process, it was noted that some of the data layers included data of all North Carolina, thus such data was eliminated. Such elimination was done in order that no time is wasted in searching records which are outside the boundary of study.

The dataset chosen, covering Wake County is made up of sixteen (16) types of road classes, three (3) State road segments, about two thousand (2000) highway segments, another circa twenty five thousand (25000) primary road segments and another almost ten thousand

(10000) secondary road segments. As one can see from Figure 16, the statistics presented in the data[22] is quite balanced for a research of this type.

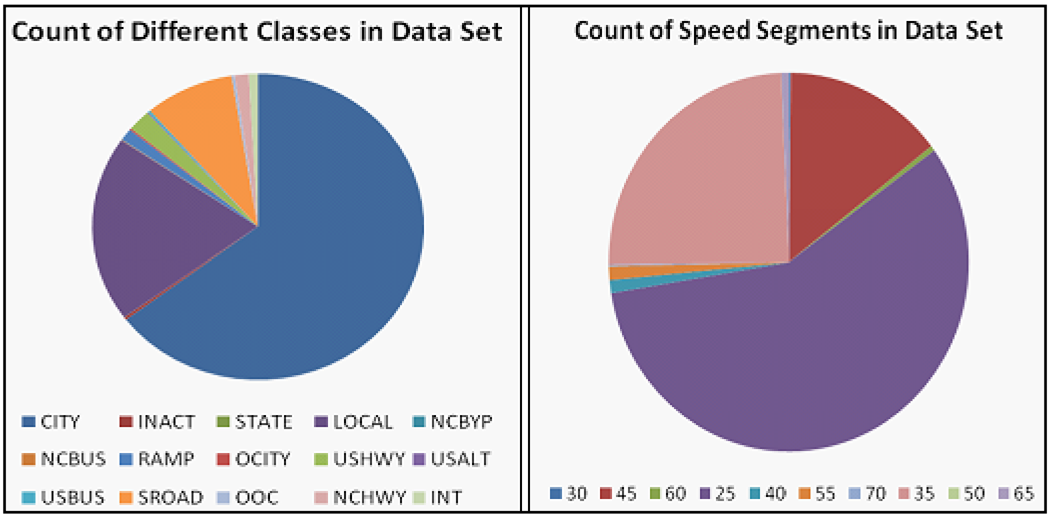


Figure 16 : Data Set Details

5.3 Preparing the Data

In this section, a closer look at the preparation of data, both spatial and non-spatial will be taken. The spatial data needed to be projected to one plane, and matched with the same projection of the Database. PostgreSQL with its spatial extension PostGIS is the most extended Open Source RDBMS with regard to spatial functions as can be reviewed in Table 2.

Once the data was stored in the database, testing began so as to make sure that the data stored in the database actually compared to real data. Several tests were conducted, which will be discussed further down in this section:

- The most basic test consisted on overlaying the data stored in the PostGIS database with that stored in Shape Files. As already mentioned the Shape Files were already modified by the GRASS community thus it was taken to be safe to work with. However, so to make sure that the data stored is fine, the original data was downloaded from the Wake County GIS department, re-projected to EPSG:3119

[22] Classification of Road Segments: City Street (CITY), Inactive (INACT), Interstate (INT), Local Neighbourhood (LOCAL), North Carolina (NC) Bye-Pass (NCBYP), NC Bus (NCBUS), NC Highway (NCHWY), Occasional City (OCITY), OOC, On-Ramp (RAMP), State Road (SROAD), State Highway (STATE), US Alternate Route (USALT), US Business (USBUS), US Highway (USHWY) (Mitasova, 2008)

using NAD83 datum, to be overlaid with the current two layers owned. To do such an operation, the Open Source GIS GUI tool: Quantum GIS (QGIS) was used. Once the three layers matched projection, and street detail it was accepted to move to the next test.

- Another test consisted in comparing the output of an area from PostGIS with the same area as presented by Google Maps[23]. The free to use mapping service provided by Google, allows for projection of personal data in their service. Thus, using GML the data was exported to Google Maps so as to make sure it compared well.

- The final test reflected routing. The Shortest Path algorithm was used to create three routes, one on the data stored in PostGIS, one on data stored in the Shape Files and the last on Google Maps. Seeing that the results matched, the data was agreed that it ranked sufficiently well.

5.4 Normalising the Data

Another consideration with regard to the choice and manipulation of data, it was thought necessary that data will be normalised so as to have data consistency in RDBMS and also more flexibility in the design (Stephens & Plew, 2002). A normalisation exercise was prepared and the output can be viewed in the Entity Relationship Diagram presented in Figure 18. Views are also included in the Entity Relationship Diagram due to connecting different tables. Since the streets data is static, it was thought that the best way to alter it was to create views and modify the table through a view, so as to limit the risk of breaking the constraints. Another advantage to use views is that the functions created to route the path, need just one table to perform the algorithm as can be seen in Figure 17. Thus, passing a view, where the proper tables would be joined is more appropriate. This method will not result in visible longer processing times, due to the fact that the joining would still be necessary. All the tables in the scheme are indexed, and constraints are set up so as to safeguard the data. As explained earlier in the document, access to modify the data is

[23] Google Maps : http://maps.google.com

strictly reserved to the Database Administrator, and any other access has to pass through the PHP scripts, where validation and verification techniques are set up to protect the data.

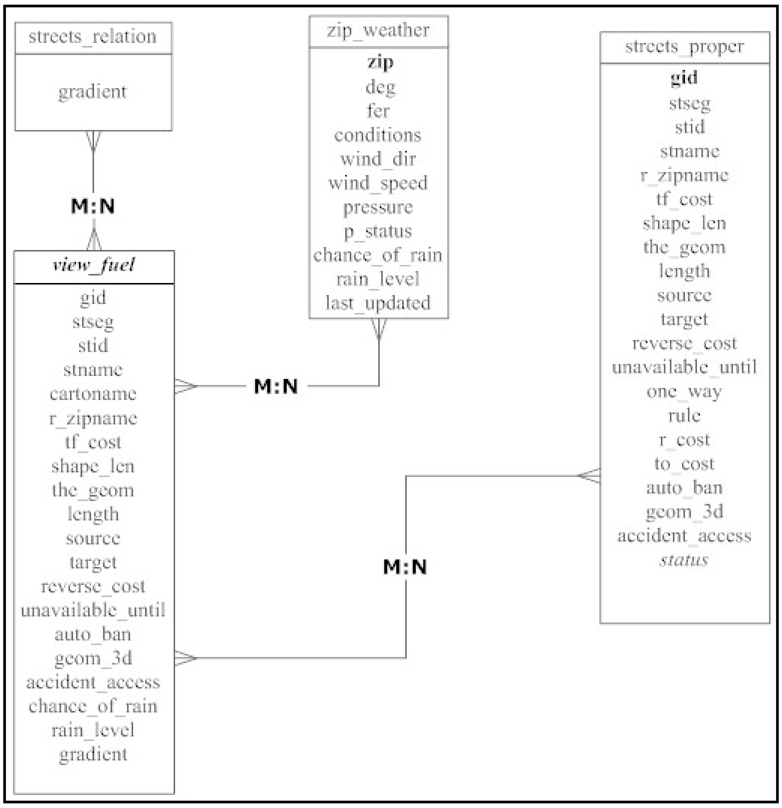


Figure 17 : ERD showing the Fuel VIEW

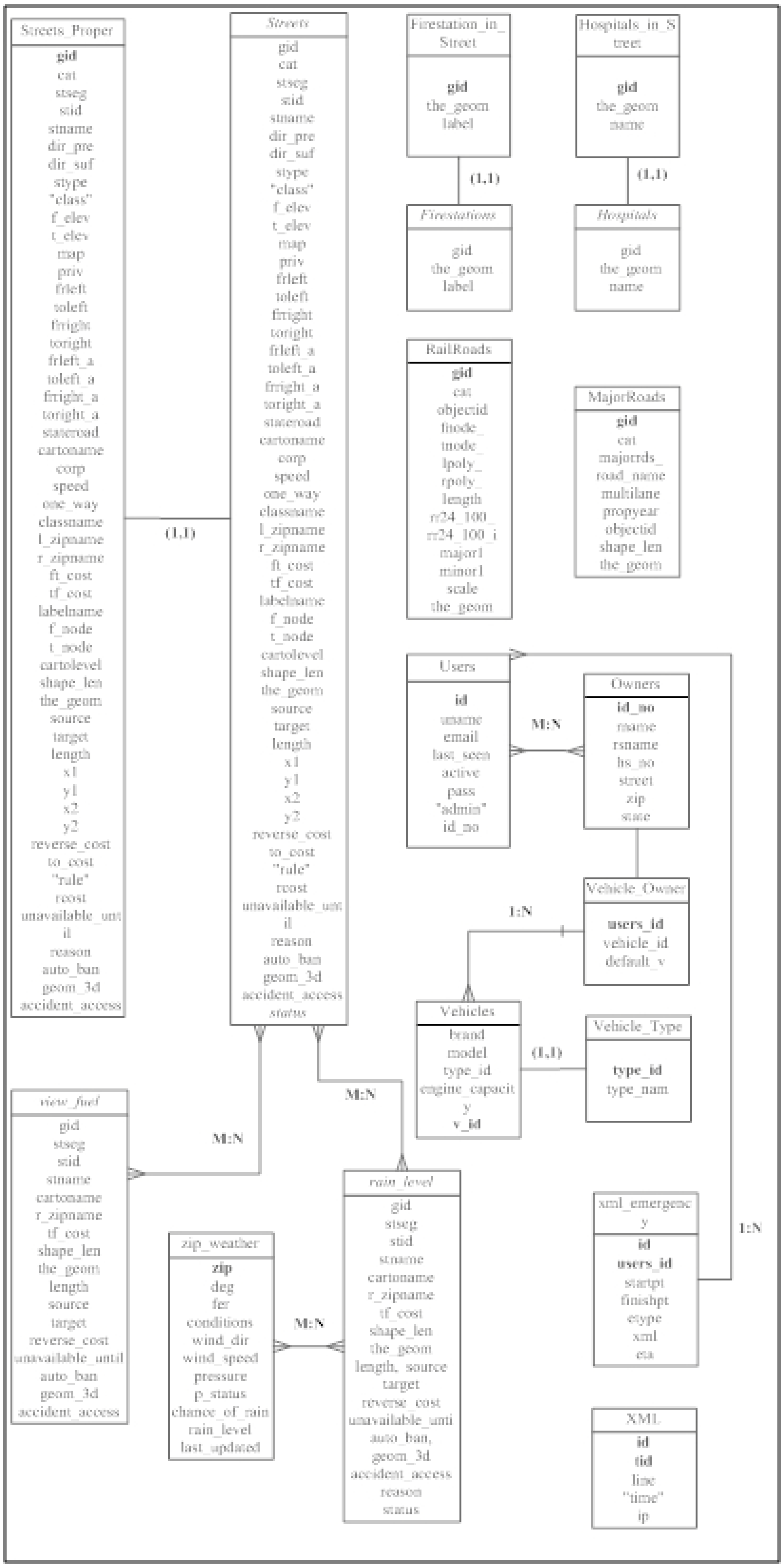


Figure 18 : Entity Relationship Diagram

5.5 Familiarising with the tools

Before embarking on this project, little to none was known on the area of Geographic Information Systems. It is true that some notion of Location Based Services was acquired however, GIS was a completely new sector.

During the first months of the project, most of the time was spent testing the tools which were foreseen to be used. One must say that the testing of the tools was done by providing custom scenarios and trying to figure out the way. Unfortunately, some tools come up with little to none documentation, and thus the only way to figure things out was learning by doing and asking on the mailing lists and Internet Relay Chat (IRC) channels. In opposition to the lack of documentation, one must mention that the constant support when using tools from the community was a push factor in the learning phase.

Section 3.10 outlined tools which are available within the Open Source GIS scene. Following such a review, Table 8 shows how each of this software will be used within this project.

OS Software	**Reason why software was chosen**	**Reasons where software was used**
PostgreSQL	• Budget: Since all software related to PostgreSQL is free of charge to download, it is much easier to use such software. ArcGIS could only be obtained for a trial period of 30 days; • Personal Inclination: I am a person who is inclined to use Open Source projects rather then Closed Source; • Procedural Languages: Making use of stored procedures in different languages can result in more productivity.	• Storing of all spatial and non spatial data; • Main RDBMS in the system, thus all queries will be executed through this DBMS
GRASS GIS	• Since the project is making use of only Open Source, GRASS GIS is the only OS software which is able to support for vector network analysis.	• Enabling elevation querying on the shape files; • Transformed data from Raster file to a 3D shape file which could be exported
pgRouting	• Written as a module to PostGIS • "*Postgres-based routing engine*" (Schmidt C. , 2007); • Extendible routing algorithms.	• Algorithms used and enhanced so as to allow the new algorithms.
QuantumGIS	• Can act as a frontend for GRASS GIS; • Can display all OGC formats;	• Display and test results of routing and other spatial queries;

	• Popular OS GIS Browser.	• Visualise the network data.
MapServer	• Can be enhanced for more functions using MapScript; • Supports presentation of OGR, OGC and GDAL.	• Presents the map to the user for routing service in a Web Interface.
OpenLayers	• Is a user side script thus no cross platforms restrictions; • Accepts API's thus can extend the presentations of MapServer.	• Used as the front end map generator for the clients, and loading the base maps from MapServer; • Also making use of AJAX function calls, the route will be displayed on-the-fly.
TileCache	• Maps generated will be split into tiles and cached on the server's hard disk, thus reducing the computation time by a high fraction. • OpenLayers can then be used to call the tiles and serve them to the client at a much faster response time.	• Enhancing the loading times of layers on the Web Interface.

Table 8 - The use of OS in this project

5.6 Setting the Environment

Since the project is set to use only Open Source software, the Operating System had to be Open Source too. Open Source software is described to be "*sweeping the enterprise IT world*" (Ramsey, 2007). Another important factor which influenced the choice of an Open Source Operating System is summarised in *Table 3* published by Steignier & Bocher (2008). All the software which is intended to use can be operated on Linux. Thus, it was decided to install Ubuntu as my host for the project. The reason why Ubuntu was installed is inspired from von Hagen (2007).

The Ubuntu server will act as the main host for all the services needed by the system. Figure 10 shows the Web Server (Ubuntu Server). This server will be offering the different services (Appendix I: Table 13). The services can be classified into three types summarised below:

- Spatial Services: Holding spatial data, both stored in the PostGIS and also stored as Shape files, called from MapServer;
- Web Services: Such services will act as the gateway to the backend data, and will be available to the Internet;

- Support Services: These services are not particulary associated with this project, however they are highly useful for the running and the safeguarding of the data. Such services would include CRON[24], iptables[25] and logrotate[26].

Once the Operating System was set up, some tests were done so as to make sure it is stable, including trying to DoS (Denial of Service) attack, and showed it is stable. Once testing was concluded, the services installation took place. The tools were installed making sure compatibility between them was kept. When the installations finished, some random data was stored in PostGIS and some tests were performed to outline the installation. Seeing the set up was fit for data, the next process of setting up the data started.

5.6.1 Protecting the Data

The Server is behind a router, protected by a Demilitarized zone (DMZ). Unfortunately there was a lack for a hardware firewall, thus the Web Server had to rely on a Software one. A DMZ is a physical sub network protected by hardware, thus the Internet (unsecured network) reaches only the Router of the network, and is allowed to serve data only if it is originating from a trusted zone. This technique is an extra layer of protection for the data stored at the server. Another protection performed on the Database Side, is that only trusted hosts, which are password protected can reach the data, and another rule to allow direct data management from local network only was set up. As one can refer back to Figure 10, one can see a physical link from the terminal used by the Database Administrator and the Web Server (as mentioned already denoting all the services).

5.7 Road Conditions

A search for an RSS feed from a state department or some public live feed resulted in only a possible good feed[27], which in turn was not usable for data extraction, due to the randomly generated description of the feed. For the scope of the project, it was decided that a Random generating function for disabling roads would be needed, which creates XML files.

[24] CRON : A time based scheduled application that runs on UNIX like systems at a specified interval
[25] iptables : A Linux packet filtering tool providing firewall security
[26] logrotate : A tool used to manage large number of log files, by allowing rotation and automatic removal amongst other operations
[27] MyWeather.net : http://wral.greenlight.myweather.net/greenlight/main.html

A Sample RSS reader was scripted to extract the random generated data. The Generated data will have random type of events with a corresponding randomised time factor. The way data is randomised can be seen below:

Blockage Type	Minimum Blocking	Maximum Blocking
Accident / Road Lights Malfunction	2 Hours	1 Day
Road Works	1 Week	3 Months
Road Maintenance / Road Construction	6 Months	2 Years

Table 9 - Types of Random Road Blocking

The algorithm to generate the random scenarios will be used within a CRON job, thus every 30 minutes a script will be fired up and calls the create accident function after a random number of minutes. All this randomisation will present as close as possible the way blockings on a real road network happen. This technique was inspired from the article by Karl, Charles, & Trayford (1999). Also, whilst preparing the randomisation function, it was felt necessary that more randomised events on a lower scare should occur then on a higher one. Thus, road accidents should be more frequent than road maintenance or even more than road closures. The randomising script should be constantly running and modifying the data set, thus making the result of routing interesting since it adapts to the varying data.

5.8 Weather Conditions

Acquiring weather data from a real live feed was easier than traffic conditions. The feed acquired was an RSS feed, which provided a unique feed for every county within the state. Data could be searched via the zip code. The feed chosen was from the Web Site WeatherUnderground.com[28]. Once the feed was acquired, randomising factors need to be applied so as to generate a prediction of weather. Following tips from Gera (2008) (Gera, 2008), it was decided that the following methodology for rain prediction is taken. As one can see in Figure 20, the algorithm relies on randomisation and invented boundaries. The randomisation can also give rise to some extreme scenarios. As already mentioned earlier, the boundaries, and how each variable affect the whole rain prediction was based on a meteorologist opinion (Gera, 2008) (Gera, 2008), thus trying to make the synthetic data as near reality as possible.

[28] Weather Underground: http://rss.wunderground.com/auto/rss_full/NC/

Similar to the Road Blocking script, the Weather Conditions will be ran using a CRON job. Since Weather conditions do not change as much as Road conditions the timings will differ. The Weather RSS feeder is set to run at six hours interval, thus four times a day. This means that once the RSS feed will be collected, the Weather condition generator comes into effect, thus following the steps presented in Figure 20 below. Finishing the script would be an SQL Update query so as to update the database with the newest weather conditions. Once the data is updated in the *"zip_weather"* table, routing can start being affected.

The UML Sequence diagram (Figure 19) shows a loop between the entities. As one can realise from the notion of a Sequence Diagram, the processes within the loop will be looped until all the ZIP codes will be traversed, and the weather conditions for each is recorded in the database. Once this process is finished, the loop exists and the PHP script returns the exit code for the CRON daemon. The activity bar for the CRON entity remained all throughout the process due to the fact that the CRON server opens the PHP page as a child to its processes, thus remains running in the background.

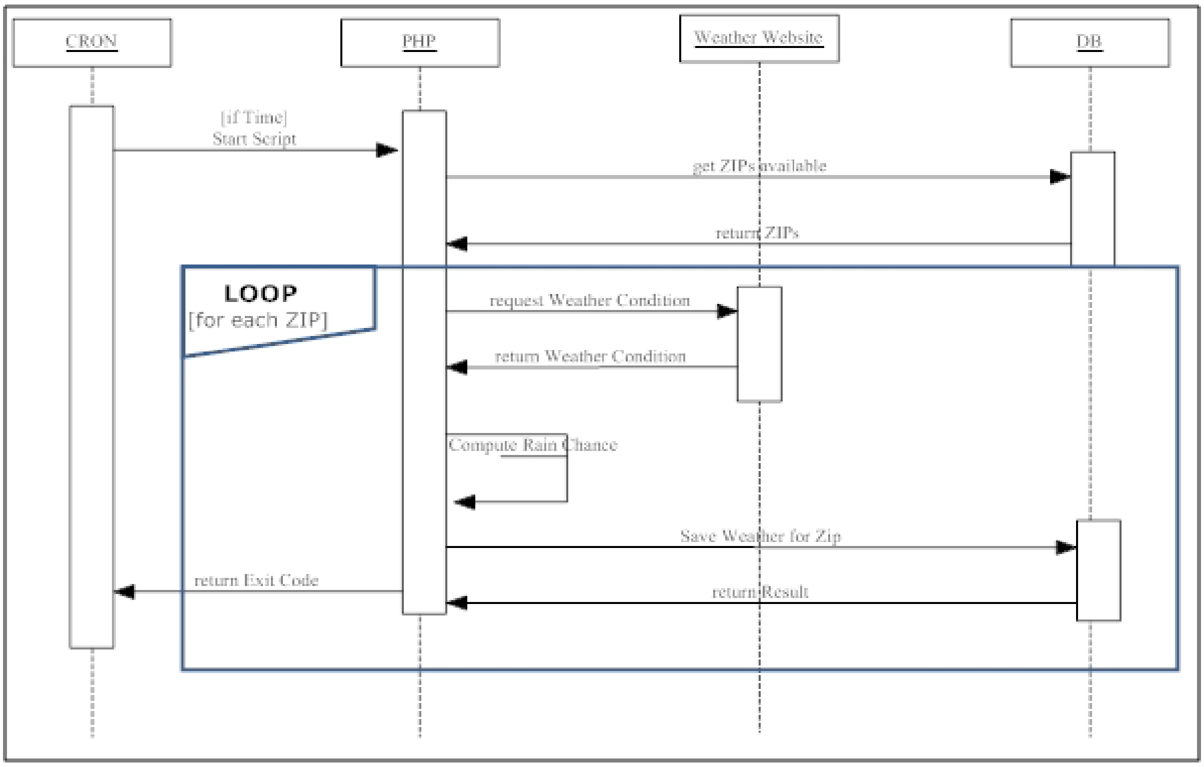


Figure 19 : Sequence Diagram for Weather automatic updating

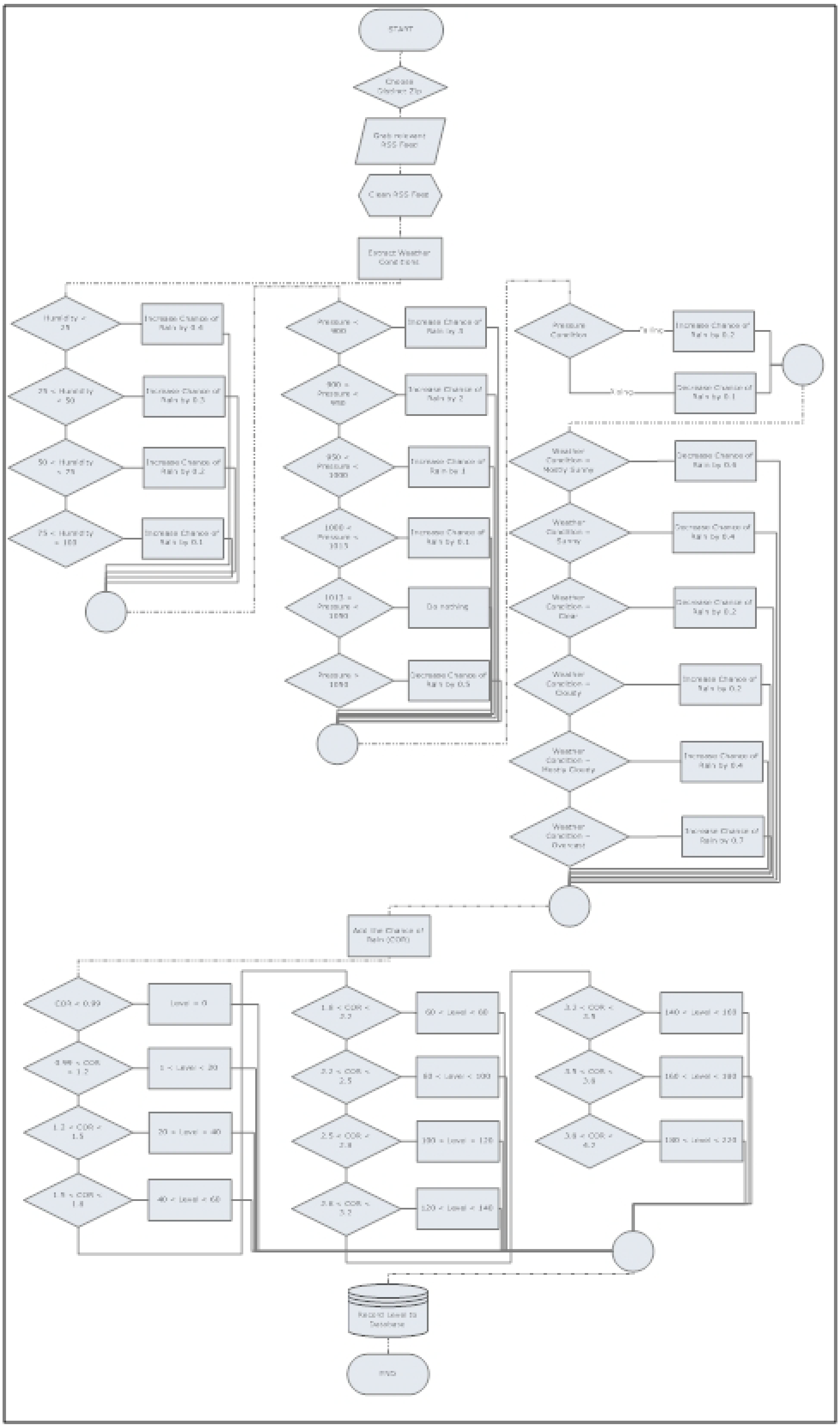


Figure 20 : Weather Prediction Flowchart

5.9 Managing the Users

The users should be regarded as the main players in the system, mainly because the Location Based Service is targeted to the user. Recalling what was written earlier on in the document, several entities benefit from an LBS practice, however in every system the most type of entity that benefits from this practice are the users. This benefit is due to the fact that the users will be roaming the network and asking for information (routing information in this project). In this section, a more insight on the users, and how each affect the data shared within the network is presented. Instead of presenting what each user must have, this section will tackle how the users actually affect the data itself.

5.9.1 Modifying the data

The normal user, known as the Vehicle Owner, has the opportunity to add and remove vehicles from the network. It is expected that any owner will be driving one vehicle on the network, thus, having one default vehicle. This default vehicle will be the vehicle used when the routing is done on the least fuel consumption exercise. This routing takes into affect the type of the vehicle, mainly its engine capacity and compares it with the altitude of the road to traverse. Thus the default vehicle, managed by the driver is extremely important in the routing choice. Needless to say it is important that the data is validated once it is entered, due to the relevance to routing.

Another user which modifies the data would be the Police Officer (denoted by “police” admin status). Any police officer, whilst on duty, can be presented with a 50 meter radius where he can report an accident. The Police Officer can request help from an Aid Station whilst reporting the incident. This exercise is deemed to modify the data, because once the report has been inputted, the system automatically bans the road segment from further routing until two hours. This accident reporting continues to add to the dynamics of the data. The road segment where there was an accident will be blocked for normal traffic but still accessible to emergency response teams. For a road segment to be blocked for emergency response teams, it has to be banned by the Road Network Administrator only, which will be discussed on the following paragraph.

The most predominant user to modify data would be the Road Network Administrator. The Road Administrator will have the opportunity to view auto banned road segments and re enable them. The Road Administrator will also be able to extend or curtail any ban existing in the database. By ban, it is meant, making the segment not available for the routing of normal vehicles and in extreme cases for emergency response teams. Despite allowing such changes, the Road Administrator will not have any access to the modification of data in the database, thus for example cannot manage any data owned by the end users.

The user who has the most responsibility to manage data is the Database Administrator. The DB Admin must realise that despite having such an access granted, minimal use should be made. Since for most scenarios, a web front has been prepared, it is suggested that this front would be used, rather than directly modifying the data in the database. Also triggers have been set up so as to ensure integrity is kept, thus it is imperative that data is edited via these scripts.

5.9.2 Inputting the Data

Whilst in its early stages, the system was developed with close collaboration between the development area and the database management one. This means that data was modified by the database administrator and prepared for use. Most of the spatial data arrived as Shape Files, thus it was necessary to export the data to PostGIS. The Database Administrator role, made sure that the data remained constant to its source, and no discrepancies were found. Another task conducted by this role was to prepare the schema as explained earlier on, and set up the constraints and views so as to protect the data. Data integrity is a major task for the DB Administrator role (Cockcroft, 1997), thus extreme care has been taken to safeguard it. Following the techniques mentioned in Borges, Laender, & Davis (1999), spatial constraints were set up also to make sure that once the database was being populated with data, data integrity would still exist.

The vehicle owners need to input data to the system too. The technique used to validate the entry scripts was taken from Merrall (2005). The vehicle owners need to specify and populate the database with the new data about their vehicle. Having a properly working database populated with real cars was out of the scope of this project, thus verification was not possible on the data provided on the vehicles. In this case verification is relied on the vehicle owner to judge that the data is really the true data.

5.9.3 Categorising the Users

In an effort to visually explain what was documented in Sub-Sections 5.9.1 and 5.9.2 a Use-Case Diagram will be set up to explain all the key users how they interact with the system and what is expected from each of these users.

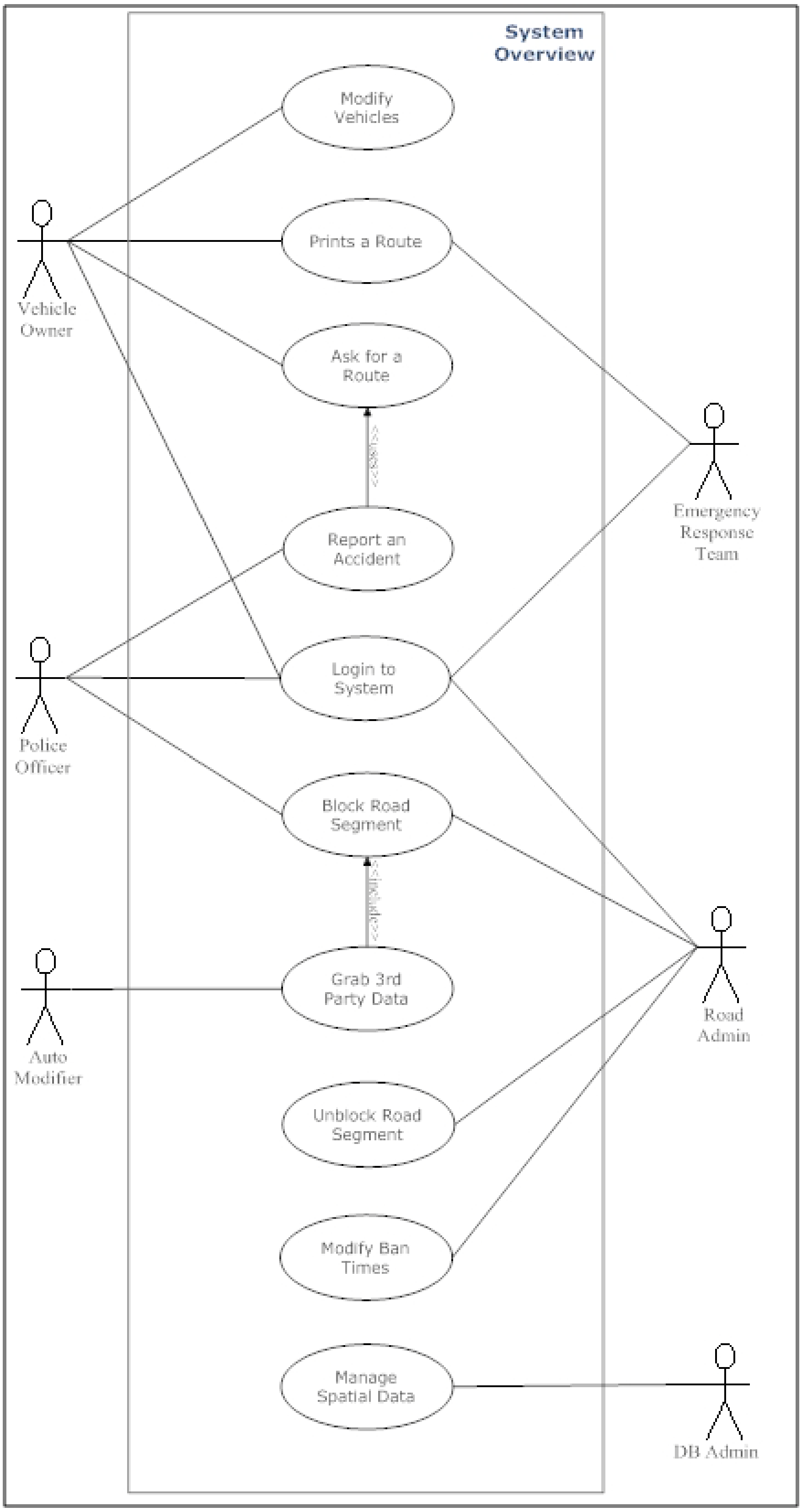


Figure 21 : Use Case Diagram

5.10 Modifying the Route

The technique used to enable or disable roads was created in two columns in the table, one showing if the road segment was disabled only for normal vehicles, and one for emergency access. So as to keep such dynamic data apart from the main spatial data, views were used. What is stored actually on the database is the expiry data of the ban, which when compared with the current timestamp of the query, if it is earlier, it means the road can be used (having a TRUE status) else, status will be set to FALSE making it impossible to be called for routing.

$$streets\_proper.unavailable\_until < 'now' :: text \\ :: timestamp\ without\ time\ zone\ AS\ status$$

Thus in the database only the expiry date was stored making it possible to be heavily indexed by using B-Tree Indexing method. A similar technique was used to display the reason, if the segment was banned:

$$CASE\ WHEN\ streets\_proper.unavailable\_until\ < 'now' :: text \\ :: timestamp\ without\ time\ zone\ THEN\ '' :: text \\ ELSE\ streets\_proper.reason \\ END\ AS\ reason$$

A rule in view was also prepared so as to update these two tables within the table only via the rule and cannot be surpassed. This would allow better handling of data, and if need be can be easily set to Trigger any function which might be deemed necessary.

The Emergency Access was set in a similar way, and stored too as Boolean. Storing data as Boolean can enable another Index running on the table. This would help lesser seeking times by a high degree. A simple query like

$$EXPLAIN\ ANALAYZE\ SELECT\ gid\ FROM\ streets\ WHERE\ NOT\ accident\_access$$

will take much less time than when it is not indexed. The result can be viewed in the following table:

WHERE Clause	Using Index	Not Using Index
WHERE accident_access	81447.448 ms	99421.336 ms
WHERE NOT accident_access	59.560 ms	78.705 ms
WHERE status	71465.964 ms	82011.864 ms
WHERE NOT status	80.231 ms	300.943 ms

Table 10 - Indexed vs Non-Indexed times

As one can see there is a difference in the outcome when it comes to indexing for banned roads. Also, due to disk caching, the WHERE NOT clause can even be reduced to less than 1 ms if the second query is ran shortly afterwards. Despite the long times when the query asks if the variable is TRUE, one must remember that the scripts will search for those which are banned not the normal accessible roads, hence why an Index is used. Indexing is more useful when it comes to extract small portions of data from the dataset, and the Planner, would prefer a Sequential Scan when it sees that most of the data will be returned, even if an Index exists (Fetter, 2007), (Wiles, 2008).

5.11 Computing the Route

After closely looking at what routing algorithms are available, the pgRouting project was pointed out. This project gives the ability for three types of Routing techniques, A*, Dijkstra and Shortest Path. The Shortest Path is the only technique that makes use of turn restrictions, so it was an obvious solution to enhance this.

The main **shortest path algorithm** was enhanced so that it takes into consideration the status of the road. This technique takes no other heuristics, except making sure that the route given makes use of usable roads. This function, extends what came with the pgRouting package, known as Shortest_Path_SP() PL/PGsql function. What was done in this extent is modifying the function to accept another parameter where the status row could be specified. Also the wrapping function made sure to capture also the source and target edges, something which required two extra SQL calls in the pgRouting function. Another enhancement provided on this function is the ability to allow the specification of the rule column to use when doing the routing.

The **shortest time algorithm** was written, based on the Shortest Path Wrapper created earlier on. What was modified in this function is that the road segments chosen were based on speed limit. Since the speed limit will be a dynamic factor in the database, it is thought that by moving on faster roads, the destination can be reached in lesser time (Coskun & Erden, 2006). Needless to say that a combination of Shortest Path and Shortest time is needed, thus the route is computed on giving the Shortest Path by traversing the highest speed possible road segments. Such traversing will sometimes find no option other than passing through slow-speed roads, thus a fall back mechanism was created, in order to give a slower route in case the route could not be computed. This fall back technique has proved to be a successful one in the numerous techniques performed whilst preparing the algorithm. What is being done is, if the function return an Empty Set, thus no route found, the Speed Limit will be lowered.

Driving safely must not be seen as the only way to lower the consumption of fuel (Ericsson, Larsson, & Brundell-Freij, 2006). The **least costly algorithm** was built in order to tackle fuel consumption. To help understand how it was built, a Flow Chart is prepared so one can be able to document each visual step. Figure 23, shows a very high definition of the algorithm that can well serve as a starting point for the developer. What one must point out from the algorithm is that where possible, extents start on a small figure, *var_max_dist* denotes the extent by which the point is searched, keeping it as small as possible. This is done so as to limit the function processing time (Pulis & Attard, 2008). Another mentioning would be that wherever possible, the function is using indexing, both spatial and non spatial. For example the "*&&*" notion make uses of the GiST index present on the geospatial table. Thus extending a point and making use of the GiST index the output would be in less than 10 milliseconds. Figure 22 outlines the steps needed for the algorithm, in the form of a UML Sequence diagram. As one can refer, the Sequence diagram starts from right after the user's login, and continues to request the route option, and on the presumption that the route chosen is Fuel Friendly follows such a sequence (Pulis & Attard, 2008). In the Sequence Diagram one can view the three tiers discussed earlier on in the Methodology Chapter, Section 4.5. This three-tier set up has been created following discussions found in Cheng et al (2004). The Sequence Diagram below can be used for any of the three routing choices.

What differs is the stored function within the database, but the process is all the way the same.

The Shortest Path function returns the closest edges, however, the custom function written for this project, extends this by forming an additional path between the vertex and the edge, so as to be able to guide the driver for the first and last sections of the journey. Such process can be further examined in Figure 24 where a more detailed pseudo-code of the Fuel friendly algorithm is explained. The pseudo-code deals with the whole process of how the route is acquired executed and displayed back to the user.

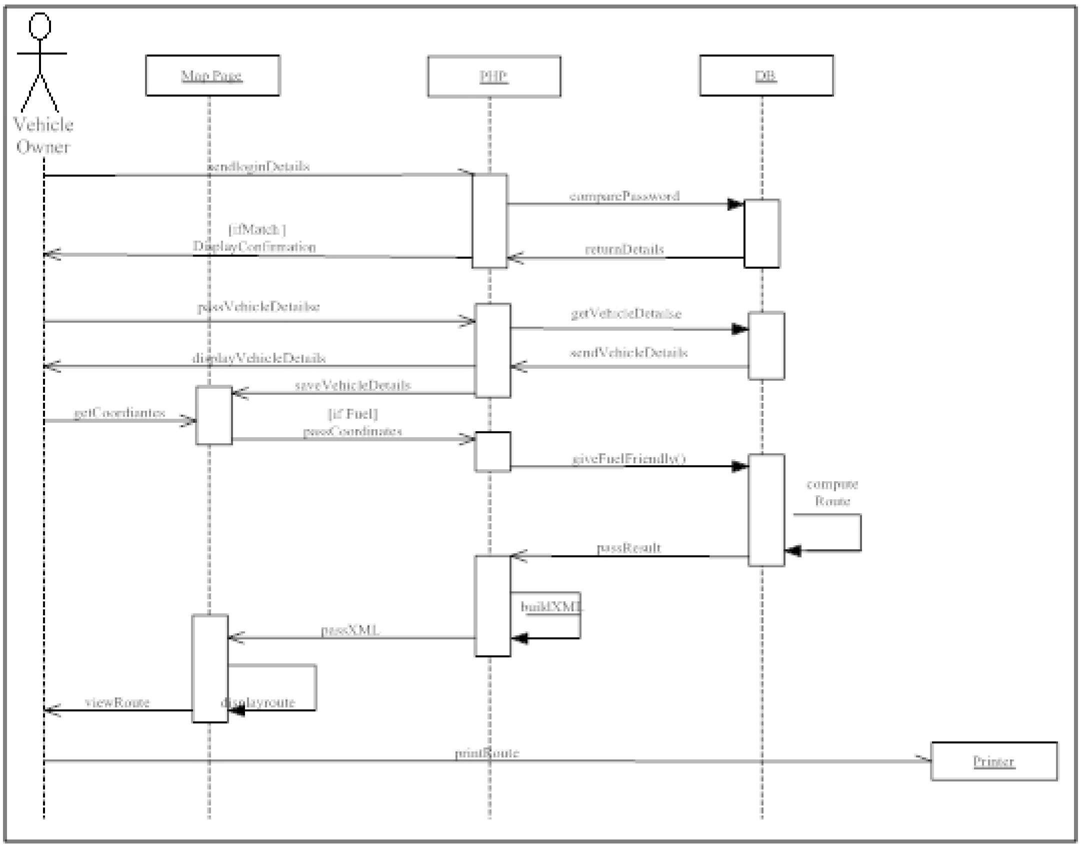


Figure 22 : Sequence Diagram for Fuel Route

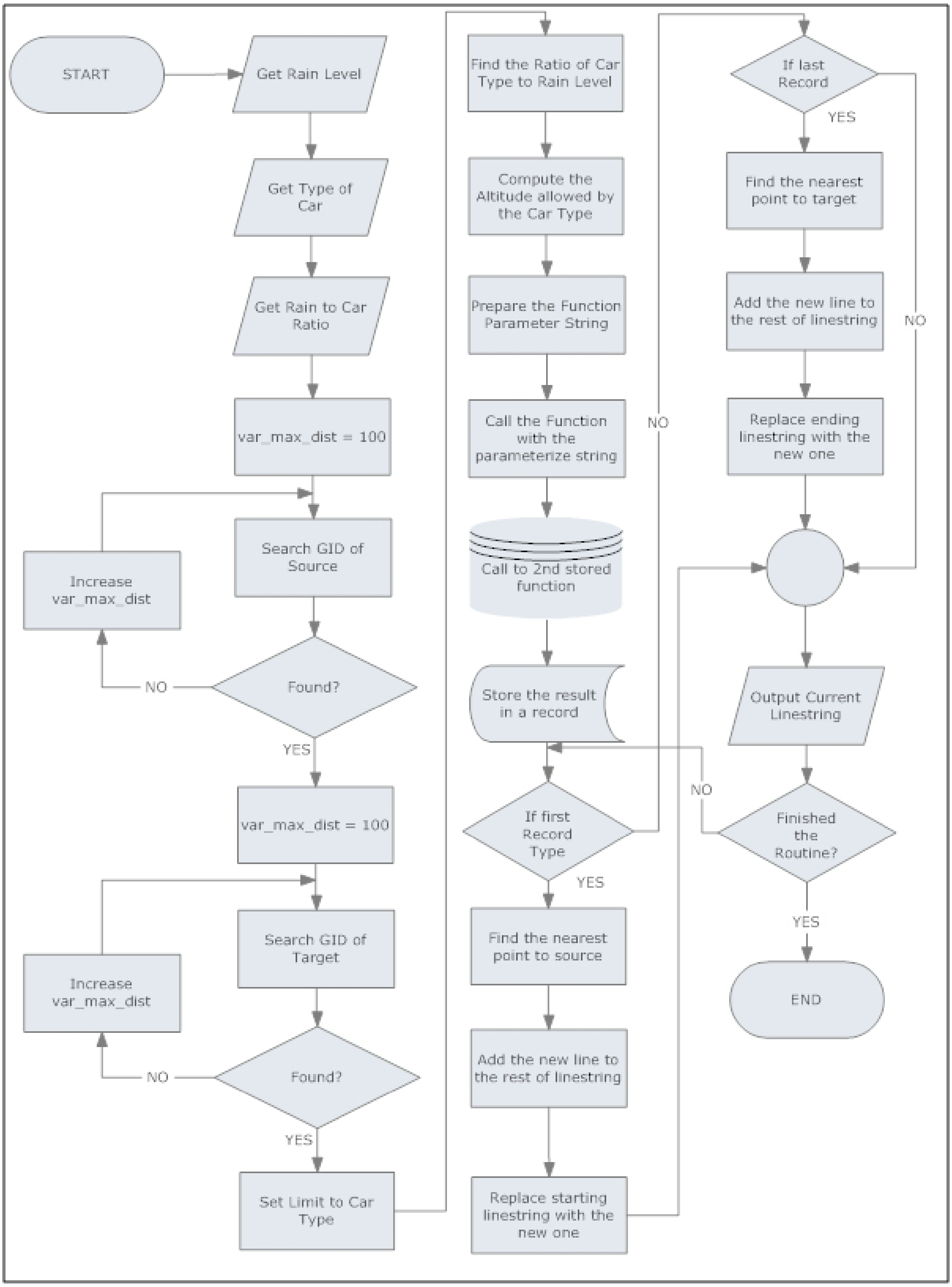


Figure 23 : Fuel Friendly Algorithm (Pulis & Attard, 2008)

1. *User successfuly logs in*
2. *Data from the default vehicle is loaded from the DB and saved in a COOKIE*
3. *User chooses the desired routing option (presuming Fuel Friendly)*
4. *Using XHTMLRequest, data regarding the default vehicle is presented to the user*
5. *If user wants to change the default vehicle the "Change Vehicle" is pressed else move to step 6*
    - 5.1 *Go back to step 2*
    - 5.2 *Load the Vehicle Management Interface*
    - 5.3 *Save and return*
6. *Using the web interface (based on OpenLayers) the user choose the Starting and Ending point*
7. *The starting and ending point, preferred routing method, default vehicle data and transaction ID are passed using XHTMLRequest to the routing PHP script*
8. *The script, depending on the route choice (presuming Fuel Friendly) gets an average height of the vehicle from the hardcoded parameters depending on the type*
9. *Sets as maximum rain level one-third (33%) of the vehicle height*
10. *The script calls the PG/Plsql function: give_most_fuel_friendly() passing the starting and ending points, maximum rain level, type of vehicle and then engine's vehicle*
11. *The PG/Plsql script starts off by finding the geographical location of both starting and ending points:*
    - 11.1 *Transform the point using PointFromText() function*
    - 11.2 *Expands the point by a 100 meter buffer to find the closest street segment*
    - 11.3 *Order by linear distance from point to street segment*
    - 11.4 *IF Found: Save the street segment ID and Go to 12*
    - 11.5 *If NOT Found: Add another 100 meters to the buffer extension and Go back to 11.2*
12. *Depending on the Vehicle Type set a limit on the Gradient*
13. *Set a limit that given road segments need to have rain level below the parameterised value*
14. *Form a text WHERE clause with results from step 12 and 13; The WHERE clause should be similar to WHERE rain level in road segment < threshold the vehicle can pass through AND the gradient < the maximum gradient the vehicle can pass through*
15. *Call the PostGIS modified function shootingstar_sp_where_new() passing the new starting and ending locations, the street cost, the column which direction is based upon, and the WHERE clause in 14*
16. *A query to find the shortest path is performed making sure that the road segments obey the constraints passed in the WHERE clause*
17. *Shootingstar_sp_where_new() return back a SET OF street segment Ids and their relative geometries*
18. *The set is sent back to the give_most_fuel_friendly() function and stored in a temporary record*
19. *The first record, is checked to see how far the original starting point is from the actual start of route*
20. *By making use of line_locate_point() function the nearest point to the actual starting point on the first linestring is found*
21. *A new linestring is formed using LineSubstring() PostGIS function*
22. *The new linestring is joined to the first linestring found in the first record (step 19) using PostGIS's GEOMUNION() function*
23. *The first row from the record is returned to the user, returning the street ID and the geometry location*

24. *The rest of the records are returned to the user as per step 23 except for the last one*
25. *An EXCEPTION is created for the result processing block (Steps 19-24) in case the route fails because no optimal routing could be found for the vehicle size which ends graciously the algorithm*
26. *For the last low Steps 19 – 22 are repeated*
27. *The give_most_fuel_friendly() returns the last record and ends*
28. *The PHP script saves the result into a record*
29. *A loop is created which cycles every record*
    - 29.1 *For every record, the maximum speed limit and also expected time to traverse the segment is found by another query*
    - 29.2 *Using PostGIS's Azimuth() function the next segment turn's angle is found*
    - 29.3 *Add to an XML string*
30. *When the loop (Step 29) is finished, the XML string is saved in a text file*
31. *The same XML string is echoed back to the User Interface (Step 6) via XHTMLRequest*
32. *The XML handler of Openlayers openXMLrequest() is called so as to interpret the XML result*
33. *The routing result is displayed to the user*
34. *A loop is started*
    - 34.1 *Each part of the itinerary is displayed to the user with the estimated driving time for the particular segment*
    - 34.2 *A counter of how many seconds the route is expected to take to be traversed is kept*
35. *Once the loop (Step 34) finishes, the temporary XML file is deleted*
36. *The seconds are transformed to Hours and Minutes and using XHTMLRequest displayed to the user*
37. *END of Fuel Algorithm*

Figure 24 : Pseudo-code of Fuel Friendly Algorithm

In Figure 24 (Step 14), the variable limits are being briefly mentioned. Such limiting factors are the variables mentioned earlier in the document, which affect the routing based on the online data. The WHERE clause is explained below:

- Weather levels

  var_text_rain_level := v rain_level <= v || max_rain_level ;

  Since the query is making use of a VIEW (Figure 17) all the data related to this algorithm is viewed in the same table. Thus *rain_level* refers to the actual amount of rain in the street as stored by the weather forecast algorithm (Figure 20). What is being done in this function is that the maximum rain level (*maximum_rain*_level) that the vehicle can pass through, is being compared to the actual rain level on the road. This results in all road segments given in the result to be safe for the vehicle to pass through (SmartMotorist, 2008).

- Gradient

```
IF type_of_car = 1 OR type_of_car = 2 THEN -- Small Vehicles + Lightweight Goods
    IF car_engine < 1200 THEN var_highest_road_factor := 15;
    ELSIF car_engine >= 1200 THEN var_highest_road_factor := 20;
    END IF;
ELSIF type_of_car = 3 THEN -- Big Goods
    IF car_engine < 4000 THEN var_highest_road_factor := 15;
    ELSIF car_engine >= 4000 AND car_engine < 10000 THEN var_highest_road_factor := 20;
    ELSIF car_engine >= 10000 THEN var_highest_road_factor := 30;
    END IF;
ELSIF type_of_car = 4 THEN
    -- Public Transport : Such Transport has to pass through everywhere so the highest road factor would be high.
    var_highest_road_factor := 20;
END IF;
```

It is very important to start this discussion by mentioning that these figures are all presumed figures and are used as a proof of concept only. There is neither engineering, mechanical nor physical expertise involved in providing such figures and limits. All of this data is formed on these two assumptions: A small engine (with a lower Break Horse Power (BHP)) finds it harder to climb a hill rather than a bigger engine. The weight of the vehicle makes an influence on the fuel consumption, thus a gentler route should be given. The *var_highest_road_factor* stores the maximum gradient the vehicle can pass through. Such a variable will then be compared to the gradient extracted from the Fuel VIEW (Figure 17) as can be seen here:

var_text_gradient := v gradient <= v || var_highest_road_factor ;

Such technique, similar to the one above, makes sure that the vehicle traverses only roads which do not have a high gradient when compared to the vehicle. This reflects in less fuel consumption (Ivo, 2007).

- Available road segment

status = TRUE;

As explained earlier (Section 5.10), the status reflects the availability of the road segment. This condition limits the network to provide only valid route segments, thus which can be traversed by the vehicle. Such constraint is available by using the Fuel VIEW (Figure 17), and is limited on the *unavailable_until* time set by the person (Road Admin / Police Officer) or script (traffic update script) setting the ban.

In Figure 25, the process by which a new call for **emergency** is raised is being reviewed. Similar to the Sequence diagram for Fuel (Figure 22), the Police officer needs to log in to the system and verify credentials met before moving on to the Report Accident Screen. This process, so as not to repeat, will not feature in the next Sequence Diagram since it is technically the same as above. Thus the process documented will start from the step where the Police Officer is logged in, making the assumption that the officer has the right credentials.

As one can refer below, the middle ware is initiating another process which will raise the alert to the emergency response team. An important sequence is that where the XML is temporary saved in the database. Testing showed that for manageability purposes, it is better to store the XML needed for the route in the database rather than on temporary files, in case two calls are dealt with at the same time, thus making use of the DBMS functionalities.

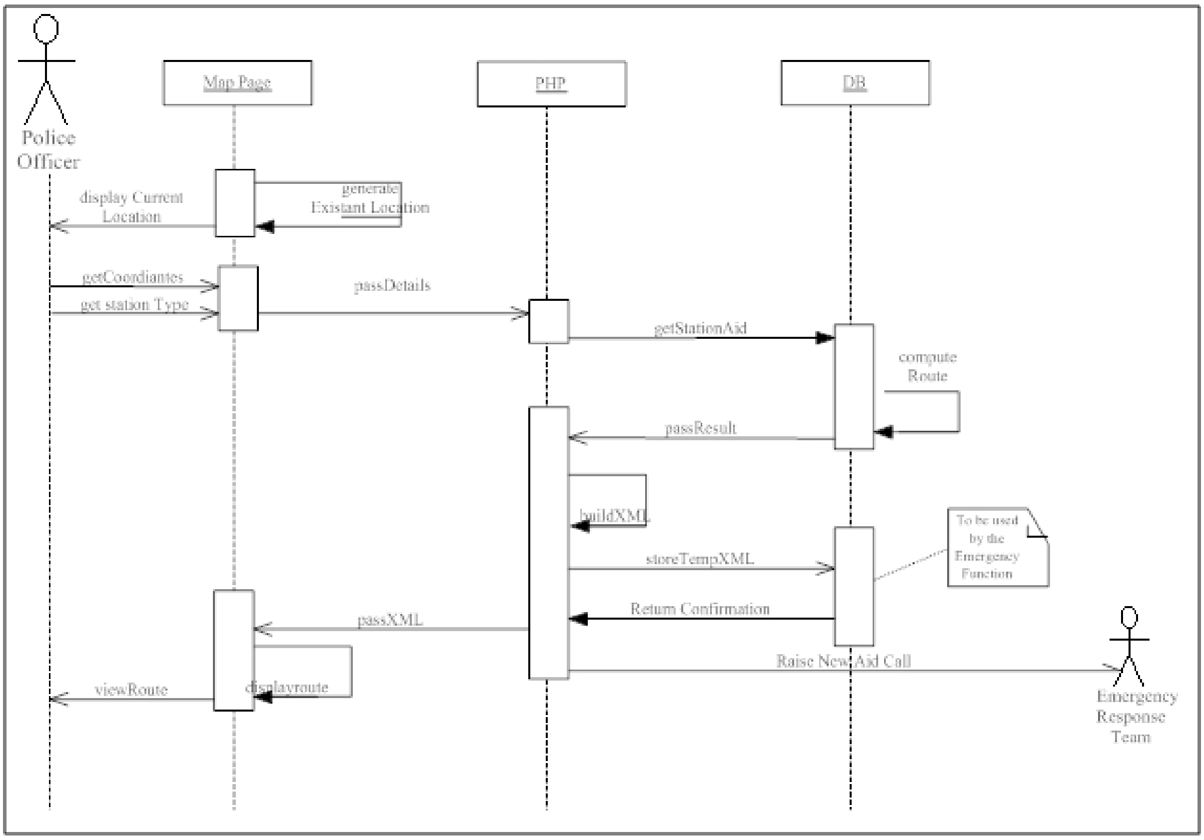


Figure 25 : Sequence diagram for Emergency Call

Similar to the above exercise (Figure 24), the steps involved in creating this algorithm are explained to the reader. The methodology used to tackle this scenario in this project can be reviewed in Figure 27. Such methodology is following the result of Thirumalaivasan & Guruswamy (2005). As one can see from the flow chart (Figure 26), the second loop which tests for the nearest station on the road network might take some time to compute, especially if the accident point is outside the boundary of influence of a station (which in this case is set at 100m).

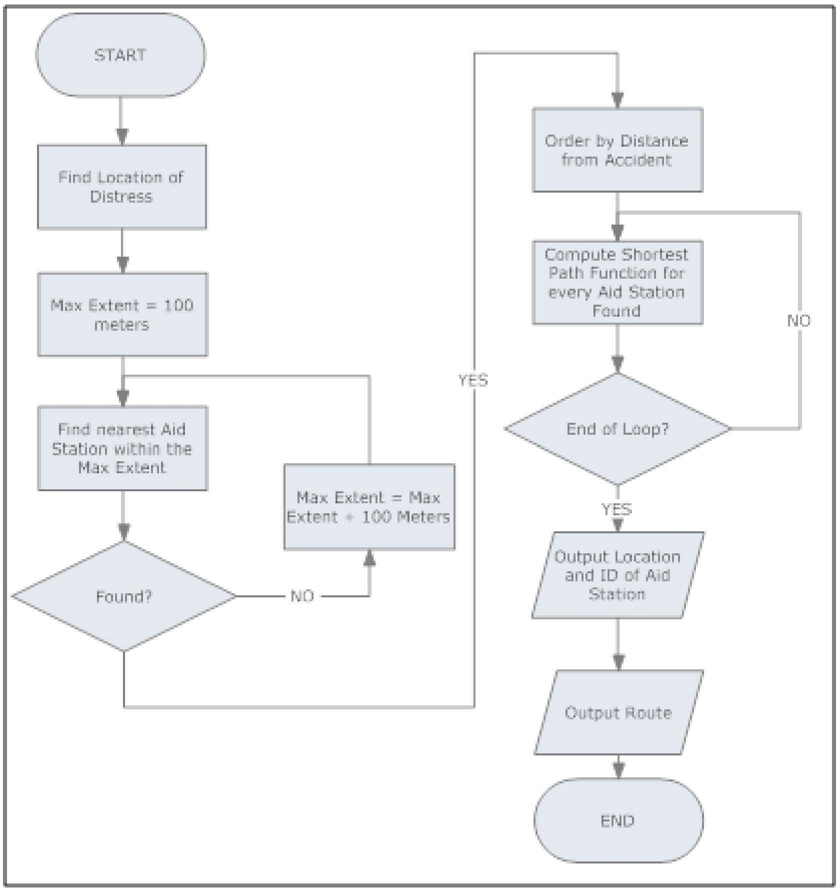


Figure 26 : High level flowchart for Emergency routing

1. *User logs in as a Police Officer (PO)*
2. *A current location is generated*
3. *The current location (CL) and the officer details are stored in a COOKIE*
4. *A map is generated zoomed in around the current generated location*
5. *The PO locates the accident place, he can pan if location is not visible on the map*
6. *A request for an ambulance or fire engine is made*
7. *The PO inputs duration until road can be used again*
8. *The accident location (AL), PO's details and CL are passed using XHTMLRequest to the routing PHP script*
9. *The script calls the PG/Psql function: give_nearest_aid_station passing the AL, type of aid required and the buffer limit*
10. *The PG/Plsql script starts off by finding the geographical location of the AL:*
    - *10.1 Transform the point using PointFromText() function*
    - *10.2 Expands the point by a 100 meter buffer to find the closest street segment*
    - *10.3 Order by linear distance from point to street segment*
    - *10.4 IF Found: Save the street segment ID and Go to 11*
    - *10.5 If NOT Found: Add another 100 meters to the buffer extension and Go back to 11.2*
11. *A voronoi point is created from the AL with a 15KM buffer (same as being shown in XXXXX)*
12. *A loop is created with the returned results*
    - *12.1 For each result, the returned aid station location is passed as the starting location and the accident location as the ending location to the PostGIS's modified function: shootingstar_sp_where_new(). The query asks for a SUM() of the total length of the*

route and puts access for aid vehicles as a constraint

12.2 The shortest route is identified and the GID and location are stored

13. *Now that the nearest aid station is identified (Step 12.2), the aid station location is used as the starting point and the ending location is assigned to the AL*
14. *Steps 16 – 17 from TABLE XXX are repeated for this process*
15. *The route is passed back to the PHP script and as explained in TABLE XXX an XML string is formed*
16. *The XML string is inserted into a temporary table*
17. *The mentioned road segment is set to disabled until the specified amount of time by the PO*
18. *The XML string is outputted back to the user interface*
19. *The XML handler of Openlayers openXMLrequest() is called so as to interpret the XML result*
20. *The routing result is displayed to the user*
21. *Another XHTML request is sent to another script to echo the Estimated Time of Arrival (ETA) of the aid station*
22. *The PHP script queries the DB and echoes back the ETA*

1. *The aid station's auto refreshing script triggers the call*
2. *IF the aid station match the current aid station:*
   2.1 Go to 3
   2.2 Else Go to 1
3. *The XML data stored in the temporary table is queried back by the PHP script*
4. *The data is sent to the user interface as an XML string*
5. *The XML handler of Openlayers openXMLrequest() is called so as to interpret the XML result*
6. *The routing result is displayed to the user*
7. *The aid station can print the route to the accident*
8. *END of Emergency routing*
9. *Go back to 1*

Figure 27 : Pseudo-code for Emergency Routing Algorithms

In Figure 28 a representation of the buffers of fire stations in this project can be found. Such a buffer will be used by the nearest aid station algorithm as explained in Thirumalaivasan & Guruswamy (2005).

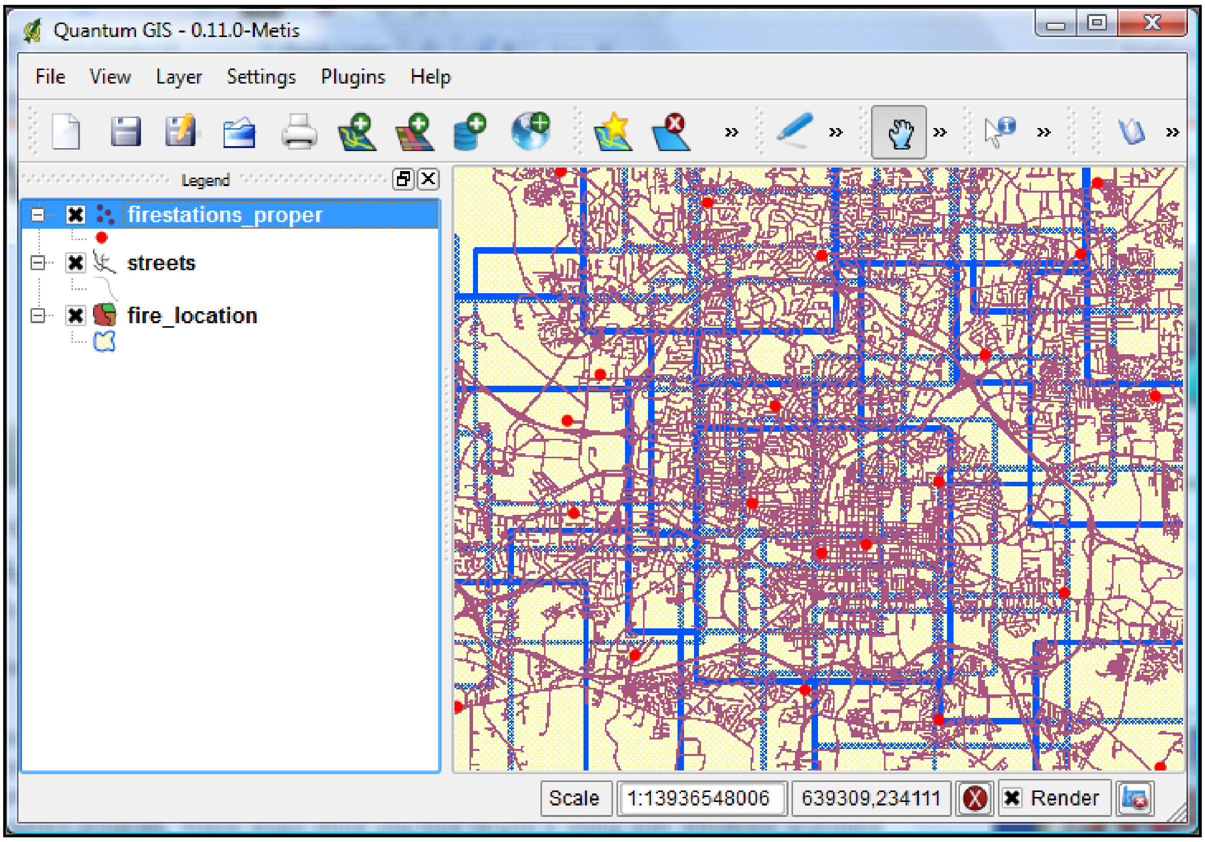


Figure 28 : Buffer zones around fire stations in my data

5.12 Summary

Preparing a Location Based Service requires planning prior to development. The service needs to be user friendly in order to ease the collaboration between the user and the system. As explained earlier, this system is enhanced with a GIS, which similar to any other IS, needs a stable methodology to handle it. This chapter reviewed the methodology used and discussed what benefits were gathered from this exercise.

In the next chapter, reviews about case scenarios in which the software can be used are given. Such scenarios can be viewed as test cases for the software, where the functionality of the software is discussed, with a technical discussion on each of the steps involved.

Chapter 6 : Testing

6 Testing

6.1 Introduction

Following the preparations mentioned in the previous chapter on how to obtain the best results possible from this software, the Testing stage follows. As explained earlier, the chosen development methodology is DSDM. Such a methodology relies on prototypes, which need to be continuously updated and tested as separate modules. In this section case scenarios are going to be studied, in which the software own sections will be analysed and tested, and the results will be documented.

6.2 Case Scenarios

6.2.1 Users Access

As explained earlier, different users have different types of access. Each of the users need to have rights to access different sections of the software, and in this section each of these rights need to be verified. Table 11 shows what is expected to happen after every user logs in. Such behaviour was observed in all the cases, and whenever needed, bug fixing was done accordingly so as to be able to move to the next scenario, with this section working. Since the users on the system were generated via a randomising script, the generated password stored in the data base had to be over-ridden by a query from the Database Administrator so as to read the data stored. Since this is synthetic software, the passwords are stored as text, but it is strongly recommended that such practice is not used in any other environment. Passwords should be stored hashed, and the scripts should compare the result of the hashed password.

6.2.2 Testing Route Planning: Comparing the Results

In this scenario, the results of the three different routing techniques are discussed. The starting and ending locations are kept constant so as to keep current weather conditions constant. Also road access and gradient are kept constant. Screenshots will be used to show the results of routing.

Inputted User Name	Expected User Right	Resulting Behaviour
GLAGS	NoAdmin (Normal User)	After login, the user was presented with the Vehicle Owner Welcome Page, where he can choose to Manage new vehicles or else Plan a route
TBUCK	Police (Police Officer on Patrol)	A random point was generated and a zoomed in map with a 50 meters radius was presented to the user. The interface for the Police Officer to map a new accident
ADMIN	Admin (Road Administrator)	After login, the administrator was presented with the Road Management page, where the options to list disabled roads or else add a new road segment to the disabled list is provided.

Table 11 - Results of Usage Rights Testing

Figure 29 : Routing Result for Shortest Distance

Figure 30 : Routing Result for Shortest Time

Figure 31 : Routing Result for Fuel Friendly

As one can see from the above diagrams, the route is being adjusted according to the route chosen by the driver. As explained earlier in the document, the routing results are based on the algorithms proposed, thus taking into consideration the choice of the user. The slowest algorithm is the Fuel Friendly, and the simple Shortest Path is the fastest. The Shortest Time algorithm is trying to make use only of road segments which are 30 miles per hour or more only.

6.2.3 User wants to find the Most Fuel Friendly Route

In this scenario, the User would like to check which is the most fuel friendly route for him, based on his default vehicle. In order to do this, the details of the default vehicle will be pulled out from the Database, and displayed to the user for confirmation. Once the Routing function is called, details about the vehicle and weather details are passed to the PG/PLSQL function. Once the result is computed, the reply is echoed back in an XML format, which will be interpreted as a line route by OpenLayers method to parse XML. Using UML's Sequence Diagram, the sequence of events in this scenario is explained to the reader.

Figure 32 shows a screenshot of the output computed by the system with regard to the vehicle details previously entered. The vehicle details can be seen further down the screenshot, and the route is the red line. Further on in the document, different results as an adaptation to the change in data will feature.

In the following scenario, the same starting and ending point were kept constant, and the default vehicle changed. The vehicles used had different engines and also different types (a small vehicle and a Heavy vehicle) so as to show the difference made on routing when it comes to different road gradients. The road segment chosen can be seen in Figure 33 marked in orange, which is a slope with a gradient of 15. Thus the vehicle needs to climb a hill. Marked in green is the Shortest Path route which would have been given to any vehicle if the fuel friendly algorithm is used. The red path is the one given to the 5000cc vehicle, which being a large engine vehicle is not affected by the slope. The total length of the route given is 6.38 KM. The same path is given to the 1200cc vehicle, which is outlined by the brown dotted line. As one can see, the blue route is the route given to the small engine vehicle, which despite being a bit longer; amounting to 6.41 KM is much more economical for the vehicle since less effort is made by the engine.

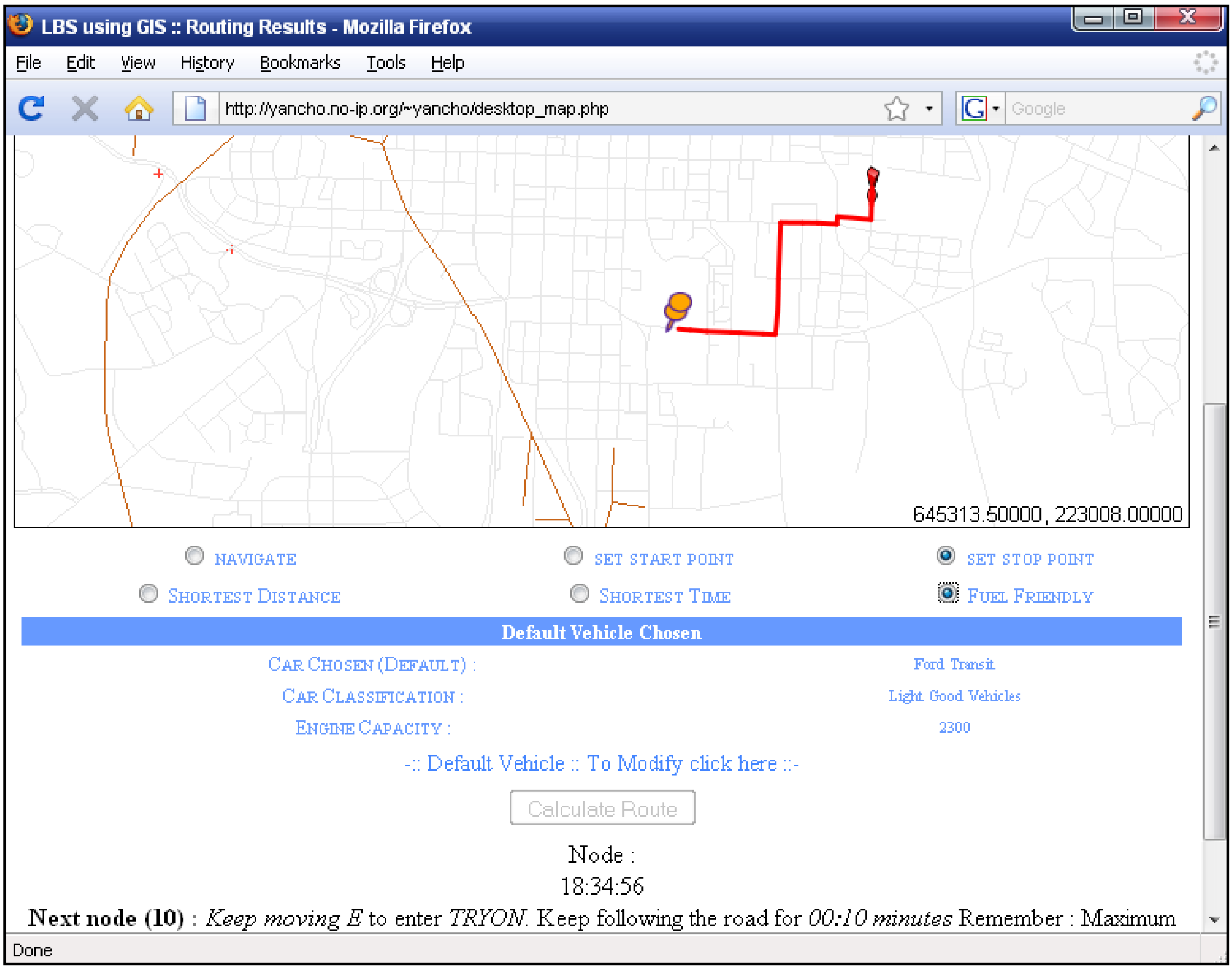


Figure 32 : Screenshot of Least Fuel Result

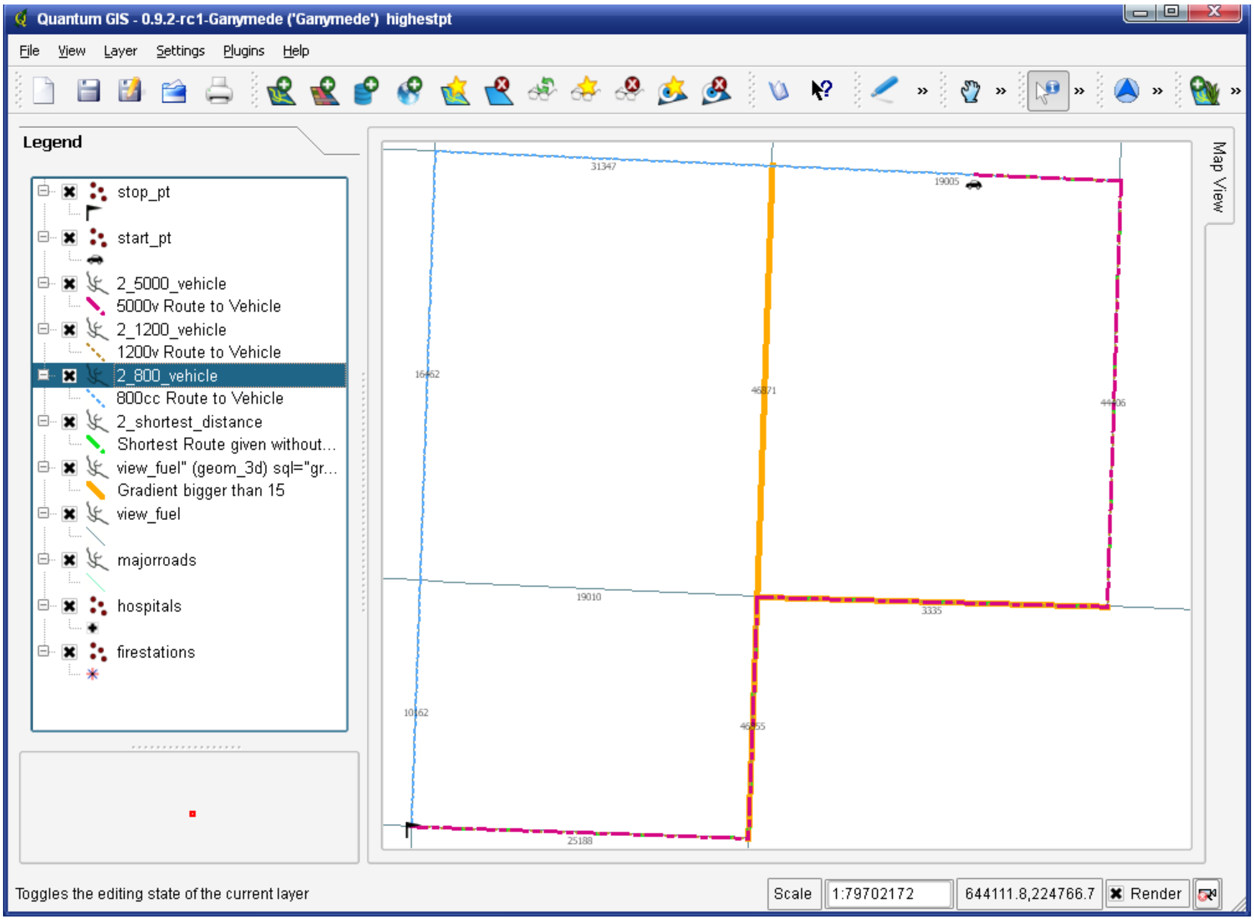


Figure 33 : Showing the different paths in a Fuel Friendly routing

As one can see from the above diagram, the routing was altered depending on the vehicle in question. The height of the vehicle from the road also changes the route as can be seen in Figure 33 . In this scenario the vehicle will be routed since one of the road segments is flooded with rain water. This case the scenario is going to be performed by two vehicles only, a 5000 cc (red) and an 800 cc (blue) with the Shortest Path route being the green one. In this scenario there is also another discrepancy in the length, having 2.86KM for the foremost and 6.65KM for the latter in total length. The suggested heights for a vehicle to pass were taken from SmartMotorist.com[29]

[29] A website with tips to drive safely : http://www.smartmotorist.com/driving-guideline/tips-for-driving-in-rain.html

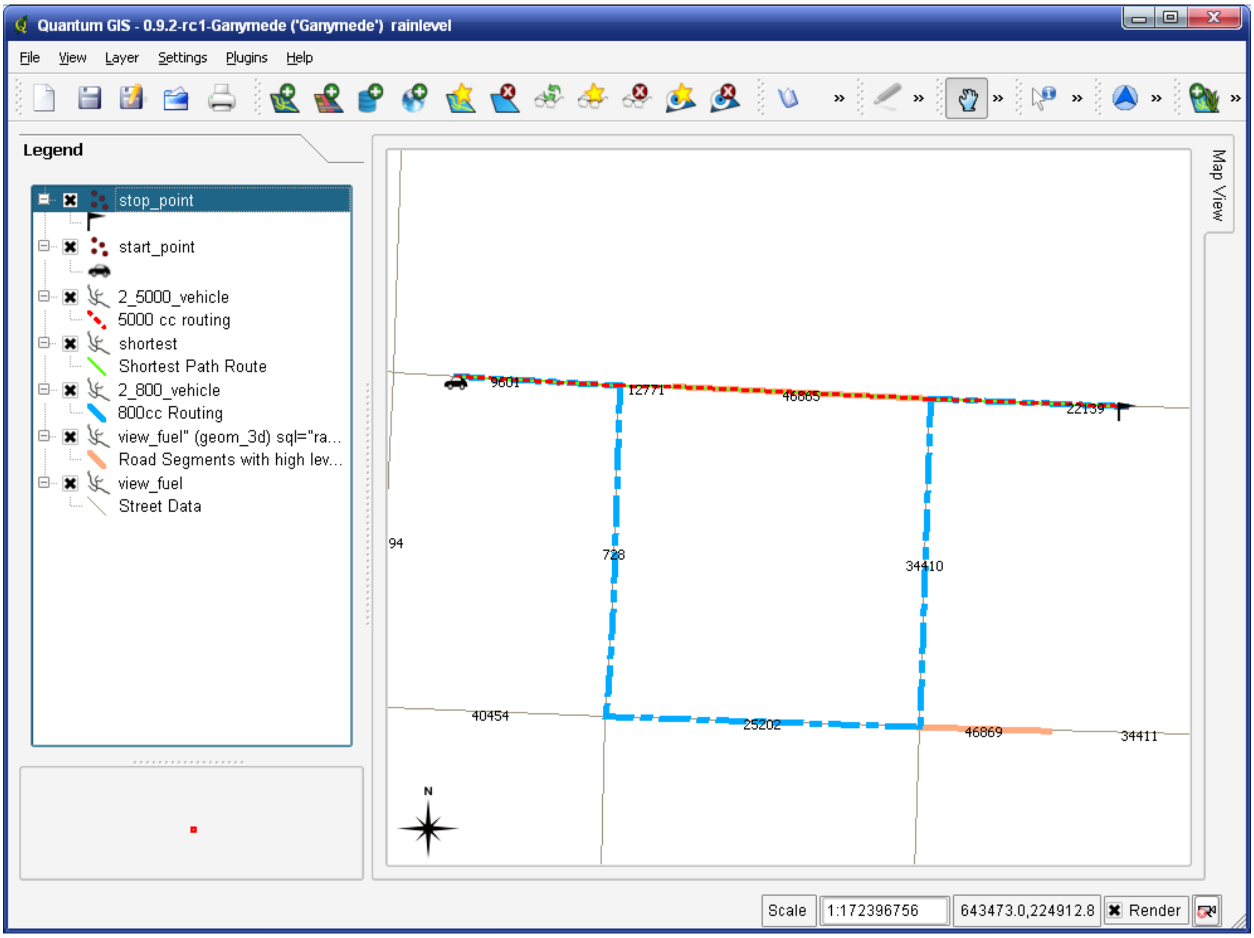


Figure 34 : Routing depending on Water Levels

6.2.4 Police Officer wants to report an accident

In this scenario, the Police Officer is logged in from a mobile device, thus position awareness is a must. For the testing which was run whilst documenting this chapter, the results are shown in Figure 36 where the route needed by the Fire Engine is shown in red, which took 6.91 seconds to compute according to FireBug[30]. The Central red target in the middle is the randomised location point of the Police Officer making the report.

[30] FireBug: Open Source Web Development Engine for Firefox: http://getfirebug.com/

Figure 35 : iPhone Simulator showing the route given to the nearest aid station

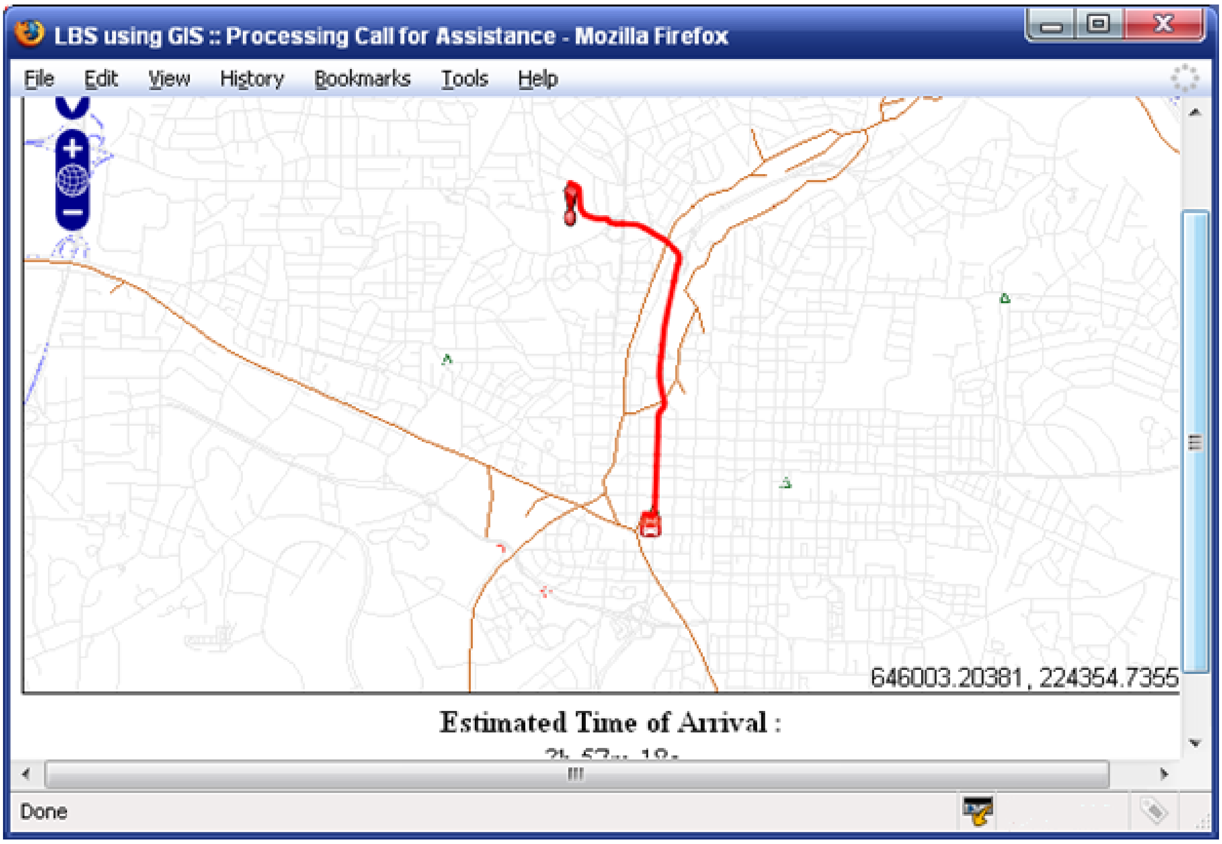


Figure 36 : Showing routing of a Fire Engine

On the other side of the call, thus, the interface for the Emergency response teams, the update has been processed successfully as can be seen in Figure 36. The data is being passed through the different modules of the Emergency section as per the Sequence Diagram (Figure 25).

In, the above figure, one can view the shortest route computed from the nearest Fire Station. The nearest station was computed by comparing the total length needed from all the stations which are nearest to the accident. The chosen Fire Station is nearer to what seems to be much nearer station on the left bottom of the accident location. As explained earlier in the document (Section 4.10.2), the nearest station is calculated on the actual road distance not on line distance.

In Figure 37 one can review the new road segment which was added to the ban list for 2 hours due to the accident reported. The banning, as described earlier will be removed automatically after the banning time is passed. The reason includes also who was the Officer who ordered the banning of the road, and thus reported the accident.

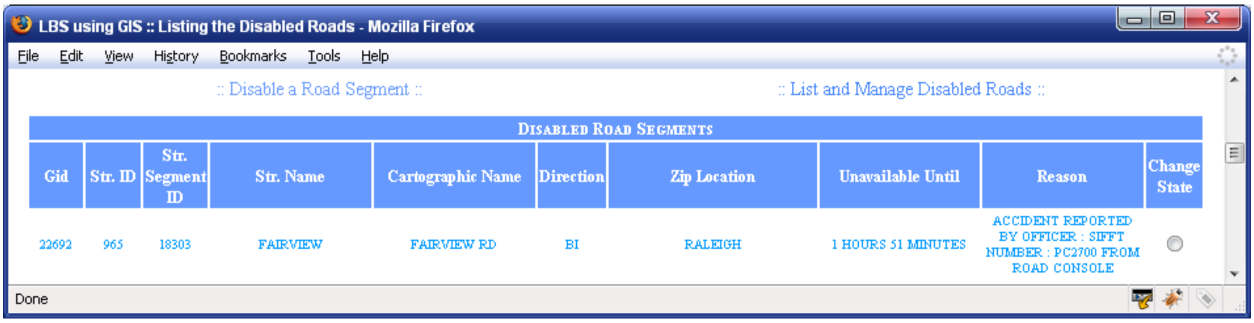


Figure 37 : Banned Road Segment due to Accident

Testing for this scenario has been performed by checking the database for the creation of the new incident reports. Testing has shown that such reporting was successful and in all the cases, the emergency response teams, were informed within 3 minutes, as per the requirements (Appendix I).

6.2.5 Updating Weather Conditions

Testing for this scenario was done using the Database Administrator role, by inspecting directly the database. The table *zip_weather* was inspected before the weather update was run. After the weather generating script was initialised, another look was taken at the table.

Since the conditions are generated randomly such tests tested only the functionality of the script, and not the outcome.

6.2.6 Automatic updating the road network

The CRON jobs were set up in order to periodically update the status of the road network. Testing such an event was done in two steps. The first part of testing was done by inspecting the database prior to a change. The two updating scripts were run manually in order to be able to observe the results. Data was successfully updated. The outcome of automatic updating was studied by observing log files showing that the script was initiated. Data was also observed to have changed after automatic updates.

6.2.7 Web testing

This test case studied the function ability and accessibility of the web pages. Such testing was split in three steps:

- Testing all functions with the Gecko Web Engine (Firefox 3 was used);
- Testing mobile pages with the WebKit Engine (WebKit for iPhone[31] was used);
- Testing all pages confirm to W3C Standards;
- Testing there are no dead links within the web pages;

The first two tests were done by loading each separate page with the web browsers mentioned. In the case of the iPhone, a simulator[32] was used. The third test was done by loading each page to W3C Validator[33]. After bug fixing, all of the pages were found to be confirming to W3C standards and are HTML 4.0 Strict. The CSS which styles the pages was also tested by the W3C CSS Validator[34] and was found to be confirming to standards. The last test that was conducted on the web pages was to validate all the links. This was performed by using the W3C Link Checker[35]. To perform such test, the security challenges were disabled, so no password checks were made. Such tests concluded that the web pages in this system all observe the W3C Web Standards.

[31] Webkit version for iPhone Website: http://webkit.org
[32] iPhone Simulator Website: http://testiphone.com
[33] W3C Validator Website: http://validator.w3.org/
[34] W3C CSS Validator Website: http://jigsaw.w3.org/css-validator/
[35] W3C Link Checker Website: http://validator.w3.org/checklink

6.2.8 Time performances

In the System Requirements (Appendix I), time limits are mentioned. This test scenario is set up to make sure that such limits are respected. Each test was performed three times, with every test starting on a cleared disk cache. The DBMS server was restarted in order to clear the cache (Ipa, 2006) and the OS disk cache was cleared using (Smith G. , 2008):

sync

sudo echo 3 | sudo tee /proc/sys/vm/drop_caches

The web browser cache file was also cleared for every test. The tests conducted were:

- Using Firebug[36] check that the routing script is called in less than 10 seconds;
- Using Firebug check that the map loading time over LAN is not more than 30 seconds;
- Using Firebug and a proxy, check that the map loading time over the Internet is not more than 60 seconds;
- Using 'EXPLAIN ANALYZE' Check loading times of the three routing functions.

The first test was conducted and in all the three tests the routing script was called in less than 1 second (Table 12). This was regarded as a good achievement. The second test was performed by loading the routing page with Firebug enabled. The results were all under the set limits (Table 12). Such tests showed that the objective was reached. The third test was run by forcing the browser to connect to a proxy server[37] and then connect back to the server. The tests resulted in longer times when compared to over LAN but still under allowed limits (Table 12). The last timed test consisted of checking how long does each routing function takes to be called. The three vehicle functions and the emergency aid function were tested and results can be seen in Table 12. As one can see the results all meet the expected timings, with the exception of the fuel friendly routine. Despite having no pre-set limit, this function still requires more development so that timing can be lowered as explained in the Chapter 7.

[36] Firebug is a Web development tool; Website: http://getfirebug.com/

[37] Megaproxy.com (Website: http://www.megaproxy.com/freesurf/) was used

Test Name	Test 1	Test 2	Test 3
Call of routing script	786ms	840ms	730ms
Map loading time over LAN	2.4s	3.6s	3.4s
Map loading time over Internet	18.4s	11.5s	27.3s
***EXPLAIN ANALYZE* shortest_path**	1186.131ms	1438.467ms	1275.345ms
***EXPLAIN ANALYZE* shortest_time**	4271.675ms	5124.452ms	4004.548ms
EXPLAIN ANALYZE fuel_function	130463.814ms	148335.531ms	184381.659ms
EXPLAIN ANALYZE give_nearest_aid_station	28895.367ms	27351.783ms	26854.168ms

Table 12 - Timed Test Results [38]

6.3 Summary

These tests have concluded the section dedicated to testing. This section described what each of the most important sections of the software do. Each section was also documented in what is supposed to do, by using UML diagrams, in order to clearly explain the steps involved in the algorithms used. As one can read in the Methodology chapter, the algorithms used here are all based upon other studies.

In the next chapter, a discussion on how the area of research proposed by this study was tackled is conducted. In the Discussion chapter the methodology used will be discussed, identifying strong points and also areas of future development.

[38] Legend for table: ms : Milli Seconds; s : Seconds

Chapter 7 : Discussion

7 Discussion

7.1 Introduction

This chapter will first review the methodology used, then the area of study and concluding with the importance of this project to other future projects.

7.2 Choosing the right Methodology

In this section, a review of the methodology chosen for this project will be given. As explained earlier on in the document, the chosen methodology was DSDM. Like in every other thing, it had its advantages and also its disadvantages. This section will analyze why DSDM was the right solution for this project and also what were the disadvantages.

Using DSDM, budgets were kept to a minimum, basically making use of existent hardware, and making use of Open Source software. These two combined, especially the Software part helped the budget factor, since all the software needed was downloaded freely. DSDM was also a good choice since it is a Rapid Application Development (RAD) methodology. Time factor was more important to meet, and at times it looked almost impossible to reach, however by making use of the MoSCoW technique which is used in this methodology, such deadlines were met. Prioritising requirements was essential so as to meet the deadlines, and allow the software to be presented in a working condition. The priorities will help as a guideline in order to make sure that no time is used incorrectly on components which are less important before higher ranked priorities are fulfilled. This guideline helped driving the project to meet the designated deadlines. Whilst planning a system, ideas keep flourishing in; however having a hard copy of these requirements will enable the development team to make sure time is dedicated to the most important functions.

Because DSDM was used, the project was outlined at the initial stages, by providing UML Diagrams, Flowcharts and Screen mockups. The functionality targets of the system were outlined in the System Requirements document (Appendix I). Key concepts were discussed prior to commencing the design. A through research was conducted reviewing the past, current and future state of LBS and GIS, thus outlining the key points on which this system has to follow. The list of requirements (presented in the MoSCoW requirements) was drafted at the early stages of the project. Another reason why DSDM was chosen is the availability to use prototyping. Modules were created and tested separately before the system started to be combined, resulting in a better product.

7.3 Validity of the approach

Following the Literature Review, the lack for fuel consumption algorithm was felt. Thus, by proposing this algorithm, in conjunction with the already present Open Source Algorithm for routing a new technique to give the shortest path, can lay the foundation for better environment friendly routes. The way the functions proposed in pgRouting have been extended, allows for further development on this area. This can lead other project teams, working in the Open Source field to continue enhancing this model. Basing the ideology used in this project on the Open Source methodology, it is felt that the project was a step tool for further development, thus making use of other code available and laying the foundations for similar projects.

Another enhancement seen in this project is the outcome of the fuel consumption technique. This technique, when used in a real life situation should minimize consumption in a way that can be managed. In contrast with techniques used earlier on, such as the one used by Ericsson et al (2006) and by van der Voort et al (2001), this technique does not rely on the driver. This technique, relies on the route it is chosen. Whilst keeping a shorter distance helps fuel consumption, since fuel needed is kept to a minimum, efforts done by the engine are minimised. An engine to vehicle ratio is kept so as to help this technique work, basically ensuring small engines to be offered a plane or downhill route as much as possible, whilst, vehicles with big engines, thus have more power at their disposition can be offered steeper roads. This technique can sometimes result in giving steeper routes to vehicles when there is no other geographical option available, however in most of the cases, it resolves many issues. Fuel consumption in this technique cannot be measured due to the traffic found in the roads and the behaviour of the driver, which are two important variables in this matter, however engine use, and low gear usage are common knowledge that take up more fuel (Oneyama, Oguchi, & Kuwahara, 2001).

Having the network update itself with the current traffic and weather situation, will enable the system to provide the user with a closer picture to reality. The current traffic situation found on the database, will never be the real world situation. One must mention that even if the data were not synthetic, and was being gathered by making use of sensors and relying on collaboration between users, real scenario would be different. The time taken to process data being uploaded from the sensors by the main server will be very long than an accident to happen. This system too has a level of user collaboration, in the sense that the police officer, when reporting a traffic accident, will be disabling the road access automatically, making routing of the vehicles easier by not traversing the congested route. The Police Officer role is a distinct role which could easily be joined with that of a Vehicle Driver, thus allowing the Vehicle Driver to report accidents too, however it was decided to keep two separate users for the scope of outlining the system, making sure to explain what type of interaction is expected from each user type. In a more advanced system, these roles can be mingled together to prepare a real world scenario, with the limitations explained earlier on.

An important feature of this project is that throughout the code, Open Standards were used. This means, that migrating this system to another system which makes use of Open Standards should be an easy task. Basing the theory on the OSGEO Consortium and following the guidelines published of the OGC, such Standards were observed. An advantage found whilst loading the data was due to the fact that the data collector integrated OGC standards whilst building the data, such importation was quite straightforward Also, whilst modifying data, the GRASS team made sure to keep observing such standards, which continued helping the final data acquiring. The GRASS book authors, joined many different layers of data from different sources, thus had to project all layers on one projection level. This projection, and coordinate system change did not affect the outcome of the output, since during all the modifications Open Standards were observed.

The Road Administrator, can be seen as an Expert in the area, thus, certain advice can be given only from a person pursuing that career. An important role that the Road Administrator should do is calculate the banning times. When a ban is recorded manually, the time taken for it to be usable again relies on the knowledge of the scene by the Road Admin making use of Expert knowledge. The system is relying also on the experience of the

Police Officer when it comes to banning a road due to an accident. The police officer would need to input the time needed for the road to remain inaccessible for traffic.

7.4 Using the project

This system can be further developed to make a stand alone project which enables the users to prepare a route. However before this project can be deployed in real life scenarios, data cleansing techniques on the syndicated data has to be prepared. Such data collections need to be agreed with the supplier in order to always have the correct feed, since the system relies a lot on the automatic feeds. If the Weather or Traffic conditions feeds are changed and such change is not modified in the system, the whole network would be lagging far behind the actual road network situation. This can lead to severe problems, in having traffic routed to closed roads, and the possibility that other segments which are functioning not used at all. Also, this project, would require, some recoding since it was not built to serve for other purposes, unlike the separate modules which were built for such a purpose.

Also, this software needs heavy improvements. Routing decisions, in some cases takes too long, and require an amount of bandwidth which till now, with the limited mobile bandwidth available (Rondin, Hailes, & Li, 2008), is far beyond usable. Table 12 shows that shortest path routes give a response in much shorter time than the Fuel Consumption query. Such results were drawn by using '*EXPLAIN ANALYSE*'.

Such a query can sometimes fail if the execution time extends the maximum allowed by the server. Also as one can see from the table, a correlation between the length of the path and the amount of time taken exists. This means that the routing function for longer distances, had this project be used outside in the real world, would take a lot of time, which in most cases would not be appreciated by drivers. Another drawback of this process is that if during the fuel consumption algorithm finds a point where all the roads connected to it do not respect the vehicle to road ratio, will sometimes fail completely. Many times such occurrences get captured via the fall back functions which were developed for this scenario, which unfortunately continue to add waiting times for the route, however during testing some cases were developed which were outside of the debugging ability.

Also the map data lacks user interface. The street names do not feature in the streets map, nor do the emergency stations' names. This is due to the project's main purpose to show the new routing and how it adapts to the network's condition. Such interface enhancements were seen as extra and did not form part of the objectives in the research.

7.4.1 Reusing the Functions

Despite the type of code used in this project, it is still very easy to port this code, naming the functions used for routing for another project. These functions, found in the Database_Functions folder on the provided media, can be loaded easily by issuing this command at the command prompt:

$$psql\ -f\ functions\_bkp\ DatabaseName$$

This will enable all the functions in the new PostGIS extended database and will let the new user to use them. These functions were written in Pl/SQL and not as C libraries so as to facilitate their modifying with limited knowledge of SQL. Also these functions are heavily inline documented, thus making it easier for altering. This technique was used to facilitate the collaboration from other persons within this project, after the thesis research is completed.

7.4.2 Reusing the XML

Another file found within the media is a Sample XML generator which can be used as a template for Traffic syndication services. This can be extended to offer a service to other organisations by acting as a Traffic Status provider. By providing the template to the "clients", they can modify their system to read the data, whilst this project could provide properly structured XML data. An extract from the XML can be found in Figure 38.

```xml
<incident>
	<segments>1</segments>
	<type>Road Works</type>
	<road_1>
		<gid>28009</gid>
		<cname>MORGANITE CT</cname>
	</road_1>
	<accident_access>1</accident_access>
	<description>Electricity works in road segment MORGANITE CT</description>
	<unavailable_time>3804691</unavailable_time>
</incident>
```

Figure 38 : XML Extract

As one can view from the above extract, data is well structured, and allows also for a combination of roads being involved in a car accident, which is done by having segments equalling to two and a <road_2> tag added to the XML. This structure can be read easily using any DOM method, for example in PHP, the scripting language used in this project is done by firing a simple method : SimpleXMLElement();

7.4.3 Reusing the Design

Another usage for this project would be to enable it in NON-GPS enabled mobiles. Since it uses WAP, this service can be given using Mobile Web Browser, be it GPRS, WAP, EDGE, UMTS, 3G or the like. What is needed is that the Network Company detects the position of the client using Triangulation Methods, or Cell of Origin, as explained in Section 2.5 instead of GPS location.

Another problem of this software is that the interface was not built to be used by touch screen devices, which according to Canalys Navigation (2008) is on a steep rise this year, recording the second highest best quarter on record for the European, Middle East, and Africa (EMEA) Region. Thus it would be a huge improvement had this project to be implemented on such devices.

7.4.4 Enabling Emergency Services

The above method can easily be the basis for the introduction of e112 services. Routing of Emergency Services is already a core function within this project, thus, making use of the technology to capture the location of the client can easily serve as the starting step for emergency routing. In a case like Malta, where there is one general hospital, this can be enhanced to provide the starting location of the ambulances from Poly Clinics too.

7.5 Roundup

The word "Reusing" featured a lot in this section. This is mainly the idea that triggered this project. The idea of building an Open Source project was to make sure that the system can be enhanced by third party. This means that the system will be available for other developers to modify the code in order to enhance it both for this system and also to enhance the project for further using. In contrary to what Nasr (2007) found out in his research, this project started with the main scope that it builds on what is existent and enhance it for the benefit of the community.

Chapter 8 : Conclusion

8 Conclusion

This research has reviewed the current situation in the field of Location Based Services. Geographic Information System play a very important role in such services and helps it to provide the value added to the users and business entities. Recalling from the Literature Review, one can state that routing, has been interesting scholars for decades. However, it is only until latest studies where the Environmental impact has been taken into account. Thus, it was felt that this study should propose algorithms that help the environment by providing fuel friendly routes to the vehicles.

8.1 Findings

As shown in Chapter 5, Figure 23, a fuel friendly routine has been developed, which provides an algorithm in a reasonable stipulated timeframe. This algorithm makes use of weather and traffic conditions in order to propose such route. Also taken into account is the relationship between the engine and the gradient of the road in order to provide as much as possible routes where the vehicle drives in a plane or downhill.

The other two routing algorithms used in this project (Shortest Time and Shortest Path), also use the current state of the traffic. Thus, if a road segment is blocked due to an accident, traffic or otherwise, such a segment will be left out from being given to users until situation is restored back. The three algorithms make use also of directional information on roads.

Such a project was developed using only Open Source software. This was a prime target before the project and it was successfully accomplished. In order to keep this project alive, it is felt that the source code and any related material will be granted to public domain, according to the Open Source Initiative (OSI) approved licence of the Q Public License (QPL)[39].

The project was developed in a Web Environment. Thus, this would allow any user who is connected to the Internet to be able to reach the project and perform the routing required.

[39] QPL contract: http://www.opensource.org/licenses/qtpl.php

As explained in the requirements, such project is compatible to the Gecko Engine for desktop interface and WebKit for the mobile interface. All pages are verified to be compliant to the W3C Standards of HTML Strict.

8.2 Summary of Contributions

In this section a review of what this project contributed is analysed. Especially since it is an Open Source project, the main idea was to analyse what is already being used by the community and enhance it with new ideas. Also a very important factor was to make sure that any section of the software can be used by other developers, in order to continue improving how the Open Source community deals with GIS and also improves the LBS initiatives it is proposing.

This project is providing an algorithm, that when called upon can give the most economical route for different types of vehicles. Such algorithm was computed by extending the functionality of the sp_shortest_distance() algorithm provided in the pgRouting extension, and adds to it attributes of weather, traffic conditions, and also the relationship of engine to the slope.

Another achievement of this algorithm is its dynamic adaptation to incidents in the road network. Therefore a road can easily be left out from being used in case any type of incident occurs. This helps also the environment since no fuel is wasted whilst the vehicles are queuing after each other in traffic.

This project is also providing an algorithm to route emergency response teams to an accident location. The algorithm proposed makes use of the nearest neighbour theorem, to propose the nearest aid station. This algorithm helps the aiding agencies neither to waste time in checking which the nearest aid station is nor to provide the routing. What is also important in this routing is that the nearest station given is the actual station compared on the road network, thus calculating how distant the station from the accident point is. The distance is not measured via a straight line, but counting the meters involved had the ambulance been called.

Finally this project has extended the functions in pgRouting in order to provide traffic awareness to all the functions used. Thus, two types of statuses have been created, one concerning the normal vehicles, and another which concerns only the emergency response vehicles. The latter can access any road blocked for normal vehicles, as long as it is not blocked for emergency services too. This would allow a real life scenario where there is an accident on a road, and the road is closed to traffic. In such scenario it is obvious that emergency vehicles will still be allowed to pass. If there are other impedances, such as road works, than the road segment would be blocked to emergency vehicles too.

8.3 Limitations

In this section, the limitations which affected this project will be analysed. Also, ideas on how these limitations could have been surpassed will be given later on in the chapter so as that further developers can use such knowledge to provide better solutions.

The most obvious limitation found in this software is that it is a concept software. All of the data used in such software with the exception of the road network is synthetic data. Also the elevation applied on the road network was per area, thus the gradients have a small scale of randomisation.

Another limitation is that the location point could not be achieved using a GPS or any other location finding device. This was due to the fact that the only data network found was that of Wake County in USA.

Unfortunately, due to the above limitation, the routing algorithm cannot be tested on real data, thus proper vehicle data (to compare and contrast the fuel used for the same trip from start to end point with and without using the results of the algorithm) cannot be entered. This leads in having no actual proof if the proposed algorithm is useful or not, and to what extent.

Since this is a synthetic project, many potential users are simulated. This created the need to wear different hats whilst building the system. One must say that this process can lead to certain flaws that a proper person in the system be it the end user or the data manager will not oversee.

This application was also limited by hardware. The hardware used as a server suffers from certain bottlenecks, naming, the memory and processing. This slows down the seeking times of the routes.

8.4 Future Development

Certainly this software allows for other development. All areas tackled in this process can gain from future work. The project will continue to be supported, and it is the thought of the developer that the project will be merged with other Open Source projects, so as to enhance the LBS awareness of the Open Source community.

This project would definitely benefit if the road network, traffic status and weather conditions are generated from real data. This would help the project to make use of Floating Vehicles as suggested by Gates et al. (2002). In this case, an attempt to keep driving habits out of the result, to attempt to quantify the benefit of such algorithm could be made. It is very important to note that this algorithm should not suffer much from the driving habits of the driver.

Also by making use of real data, the service offered can be useful to the end users. This would imply also creating mechanisms in the data handler so as to be able to handle data from different clients without affecting the result timings. Another improvement would be seen if the roads are segmented using a “highway algorithm”. This would mean that the road network will be segmented into a hierarchy; therefore the resulting path will try to make use of fast moving roads prior to giving secondary ones.

One can also mention that a significant improvement seen in the outcome of the result would be the enhancement of the web server which will handle the routing algorithm. This would improve the timings of the query. Another benefit would be if the software will be made compatible with more web browser engines. As explained earlier most of the functions are usable with other browser engines, it is just that no time was allocated to fix certain bugs produced with the use of other software.

8.5 Application in Society

The ideas proposed in this research can be further studied in order to achieve routing techniques which will help both the environment and the user. Such algorithm helps the user in order to allow the car to run on least fuel consumption. With such algorithms in place there is the potential to reduce the pollution emitted by vehicles either stuck in traffic or choosing lengthier, fuel consuming routes.

This research can also be used by other developers in the Open Source community to serve as the basis for their project. The limitations and future development sections can be used in order to enhance the model, and thus provide better routing algorithms for the community.

Appendix I : Software Requirements

I Software Requirements

I.I Introduction

I.I.I Purpose

This project is aimed to provide a Location Based Service model for End users applying Geographic Information Systems to Route Planning Software. End users of the system will be allowed to query a route using either fixed or a mobile devices where the user do not need to enter current location since this will be auto generated. Due to its nature, this model is a concept and will be using randomised data. This model will plan routes depending on the type of route, (decided by the driver), and taking into consideration also weather and current state of the road. This model avoids blocked roads, due to an accident or weather conditions, and re-routes accordingly.

I.II Intended Audience

This document is intended to serve as an agreement between the different stakeholders of the system. All of the stake holders were identified prior to the start of the project ensuring that all of such are involved in the planning of this document, and understand completely each of its parts. The stakeholders mentioned here are believed to be the development personnel, the end users of the system, and also the database administrator. Since this project is a concept one, most of these are not distinct persons but are just identified as roles.

I.II.I Project Scope and Objectives

Following the literature review, it is felt that fuel consumption is not well studied within routing techniques. Thus this project is proposing new features and ideas in the Open Source field in order to adapt these techniques. Also, this project proposes ways as to deal with dynamic routing, by taking into consideration the current state of the road. The objectives this project aims to achieve are:

- To build up a Location Based Service which act as an intelligent route planner. Such a route planner should allow the user to specify variables affecting the route chosen such as consumption and congestion, and build the Information System using these variables.
- To design and build a Geographic Information System that will act as the backend for the system offering these services.
- To allow queries to be planned against live data and thus minimizing risks of finding traffic jams.
- To design and build a front end for the users to communicate with the backend. Such an Interface should also be accessible from mobile devices for the route planning section.
- To provide an updated route at pre-set interval, and allow the user to minimise the risk of congestion along the route.

I.III Overall Description

I.III.I Product Perspective

This project needs to be built in order to incorporate different parts which should interact together. In certain cases, such data is collated from third party sources, in which case should first be cleaned and organised before adding it to the current data.

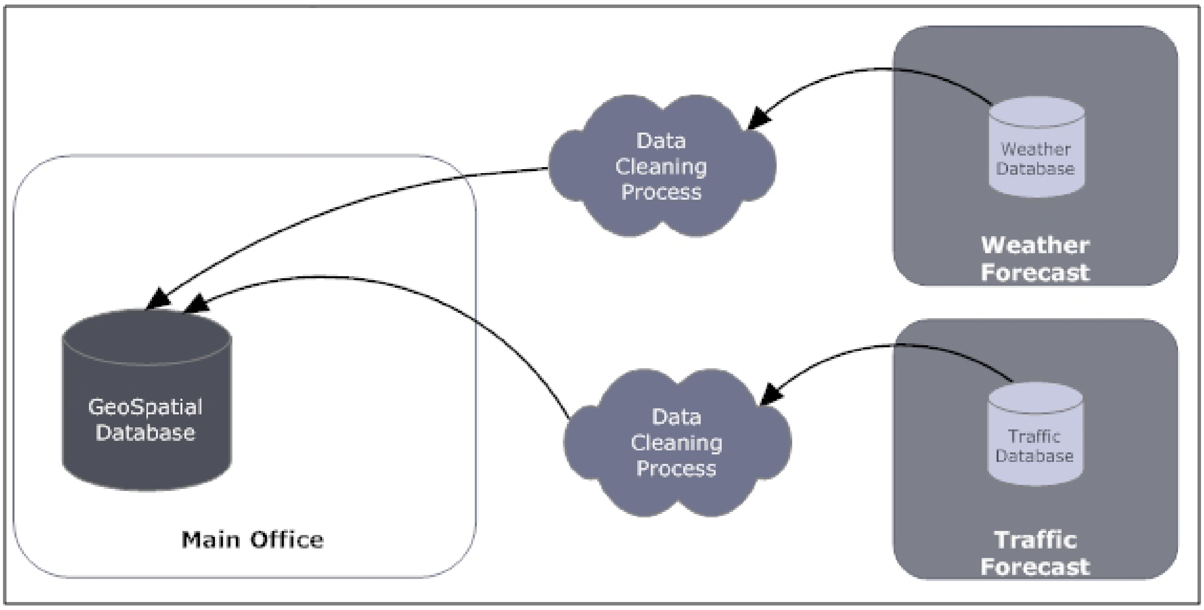


Figure 39 : Database Model

I.III.II Product Features

After the ideas started forming requirements, the following features were drafted and presented for approval. Following one can find a very high level specification of what the project is intended to do

- This software takes the generated starting and inputted ending coordinates of a route and plans it according to the user specifications. The starting location in the case when the software is being used from a mobile device shall be auto computed.
- This software should be aware of the current traffic situation so as to provide an effective online route planning tool.
- This software should allow for emergency response teams to possibly find the shortest route to get to an accident from the nearest aid station.

A more detail list of requirements with their associated priorities, used with the MoSCoW priority guidelines can be found in Chapter 4 Section 4.7.

I.III.III User Classes and Characteristics

The System should cater for different type of users. To further explain what each user's role is expected to be, such roles were identified and a list has been outlined in this section.

Vehicle Owners

These users can be regarded as the end users of the system. The vehicle owners need to be provided with a Web Interface where vehicles can be managed and the default vehicle assigned. Such users will be assigned a login id, which will then need to be used when planning a route. The login is important due to the saved data regarding the default vehicle. A vehicle owner must have the opportunity to pre-plan a route before actually taking it, thus a Web Interface for a Desktop Environment is needed where both the Start and End points are required. Personal data for the vehicle owners is stored in a Database and such data can be edited by the Vehicle Owners themselves.

Vehicle Drivers

Another role similar to the previous one is that of the Vehicle Drivers. Such end users take on this role when they are on the road, and planning or driving along the route itself. Location awareness is a must for this user role. The only option allowed for such a user type will be to specify the ending location of the route and what type of route wants to be executed, if a short distance, least time or least fuel consuming.

Police Force

Each end user logged in as a Police officer and logging in from a mobile device will need an interface to report a traffic accident in a 50 meters radius around his current location. The current location is a randomised point, and on such point the radius where to report to is computed (based on the identification of the nearest aid station).

Emergency Dispatch

The Emergency dispatch of each office will have a web interface refreshing at intervals to show if a call for aid from their station is raised. Once a call is recorded a route assigned for the aid vehicle accordingly.

Road Administrator

A Road Administrator will have the role of maintaining the road network. Weather conditions and Traffic reports will automatically alter the data, but this role has the access to override such status alterations, by adding and/or deleting unused roads. The Road Network Administrator will be the administrative body that takes care of keeping an updated road network.

Database Administrator

This role is given to the users who are in charge of uploading the spatial data to the DBMS. The main responsibility is to find and upload data in the Database. Raw Spatial Data found in Shape format, meaning that, OGR techniques need to be used to transform the data in the desired format. Non Spatial Data handling is another role of this user group. Data which is related to the system need to be secured for use. Data maintenance remains the sole responsibility of this user group, where Data has to be maintained through back-ups and monitoring for data integrity.

A systematic diagram of how the User Roles are identified can be found Figure 40. The arrows in this diagram are showing how the classes of users are inheriting roles from each other.

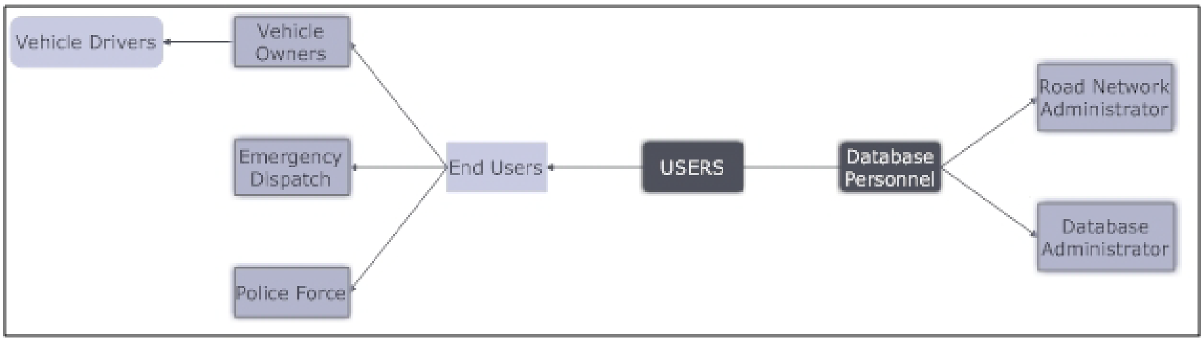


Figure 40 : Users in the system

I.III.IV Usage Patterns

The different type of users each requires different usage patterns. Since the system is as a concept model, peaks do not influence a lot the usage, however nonetheless possible peak times are still identified:

- Due to traffic loading, it is assumed that peak hours with traffic congestion would occur between 7 – 9am and 4-6pm.

One must also note that Emergency dispatch offices will have a web-page refreshing to check if there is an accident reported

I.III.V User Growth

User growth is expected to be as minimal as possible, due to the fact that deployment is conceptual.

I.III.VI User Transactions

This system can be regarded as a Decision Support System (DSS), where the users log in, perform the query needed and log out. In the case when a user logs in to plan a route, such an activity can be regarded as one transaction, since no other user input is involved in the process.

I.III.VII Operating Environment

After reviewing many case studies (see Chapter 3) it was decided that this project will use Open Source Software. The software should be implemented in a Web Interface environment, which has to be accessed through the Mozilla Engine browsers for Desktop Applications and WebKit Engine for Mobile Devices. Both engines support SVG natively and are available freeware which can be enhanced freely due to their Open Source license. The Operating System chosen to run this software was decided to be Ubuntu, which is an easy to use flavour of Debian. The main idea behind this choice was that Ubuntu allows PostgreSQL to create clusters, which are concurrent instances of the DBMS, thus testing can be done alongside the running version and ensure data remains intact until fully tested. Below one can find a list of all the Software systems which are going to be used in this software.

Software Usage	Software Used	Version
Relational Database Server	PostgreSQL DBMS	8.1.9
Geospatial DBMS Extension	PostGIS	1.1.3
Geospatial Library	GEOS	2.1.4
Routing Extension	pgRouting	1.0
Geospatial Analysis Tools	GRASS	6.3.0RC1
Web Scripting Languages	PHP CGI Scripting (fastCGI)	5.1.2 2.4.2
Web Server	Apache	2.0.55
Operating System	Ubuntu	6.06 LTS
Web Mapping Server	UMN MapServer	5.0.0.
Front End MAP Generator	OpenLayers + TileCache	2.5
Desktop GIS Viewer	Quantum GIS	0.9.1

Table 13 - Software Used

The hardware used to model and test this project is described in Table 14. Such hardware can affect the result of this software. Under normal testing environment and using a Local

Area Network testing complied to the Timings described in the Project Specification section I.III of this Appendix).

Type	Model	Scale
Processor	AMD Athlon(TM) Processor	1000 MHz
Memory	DDR	512 MB
Storage	IDE Hard Disk	80 GB
Network Interface Card	PCI Card	10/100 Mb/s
Internet Connection	ADSL	512 Kb/s Upload
UPS Backup	PowerWare	P500

Table 14 - Hardware Specifications

I.III.VIII Design and Implementation Constraints

In this section the design implementation constraints are discussed. Such constraints have been agreed by all the three main stake holders of the system, the Developers, the Users and the Owners of the system. Below one can find a list of constraints for each function in the system:

- Database Access of every routing query should take less than 10 seconds
- A Map generated on a Desktop Interface with a resolution of 1024 x 768 should take not more than 30 seconds in a LAN connection and not more than 60 seconds in a WAN environment
- Routing Result should take not more than 5 seconds to be transferred to a LAN connected client and not more than 10 seconds to a WAN environment.

I.III.IX User Documentation

The different screens on the system used by the software should all be self explanatory, thus it has to be clear what the system is expecting from the user. A User Manual should be provided with Screen Shots of every scenario in which the Software can be operated. Error messages should explain clearly what the software is expecting from the user.

I.III.X Assumptions and Dependencies

In this section, any assumptions made are defined. Such an exercise is important so all the parties involved in reading this documentation, understand any assumptions one of the parties might be doing, so as to have a clear picture what to expect. Dependences are also discussed, thus, any results depending on other data are outlined in this section.

The main assumption is that this software is a conceptual model. Thus it is agreed that all variables in the system will be randomly populated; as follows:

- Spatial data is to include elevation information a fair amount of roads, ranging from State roads to town roads. Location Data on at least two different types of Emergency Services must be included.
- The data on vehicle owners and vehicle details is all randomly generated.
- Current locations use mobile devices will be randomly generated as well. Such software is using random locations due to the problem that it is impossible to be at the current location (in USA) so as to be able to be matched with a GPS.
- Weather and traffic data has to be collected via RSS feeds. When such data is absent the system generates random data.
- A car with a bigger Cubic Centimetre (cc) is able to generate more 'torque' and more 'power' than a car with a smaller ‘cc’

I.IV System Features

I.IV.I Routing Features

This feature can be seen as the most important feature of the project. This feature provides the user with various options to choose from whilst querying a route. These routing techniques were discussed and are made available:

- Least Time Consuming
  - This algorithm plans the route takes into consideration the time factor. In this route plan, the shortest time to arrive at destination must be the most important factor.
- Shortest Route

        - This algorithm plans the route using the shortest path route, thus, measures only the distance on the road covered by the voyage. Distance has to be measured on the road network itself, and must plan the route from the available streets on the network.
- *Shortest Route for Emergency Services*
    - This routing will be the default type of routing used for Emergency services. However, in this case, the routing will be done on the Original Dataset, on the presumption that such vehicles have access to all the roads on the network. This rule presumes that emergency services can have access to any segment that is affected by climate conditions and also by traffic blocks.
- Most Fuel Friendly
    - This algorithm takes into consideration the elevation. Before planning the route, the system computes the best route which keeps a stable elevation versus power. This assumes more fuel efficiency in driving along a plain, rather than up and down a hill.

One must mention the fact that in all route plans (except the Emergency Response one), the dataset must be aware to weary weather conditions.

I.IV.II The Dataset

A Dataset which allows for street segment level analysis has to be found. The streets should have a Geo-Coding data field (using ZIP) and also information about network routing. The street dataset requires information about the maximum speed allowed and also what type of road it is. Street data is required to be segmented with details easily retrievable, for query purposes. Elevation has to be introduced to the streets dataset. Such an operation need to be done by combining a raster file with elevation to a plane 2D representation of the data. This is critical for queries related to elevation and fuel use.

I.IV.III User Interface

In this section, a review on the two mentioned user interfaces is going to be discussed. Details on each of these user interfaces are described below:

Desktop Web Environment

This environment will be used primarily by these types of users:

- Vehicle Owners: These require a user interface where the owners can manage their vehicles and assign the default vehicle. The default vehicle will be the vehicle chosen for the next routing plan when it is accessed from a mobile device.
- Emergency Dispatch: These require a auto-refreshing Web Interface that alerts the office that a new call from their hospital / fire station is requested. Once this user is alerted, the route planned for aid is displayed. These users should have means to report that the aid has been attended so as the system can be alert for another call.
- Database Administrator: This type of user must have an easy way to search and find any road / road segment so as to disable it and sets the amount of time it can be re-enabled again. Also this user need to have an easy to read list of all the road / road segments that are currently blocked, where he can over ride such settings and re-enable them.

Mobile WAP Environment

This environment will be primarily used by these user types:

- Vehicle Drivers: This type of users must be presented with an adequate sized map at an adequate map zoom level where they can click on the end location of their Thus the map should be user friendly and allow for whole network to display. The route should be well identifiable and textual confirmations should be given every end of segment. In this scenario, since actual road position cannot be computed, randomizations are expected.
- Police Officers: This type of users when logged in, should be presented with a detailed map with a 50meters radius, and their current (randomized) position as the centre of the radius. Such users should be made available to click on a current position to report an accident and request for emergency aid. The aid station and the route chosen to be followed should be presented to the Officer and also the estimated time of arrival.

I.V Other Non-Functional Requirements

I.V.I Performance Requirements

- The Open Source software should make use of Valid HTML code. The software should be implemented in a Web Interface environment, which needs to be accessible by the Mozilla Engine browsers for Desktop Applications and WebKit Engine for Mobile Devices.
- The Rain Level in an area (ZIP area) is read from an RSS feed. The street network is compared with the rain level, and in areas where the water is too high for a vehicle to pass will be left out from the routing algorithm.

I.V.II Security Requirements

- The system should be protected with a user identification screen. Since no private data will be transferred between the server and client, there is no need that the transaction is secured with encryption.
- As with every online system, it is a must that the text fields which will edit data when submitted, should be escaped so as to minimize the risks of SQL injection. No data should be submitted from a web form straight to the database without being cleaned and escaped.
- Password field in the database should be stored as an md5 hash string and not as plain text.

I.V.III Software Quality Attributes

- HTML 4.0 Strict will be used for the web-interface. Making sure that every page is validated with the w3c validator so as to make sure that the page can be rendered by most web-browsers. However as explained earlier the software should be made fully compatible with the Mozilla and WebKit engines for desktop and mobile respectively.
- Javascript libraries will be used so as to handle the requests for PHP Scripts
- OpenLayers library will be used to handle map-related variables, setting the start / end location, and display the route.

- The software should be kept to a small clean web interface, so as to minimize unnecessary usage of bandwidth, especially when data is sent to a mobile network.
- It is in the best interest of all parties that such a project is built using prototyping, so as to test every module before implementing the whole project.
- The software need not be a portable software, thus there should be no constraints if the code is written to support the system it is being built upon.
- The Routing Algorithm should take the following structure:
    1. Randomize starting point
    2. Get ending point
    3. Using OpenLayers loadURL() function, the routing script is called
    4. Read indexed dataset
    5. Use shortestpath_sp() pgRouting function using the above mentioned parameters
    6. Two scripts are polled together :
        i. XML File with path is created
        ii. A transaction log stored in a Database with the path from 6i. This means that the path is stored in a temporary table in the Database. This would allow that further on in the process, the path can be read one row at a time.
        iii. XML File is passed back to OpenLayers so it is displayed using displayRoute() function
        i. Using XHTMLRequest() another script is called to generate a random point by generating a random segment on the road on which a random point is generated
        ii. The first row (order by id, limit 1 ASC) from the table is passed back to the display script and echoed to the user. The same row from the table is to be deleted.
    7. Set starting point as the current randomized point
    8. Pause for x minutes
    9. If distance between start and end is more than the user defined distance
        i. Go to 2
        ii. Else : Display : You have arrived

I.VI Conclusion

In conclusion to this document, one must stress the importance that the project is by now fully accepted, and all points here discussed understood. This document marks the end of the requirements stage and acts as a basis for the analysis and design stage. Any requirements that need to be altered hereafter should be reviewed again and upon approval added again to the main document.

Appendix II : User Manual

II User Manual

II.I Preface

This manual introduce you to the Automated Routing Software and helps you get the system up and running right away. In addition to learning how to install the software, you learn how to use the application. You will also get information about advanced routing, and how to use in different scenarios.

II.II Installing the Software

This section will introduce you to setting up your environment to be able to serve this application to others. If you are just willing to use the application then just step to Section II.III.

II.II.I Before you Begin

Before you set up this application make sure your computer which will act as a server meets the following minimum system requirements:

- 256MB of RAM (More memory improves performance.)
- Adequate space available on your hard drive. The amount of space required varies with the server applications you choose to install, however remember that the Cacheing mechanisms, and the database backups will need space.

II.II.II Installing the Server Applications

In this section the Server applications mentioned needed to run this software will be briefly explained in order to have the system ready for the software. The screenshots and data supplied in this manual are from Ubuntu 6.06 LTS, however can apply to any *NIX based system.

Installing PostgreSQL

Using your favourite download package manager, download and install PostgreSQL binary files. In the case of Ubuntu users, you can do:

sudo apt-get install postgresql-8.1

Once the installation is done, some changes are needed. From the command prompt, type in the following command so as to be able to login to the database:

sudo -u postgres psql template1

This will take you to the psql command prompt logged in as user "postgres". Type this command so as to be able to create your own password, remembering to replace **pass** with your own password:

*ALTER USER postgres WITH ENCRYPTED PASSWORD '**pass**';*
\q

This should have set up PostgreSQL, ready to start being loaded with data. If you are stuck or require further information, please visit the Ubuntu PostgreSQL Installation manual at: https://help.ubuntu.com/community/PostgreSQL

Installing PostGIS

To install PostGIS the same technique as explained above should be used. If your OS supports already the pre-packaged PostGIS libraries than the command needed to be typed is similar to the one above:

sudo apt-get install postgresql-8.1-postgis postgis

However, if you are using Ubuntu 6.06, PostGIS needs to be compiled from source. In order to do this first set up the right environment for it by firing this command, in order to make sure that all of the required libraries are installed:

sudo apt-get install build-essential libgeos-dev python2.4-dev libreadline5-dev libperl-dev flex byacc proj zlib1g-dev

Then you need to download the source files which can be done from here: http://postgis.refractions.net/download/. Using '*wget*' is suggested to download the source code and extract it using '*tar –xf'. Once* the PostGIS files are extracted you need to move the files to the contrib directory, which can be done using:

mv postgis-1.1.3 postgresql-8.1.1/contrib/

Once all the files are moved to the contrib directory, change the active directory to the contrib. one (using '*cd'*), and fire the configure command so as to be able to prepare the compiling process. This can be done as:

./configure --with-pgsql=/usr/bin/pg_config

Once the configuration terminates you can compile PostGIS using '*make*' and '*sudo make install*'.

Installing pgRouting

After downloading the necessary files, which can be done by '*wget http://files.postlbs.org/pgrouting/source/pgRouting-1.03.tgz'* you can extract the files using '*tar –xf*' . Once this process has been finished, it is now time to set up the new libraries to be known by PostgreSQL server. Before doing so a DataBase needs to be created, and this can be done by:

createdb -U postgres -E UNICODE routing
createlang -U postgres plpgsql routing

Once the DataBase has been created and PL/PGsql has been added to it, you can go back to the PostgreSQL contrib directory and fire these commands:

psql -U postgres -f postgis-1.1.3/lwpostgis.sql routing
psql -U postgres -f postgis-1.1.3/spatial_ref_sys.sql routing
psql -U postgres -f routing_core.sql routing
psql -U postgres -f routing_wrappers.sql routing

This should finish the setting up of the database. To confirm all settings are working, restart the PgMaster using, in case of Ubuntu:

sudo /etc/init.d/postgresql-8.1 restart

Once the PostMaster has been restarted, enter in the psql command prompt, and type '*SELECT postgis_full_version();*'. The reply should be similar to this:

```
routing=# select postgis_full_version();
postgis_full_version
---------------------------------------------------------------------
POSTGIS="1.1.3" GEOS="2.1.4" PROJ="Rel. 4.4.9, 29 Oct 2004"
USE_STATS
(1 row)
```

This should conclude the tutorial on how to prepare the database for the GIS application. If you find any problems whilst compiling, you can request further help in any of these websites:

- PostgreSQL : http://www.postgresql.org/community/
- PostgreSQL support on Ubuntu : https://help.ubuntu.com/community/PostgreSQL
- PostGIS : http://postgis.refractions.net/support/
- pgRouting : http://pgrouting.postlbs.org/wiki/pgRoutingSupport

Also, if you require just to test the software on your own private server, you can download a VMWare image provided by the team of pgRouting which can be found here: http://files.postlbs.org/foss4g2007/W-12/. Another option would be to use the beta version LiveDVD which is being prepared for the FOSS4G 2008 which can be found here: http://wiki.osgeo.org/wiki/Live_GIS_Disc.

Installing the Web Server

In this section a review on how to install Apache2 and PHP5 with PostgreSQL support will be given. In this case use of the '*apt get*' method is going to be done again. You can modify this to your own Operating System. For Ubuntu and Debian users, you can run this command:

sudo apt-get install apache2 apache2-common apache2-doc apache2-mpm-prefork apache2-utils libapr0 libexpat1 ssl-cert

When Apache2 finishes to install, you can start installing PHP5 (note that some parts of the software makes use of PHP5 functions so please install this version not the 4th one. Also remember that the 5th version is the only one currently supported by the current developers).

sudo apt-get install autoconf automake1.4 autotools-dev libapache2-mod-php5 php5 php5-common php5-curl php5-dev php5-gd php-pear php5-ldap php5-mhash php5-pgsql php5-snmp php5-sqlite php5-xmlrpc php5-xsl php5-imap php5-mcrypt php5-pspell

When asked if you want to install without Maildir support answer 'Yes'. This should finish the quick installation procedure for your server. In order to test, using your favourite text write, such as vi create a file in your public_html directory, naming it info.php and in it write:

```
<?PHP
        Phpinfo();
?>
```

By cycling to you webserver address, and loading the page info.php you should have a page similar to this:

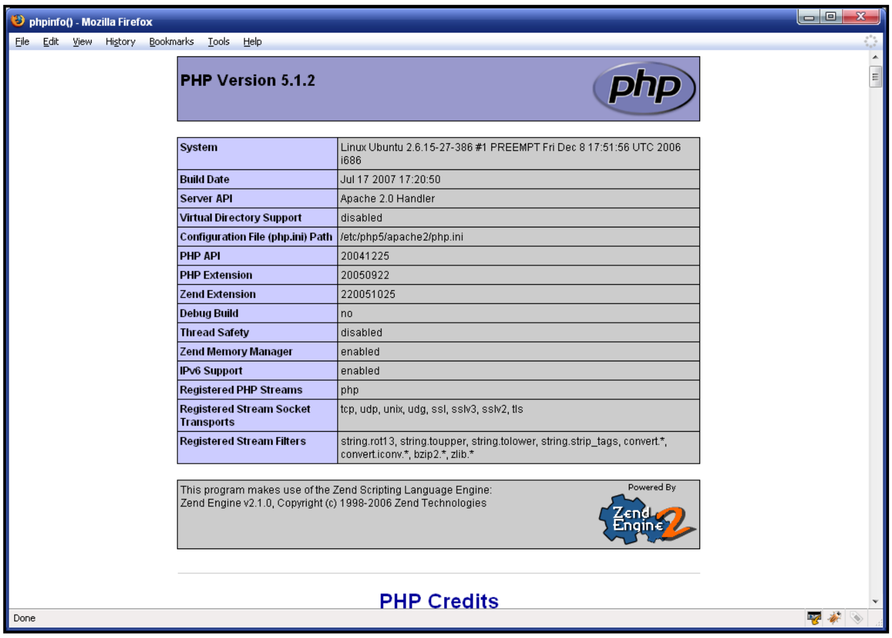


Figure 41 : PHP Screen confirming successful installation

If you require further help it is suggested you visit these websites:

- Installing LAMP : http://wiki.vpslink.com/HOWTO:_Install_LAMP_on_Ubuntu_6.06
- Quick Apache2 : http://www.ubuntux.org/quick-apache2-ssl-mysql5-php5-dapper-install
- ApacheMySQlPhp : https://help.ubuntu.com/community/ApacheMySQLPHP

II.II.III Installing the Software

Copying the files

After testing that the services are up and running, it is time to install the actual software. The package comes in two flavours, either with the attached document as a CD or else you can get it from http://yancho.no-ip.org/thesis/project.tar.gz. What is needed is that you extract the contents somewhere where your Apache2 server can read, either by adding the directory in a <Location> or else in the www-data folder you extract the Zipped project. If you are using the attached CD you can just copy and paste the folder.

Loading the Database

In the root folder you can find the folder database_files. In this folder there is the backup of the database used for this project. To load it, you can use the loading mechanism in psql.

NOTE: This will also add another database, named thesis and adds it to owner yancho. To load this backup you can enter '*psql*' and type:

routing=# \i path/to/database_file.sql

This will take a while to finish. When it finishes, you can query the database by trying the below command and see if it matches the output:

```
routing=# SELECT gid, l_zipname, stname, stseg, length(the_geom) FROM streets WHERE gid='1234';
 gid  | l_zipname | stname | stseg |   length
--------+--------------+-------------+--------+------------------
 1234 | APEX   | LICORICE | 32822 | 119.130743764356
(1 row)
```

Adding the Cron Jobs

In order to have the database updated at certain periods, you need to add the files to crontab. It is up to your discretion in how many intervals you want the database to be updated. In this case I am updating the road network every 5 minutes with different blockages and with weather data every 2 hours. This is done by first calling '*crontab*' then appending these two lines:

```
*/5 * * * * /usr/local/bin/php -q /path/to/generate_traffic.php
* */2 * * * /usr/local/bin/php -q /path/to/generate_weather.php
```

II.II.IV Concluding the Installation

This should conclude the installation. If you find any problems you can open a ticket by registering to the projects TRAC, which is a way to track the tickets opened by the users / other developers of the project. The address is: http://yancho.no-ip.org:8000/thesis. Remember you need to register before to be able to open a ticket.

II.III Using the Software

This section can be used either by users who want to try the software out, without using their own private server, or else by users who already have a running copy of the software. In this section, the examples will be used using the data stored in the Database which was handed in with this project.

II.III.I Logging in the system

As already explained, the system has different types of users. In order to have a working copy, especially for the general users, a list of different usernames with different access is given out. For each of these access types a username can be used for this demo. In the case of a vehicle driver two user names are being given so as to be able to compare and contrast the vehicles.

Note: If you installed this application in your server remember to add a hashing method on the password fields, and that you compare with an encrypted password.

Access Type	***Username***	***Password***
Vehicle Driver	*glags*	*nhzez*
Vehicle Driver	*woede*	*x4o54*
Police Officer	*tbros*	*ez5eo*
Road Administrator	*admin*	*admin*

Table 15 - Access Usernames and Passwords

To be able to login to the system, please point your browser to *http://yancho.no-ip.org/~yancho* and use any of the above login details to access the system. Once you are logged in you will be redirected to the particular landing page of each.

II.III.II Using the System

In this section different screenshots from the various sections of the web interface will be discussed and explained. Please remember that this system was only tested on a Mozilla Engine Web Browser. Thus some sections might not display good in other browsers, or some functionality might be decreased.

Vehicle Owner / Driver Functions

The three main functions you have available are to register a new vehicle, list and modify the vehicles, and also to plan the route. Figure 42 shows the screen you will be presented if you choose the register new vehicle. This screen is a self explanatory one, just remember that all the data needs to be inputted. The TypeID and the Engine Capacity are of utmost importance since it is such values that the routing algorithm will use.

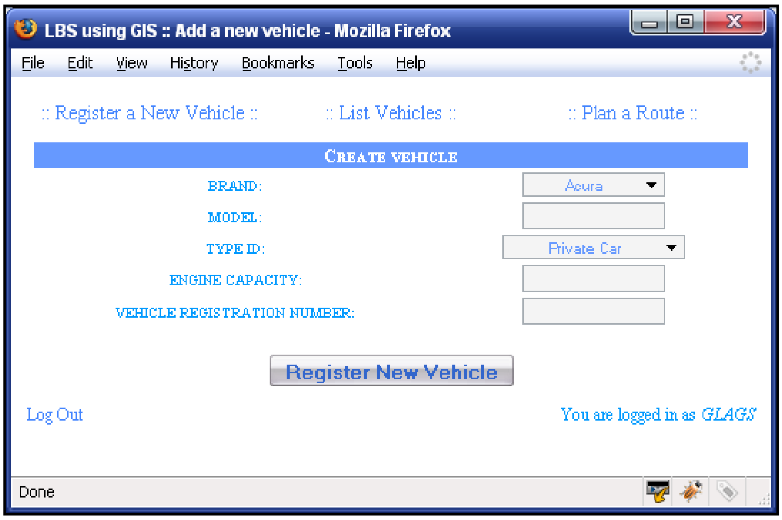


Figure 42 : Register New Vehicle

The newly added vehicle, will be set to the default vehicle used after addition. If you want to modify the default vehicle, or delete a vehicle not used any more you can do so from the List Vehicles section of the software as showed in Figure 43 below.

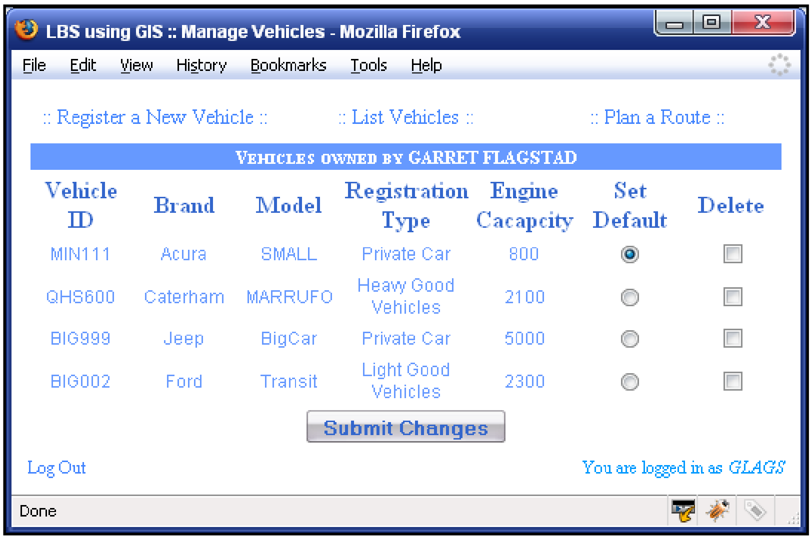


Figure 43 : Listing the Vehicles

The last section which the Vehicle Owner can use is the Plan a Route interface. In this section the route can be queried making use of the three different types of queries as can be seen in Figure 44.

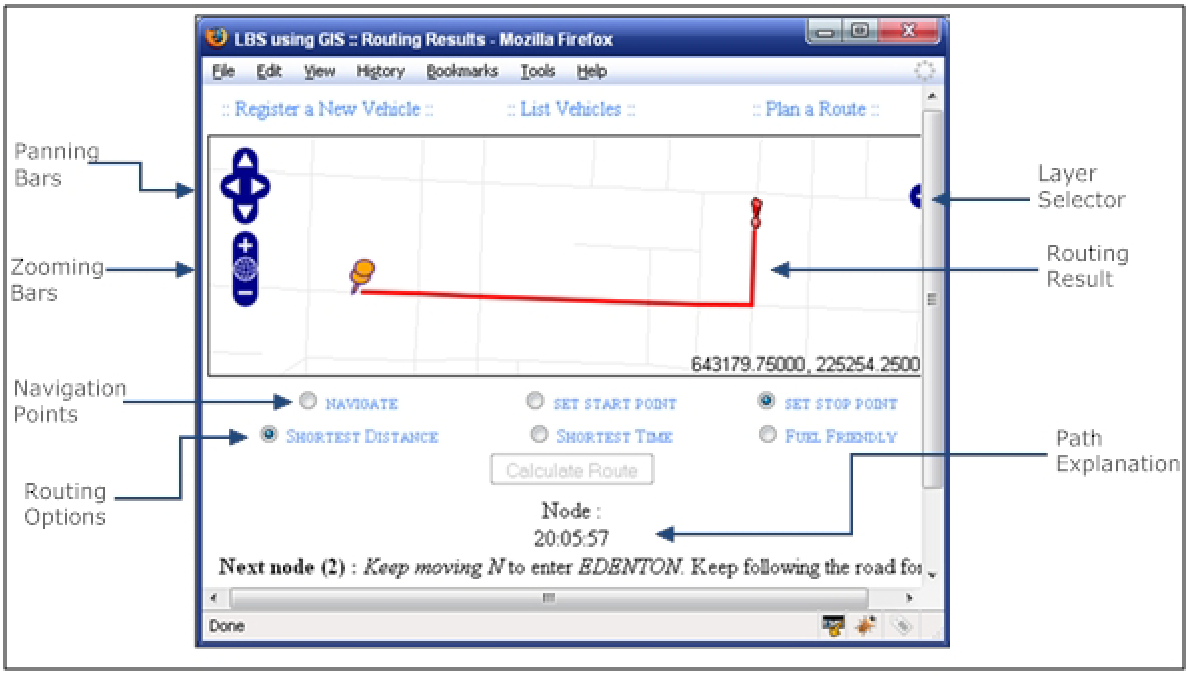


Figure 44 : Route Display

Also such an interface returns the list of routes the driver need to pursuit to reach his route. This list can also be printed, by using the Print function in your browser, in a special format prepared by the application as can be seen in Figure 45. In order to query for a route, you can follow this procedure:

- Locate the area where you need to do the routing. You can use the Panning and Zooming Bars to achieve this.
- Once the area is found, choose the '*Set Start Point*' and '*Stop*' point options to display where you want to have the route.
- Decide which option of Routing you would like, you can choose from Shortest Distance, Time or most Fuel Friendly
- Press '*Calculate Route'*
- When the route is computed, the Path Explanation starts, and will explain step by step the route. When it is finished, the '*Calculate Route'* button will be enabled again to allow you to perform another routing.

- You can press Print on your browser's window to print the result, see Figure 45 for the output.

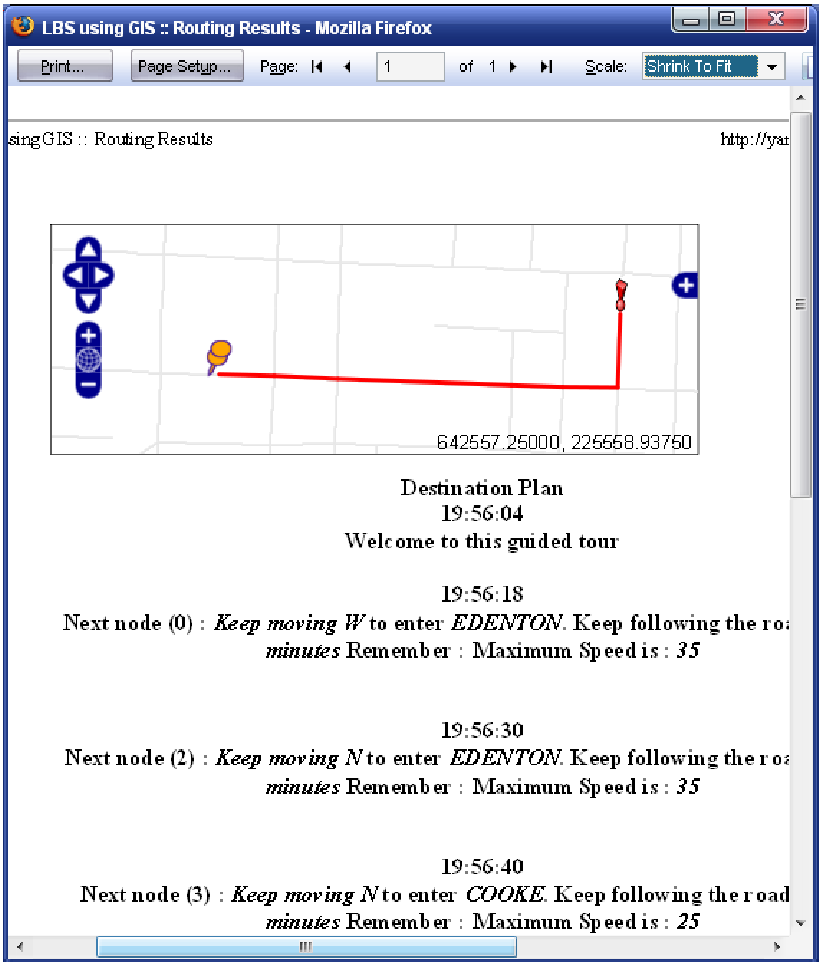


Figure 45 : Printing the Route

This section concludes the interface viewed and useful for the Vehicle Driver. In the following section the police officer interface is going to be studied.

Police Officer

This interface is useful for the officer which needs to report an accident. After logging in, the officer will just have the map zoomed on his location, marked with a red point, where he can set the accident location around. Figure 46 shows the interface presented with, after a successful login. As you can see, the officer can either choose from asking for a fire engine or an ambulance, and can also ban the road for access for normal vehicles until the peril is cleared from the road. It is very important not to close the application whilst it is computing a route since the road segment is banned only at the end of processing. This will enable the emergency response teams to view the route their vehicle needs to pursue straight after the call was inputted.

Figure 46 : Police Officer Interface

This concludes the interface used by the Police Officer. In the next section the interface used by the Road Administrator will be discussed.

Road Administrator

The Road Administrator has two options whilst logged in. The first one is to disable a new road segment, and the second one is to List and Manage the Road. The Disable Road interface looks like Figure 47. As you can see in the below figure, you can search for a particular road segment using various parameters. In this example we are searching by Street Name.

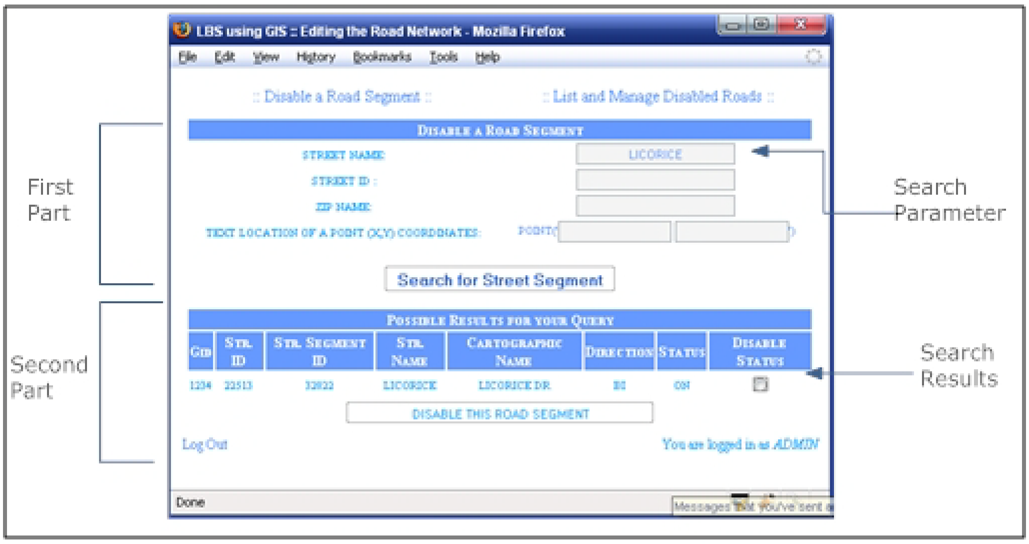


Figure 47 : Disable Road Segment Interface

Once the result of the search is computed, it is displayed and allows the Admin to disable the road. To do so you need to choose the according radio button and then press the '*Disable this Road Segment*' button. This will lead you to the next screen (Figure 48), where you can verify the details, decide on how long to ban the Road Segment and finally provide a reason why to ban the said road segment.

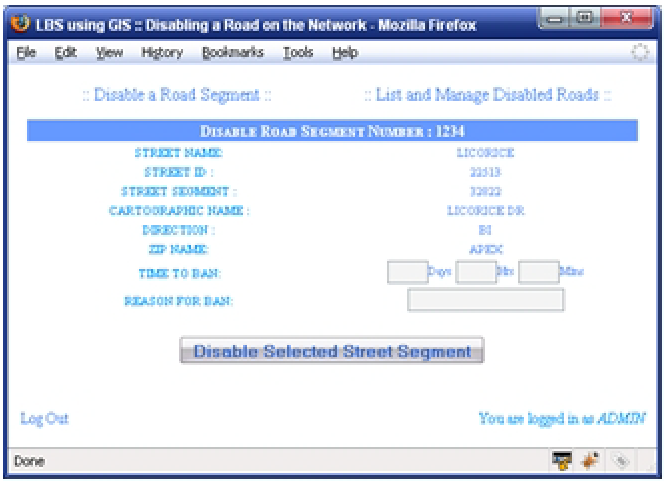


Figure 48 : Banning a Road Interface

The List and Managed Disabled Roads will provide a list of all the banned roads for the attention of the Road Administrator. As one can verify in Figure 49, the road is banned.

Figure 49 : List of Banned Road Segments

This interface allows you manage the banned road segments. You can either Remove completely the ban, Extend or Curtail the Ban. In order to do this, choose the desired road segment, scroll down the list of the banned road segments, and choose the desired option from the Radio Buttons. In this example we are extending the ban by another 1 hour. This can be seen in Figure 50. A confirmation dialog will follow.

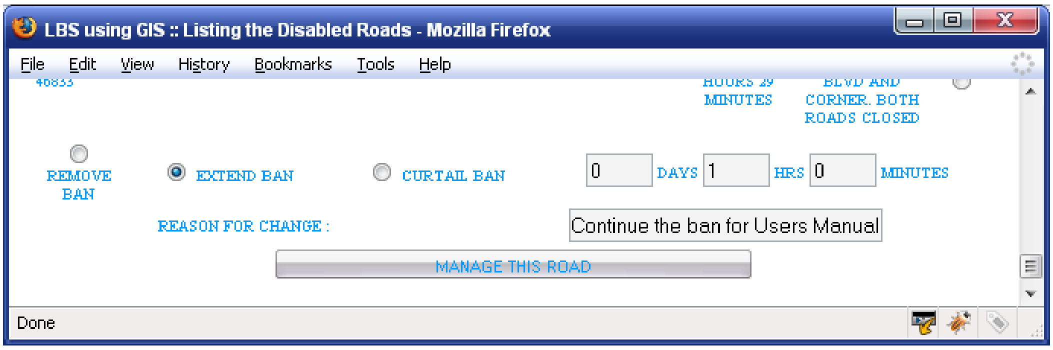


Figure 50 : Managing a Banned Road Segment

This concludes the explanation about how to use the interface, for the different types of users. As one can note from above the different users have different interfaces but each keep to the same interface, which can be loaded on most Web Browsers which support SVGand Javascript. Most latest browsers support by default in a way or another.

II.IV Conclusion

This manual explained the different parts of the software, from the early stages to installing it to how to use the software. If you still require any further help you can reach the developer on http://yancho.no-ip.org:8000/thesis and open a ticket there. Remember you need to register in order to open a ticket. Bugs will be kept track of there and any fixes will be reported on this website.

References

References

ABI Research. (2008, April). Asia Navigation Device Shipments to Exceed 185 Million in 2012; Connectivity and Integration Will Be Key. (N. Fabris, Compiler) Singapore: LBS360.NET.

ABI Research. (2008, April). United States Leads in Location Aware Services Deployments Today, but the Future Belongs to Europe and Asia. (N. Fabris, Compiler) London, United Kingdom: LBS360.NET.

Abrahamson, P. (2003, July 16). Open Source Software in the Geographic and Location Industries: A Case Study. *Directions Magazine* .

Adams, P. M., Ashwell, G. W., & Baxter, R. (2003). Location-Based Services — An Overview of the Standards. *BT Technology Journal , 21* (1), 34-43.

Altman, S. (2008). Best Practices for Location-based Services: Privacy, User Control, Carrier Relations, Advertising & More. *where 2.0.* Burlingame, CA: O'Reilly.

Alves, L. L., & Davis, C. J. (2006). Interoperability through Web Services: Evaluating OGCStandards in Client Development for Spatial DataInfrastructures. *VIII Brazilian Symposium on Geoinformatics Proceedings.* geoinfo.info.

Ambrosino, G., Bielli, M., Carotenuto, P., Romanazzo, M., & Sassoli, P. (1999). A modelling framework for impact assessment of urban transport systems. (M. Papageorgiou, & A. Pouliezos, Eds.) *Transportation Research Part D , 3*, 989-993.

Anderson, J., & Bell, M. (1997). Travel time estimation in urban road networks. *IEEE Conference on Intelligent Transportation System, 1997. ITSC '97* (pp. 924-929). Manhattan: IEEE.

ArcNews Online. (2005, Summer). *Managing Spatial Data in ArcGIS 9.2*. Retrieved November 23, 2007, from ESRI GIS and Mapping Software: http://www.esri.com/news/arcnews/summer05articles/managing-spatial-data.html

Armstrong, R., Gannon, D., Geist, A., Keahey, K., Kohn, S., McInnes, L., et al. (1999). Toward a Common Component Architecture for High-Performance Scientific Computing. *Argonne National Laboratory Mathematics and Computer Science Division (Proceedings of the 1999 Conference on High Performance Distributed Computing)* .

Beaudette, D. (2006, August). *UC Davis Soil Resource Laboratory.* (PostGIS, Ed.) Retrieved January 10, 2008, from PostGIS Case Study: http://postgis.refractions.net/documentation/casestudies/casoil/

Berg Insight AB. (2008, April). *More than 100 million users of mobile LBS in Europe by 2012 says Berg Insight* . (J. Fagerberg, Compiler) Gothenburg, Sweden: LBS360.NET.

Bernhardsen, T. (2002). *Geographic Information Systems: An Introduction.* Canada: John Wiley and Sons.

Berry, D., Usmani, A., Torero, J., Tate, A., McLaughlin, S., Potter, S., et al. (2005). FireGrid: Integrated emergency response and fire safety engineering for the future built environment. *Workshop on Ubiquitous Computing and e-Research.* Ediburgh: National eScience Centre, Edinburgh.

Bierlaire, M., & Frejinger, E. (2008). Route choice modeling with network-free data. (M. Papageorgiou, Ed.) *Transportation Research Part C: Emerging Technologies , 16* (2), 187-198.

Bonaccorsi, A., & Rossi, C. (2005). Comparing motivations of individual programmers and firms to take part in the Open Source movement. From community to business. *SSRN* (PDF). Pisa, Italy.

Borges, K. A., Laender, A. H., & Davis, C. A. (1999). Spatial data integrity constraints in object oriented geographic data modeling. *Proceedings of the 7th ACM international symposium on Advances in geographic information systems* (pp. 1-6). Kansas City: ACM.

Borgmann, A. (1984). *Technology and the Character of Contemporary Life: A Philosophical Enquiry.* Chicago: University of Chicago.

Brock Anderson, J. D. (2007, September 24-27). WMS Performance: Mapserver vs. Geoserver. *FOSS4G 2007* . Victoria, Canada.

Brooks, K. R. (2003). *GIS Raster/Vector Data Structures, Database Management Systems, Database Organization*. Retrieved December 3, 2007, from GIS Data Structures: http://www.spokane.wsu.edu/gis/Classes/LA467/gistrucx.htm

Brundell-Freij, K., & Ericsson, E. (2005). Influence of street characteristics, driver category and car performance on urban driving patterns. *Transportation Research Part D: Transport and Environment , 10* (3), 213-229.

Burnett, G., & Lee, K. (2005). The effect of vehicle navigation systems on the formation of cognitive maps. In G. Underwood, *Traffic and Transport Psychology: Theory and Application* (p. 656). Nottingham, UK: Elsevier.

Camara, G., & Onsrud, H. (2004, November 23). Open Source GIS Software : Myths and Realities. (N. I. Director for Earth Observation, Ed.)

Camilleri, J. (2007, December 03). Personal Communication. *e112 in Malta* . (D. (. Implementation), Ed.) Malta: Ministry for Justice and Home Affairs.

Canalys Navigation. (2008). *Almost 40% of smart phones shipping in EMEA have GPS integrated.* Reading: Canalys Navigation.

Capella, A., & Bennett, V. (2002). *Wherever you are, wherever you go, get the information you want to know.* Online, IBM.

Centre for Advanced Spatial Analysis. (2007). *GIS Timeline.* Retrieved November 26, 2007, from UCL Centre for Advanced Spatial Analysis: http://www.casa.ucl.ac.uk/gistimeline/

Chang, M.-H., Wu, C.-H., & Ho, J.-M. (2000). An Embedded Route Information Gathering Tool for Supporting ITS. *Institute of Information Science, Academia Sinica* . Taiwan.

Chen, Q., & Stauss, H. J. (1997). Evaluating traffic effects of a route guidance system by dynamic simulation. *Simulation Practice and Theory , 5* (7-8), 793-804.

Cheng, Q., Yang, C., Shao, Z., Liu, D., & Gao, L. (2004). Design and Implementation of WebGIS-based GPS Vehicle Monitoring System. (Z. Q. Wenjiang, Ed.) *Geo-Spatial Information Science , 7* (2), 96 - 100.

Chien, S. I.-J., & Kuchipudi, C. M. (2003). Dynamic Travel Time Prediction with Real-Time and Historic Data. (A. S. Engineers, Ed.) *Journal of Transportation Engineering , 129* (6), 608-616.

Cockcroft, S. (1997). A Taxonomy of Spatial Data Integrity Constraints. *GEOINFORMATICA -DORDRECT- , 1* (4), 327-344.

Conceição, H., Damas, L., Ferreira, M., & Barros, J. (2007). The DIVERT Project: Development of Inter-Vehicular Reliable Telematics. *Proceedings of FOSS4G2007. 3*, pp. 48-53. Porto: OSGeo Journal.

Coskun, M. Z., & Erden, T. (2006). Analyzing Shortest and Fastest Paths with GIS and Determining Algorithm Running Time. In *Visual Information and Information Systems* (Vol. 3736/2006, pp. 269-278). Springer Berlin / Heidelberg.

Cova, T. (1999). GIS in emergency managament. In P. Longley, .. M. Goodchild, D. Maguire, & Rhind, *Geographical Information Systems: Principles, Techniques, Applications, and Management* (pp. 845-858). SpringerLink.

Crossland, M. D., Wyne, B. E., & Perkins, W. C. (1995). Spatial decision support systems: an overview of technology and a test of efficacy. *Decision Support Systems , 14* (3), 219-235.

Databasedev. (2007). *Flat File Database Design vs. Relational Database Design:*. Retrieved August 03, 2008, from Database Solutions: http://www.databasedev.co.uk/flatfile-vs-rdbms.html

Davidsson, F., Matstoms, P., Lillienberg, S., & Andersson, H. (2002). *OPTIS statistical analysis and PROBE simulation study.* Centrum för trafiksimulering. Royal Institute of Technology.

Dechter, R., & Pearl, J. (1985). Generalized best-first search strategies and the optimality af A*. *Journal of the ACM (JACM) , 32* (3), 505 - 536.

Deguara, C. (2007, December 3). e112 in Malta. (P. Secretary, Ed.) Malta: Ministry for Justice and Human Affairs.

Department of Transport, UK. (2007). *Road Transport and the EU Emissions Trading Scheme.* London: Department of Transport, UK.

Dept. Trade and Industry. (2006). *Intelligent Infrstructure Futures Project: The Scenarios - Towards 2055.* IIS Project. Crown.

Derekenaris, G., Garofalakis, J., Makris, C., Prentzas, J., Sioutas, S., & Tsakalidis, A. (2001). Integrating GIS,GPS and GSM technologies for the efective management of ambulances. (J.-C. Thill, Ed.) *Computers ,Environment and Urban Systems , 25* (3), 267-278.

Detcher, R., & Perl, J. (1985). Generalized best-first search strategies and the optimality of A*. *Journal of the ACM , 32* (3), 505-536.

Devogele, T., Parent, C., & Spaccapietra, S. (1998). On spatial database integration. *International Journal of Geographical Information Science , 12* (4), 335-352.

Donlon, J. J., & Forbus, K. D. (1999). Using a Geographic Information System for Qualitative Spatial Reasoning about Trafficability. In Q. R. Group (Ed.), *Proceedings of QR99* (pp. 1-11). Loch Awe: Northwestern University.

Downey, L. (2006). Using Geographic Information Systems to Reconceptualize Spatial Relationships and Ecological Context. *American Journal of Sociology , 112* (2), 567-612.

Downs, A. (2004). *Still Stuck in Traffic: Coping with Peak-hour Traffic Congestion.* US: Brookings Institution.

Du, J., & Aultman-Hall, L. (2007). Increasing the accuracy of trip rate information from passive multi-day GPS travel datasets : Automatic trip end identification issues. *Transportation research. Part A, Policy and practice , 41* (3), 220-232.

Dueker, K., & Butler, A. (2000). A geographic information system framework for transportation data sharing. In J. C. Thill, *Geographic Information Systems in Transportation Research* (I ed., pp. 13-36). Tenessee: Pergamon.

Dutta, S. (2006). *Mobile Positioning Technologies.* Retrieved November 27, 2007, from Mobile Positioning & Location-Based Services: http://www.surujdutta.com/technology.htm

EAA. (2006). *Transport and environment: facing a dilemma (TERM 2005).* Bruxelles: EEA.

Edstrom, W. (2007, May 09). *Microsoft SQL Server "Katmai" Builds on Proven Success of SQL Server 2005, Empowering Customers to Manage the Data Explosion.* Retrieved November 30, 2007, from Directions Magazine: http://www.directionsmag.com/press.releases/index.php?duty=Show&id=17374&trv=1

EEA. (2003). *An evaluation of possible EU air transport emissions trading scheme allocation methods* . EU.

Elliot, R. (2006, October). *Infoterra.* Retrieved January 10, 2008, from PostGIS Case Studies: http://postgis.refractions.net/documentation/casestudies/infoterra/

Ellis, F. (2001, March 12). *Raster Representation of Data*. (A. P. Dr F. Escobar, Editor, & The University of Melbourne) Retrieved December 3, 2007, from Introduction to GIS: http://www.sli.unimelb.edu.au/gisweb/menu.html

Ericsson, E., Larsson, H., & Brundell-Freij, K. (2005).

Ericsson, E., Larsson, H., & Brundell-Freij, K. (2006). Optimizing route choice for lowest fuel consumption – Potential effects of a new driver support tool. *Transportation research. Part C, Emerging technologies , 14* (6), 369-383.

ESRI Support Centre. (2008, March 12). *Shapefile file extensions*. Retrieved September 12, 2008, from ArcGIS Desktop Help 9.2: http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?id=2729&pid=2727&topicname=Shapefile_file_extensions

ESRI. (2007). *What is GIS?* Retrieved September 2008, from gis.com: http://www.gis.com/whatisgis/index.html

ESRI(UK). (2002). *AA - Getting the Motorist Out of a Jam.* ESRI (UK).

EU. (2006). *Biofuels for transport provide a rosy future, according to draft report.* European Communities, Biofuels Research Advisory Council. Luxemburg: CORDIS.

EU Commission. (1999). *Towards a new framework for Electronic Communications infrastructure and associated services.* Retrieved June 23, 2006, from The 1999 Communications Review: http://ec.europa.eu/comm/information_society/policy/telecom/review99/pdf/rev

EU-15. (2006). *Greenhouse gas emission trends and projections in Europe 2006.* EU Commission. Bruxelles: EEA (European Environment Agency); OPOCE (Office for Official Publications of the European Communities).

European Union Road Federation. (2007). *The ERF's contribution towards enhanced Transport statistics.* Bruxxelles: European Union Road Federation (ERF), the Brussels Programme Centre of International Road Federation (IRF).

FCC. (2007, August 16). *Enhanced 911 - Wireless Services*. Retrieved November 25, 2007, from Federal Communication Commision: http://www.fcc.gov/pshs/911/enhanced911/Welcome.html

Federal Communications Commision. (1999). *9-1-1 Services.* Retrieved June 04, 2008, from Federal Communications Commision: http://www.fcc.gov/pshs/services/911-services/

Fetter, D. (2007, October 12). When to use an Index? (M. Pulis, Interviewer)

Fiedler, R. S., Schuurman, N., Grzybowski, S. C., & Grund, D. (2006). Defining rational hospital catchments for non-urban areas based on travel-time. *International Journal of Health Geographics , 5*, 43.

Fishburn, P., & Taaffe, K. (1994). Simulation optimization methodologies. *Proceedings of the 26th conference on Winter simulation;. 1*, pp. 22-25. Orlando: Society for Computer Simulation International.

Folmer, E. (2005, February 28). *Dutch use open source for Geoservices in public works.* (D. Duijnmayer, Ed.) Retrieved January 08, 2008, from iDADBC: http://ec.europa.eu/idabc/servlets/Doc?id=19454

Fontana, J. (2007, October 29). *No price hike for SQL Server 2008, Microsoft says*. Retrieved November 30, 2007, from Network World: http://www.networkworld.com/news/2007/092007-no-price-hike-sql-server.html

Furieri, A. (2008, April). *Quick Intro.* Retrieved May 1, 2008, from SpatiaLite - VirtualShape: http://www.gaia-gis.it/spatialite/

Gates, G., Burr, J., & Simmons, N. (2002). Commercial applications arising from a floating vehicle data system. *ITS UK Summer Conference* (pp. 1-8). Birmingham: 9th World Congress On Intelligent Transport Systems.

Gera, M. V. (2008, February 11). Info about Weather. Malta.

GISdevelopment. (2000). *Milestones of GIS.* Retrieved November 21, 2007, from The GeoSpatial Resource Portal: http://www.gisdevelopment.net/history/index.htm

Goodchild, M. F. (1992). *Geographical data modeling. Computers and Geosciences* (Vol. 18). Tarrytown, New York, USA: Pergamon Press Inc. .

Goodchild, M. F., & Noronha, V. (2000). Map accuracy and location expression in transportation - reality and prospects. (V. I. Laboratory, Ed.) *Transportation Research Part C: Emerging Technologies , 8* (6), 53-69.

GRASS Development Team. (2008, September 15). Geographic Resources Analysis Support System (GRASS) Software. (6.3). Trento, Italy. Retrieved from GRASS GIS: http://grass.osgeo.org

Greenroad. (2007, March). Is Safe Driving More Economical? Driver Safety and Fuel Consumption. (C. Bannis, Ed.) *Green Manouvers Newsletter* .

Gupta, P., Jain, N., Sikdar, P., & Kumar, K. (2003). Geographical Information System in Transportation Planning. *Map Asia 2003.* New Delhi: GISDevelopment.Net.

Hafberg, G. (1995). Integration of geographic information systems and navigation systems for moving (dynamic) objects like vehicles and ships. *Proceedings of the ESRI User Conference.* ESRI.

Hagen, W. v. (2007). Why Choose Ubuntu? In W. v. Hagen, *Ubuntu® Linux® Bible* (p. 936). John Wiley and sons.

Hang, S. (2006). *UMTS Bradmand Mobile Technology Rolled Out.* Frankfurt: Deutsche Bank Research.

Hawkins, R., & Stopher, P. (2004). Collecting data with GPS: those who reject, and those who receive. *Proceedings of 27th Australian Transport Research Forum.* Adelaide: TSC.

Hayashi, Y., & Morisugi, H. (2000). International comparison of background concept and methodology of transportation project appraisal. *Transport Policy , 7* (1), 73-88.

Hayes, B. (1998). Computing Science: Collective Wisdom. *American Scientist , 3*.

Haynes, R., Jones, A., Sauerzapf, V., & Zhao, H. (2006). Validation of travel times to hospital estimated by GIS. *International Journal of Health Geographics , 5*, 40.

Hiltbrand, R. K. (2006). *A Comparison of Proprietary and Open Source Software as a Way to Maintain Availability in an Open Access Environment.* University of Houston: The Faculty of the Department of Information & Logistics Technology.

HiVoltage. (2008, June 5). *Google Maps acknowledges Malta* . Retrieved June 30, 2008, from Maltameter.com: http://maltameter.com/index.php/option/content/task/view/id/97

Horanont, T., Tripathi, D. K., & Raghavan, D. (2002, November). A Comparative Assessment of Internet GIS Server Systems. (D. Santitamnont, Ed.) *GIS Development* .

Huang, H.-J., & Lam, W. H. (2002). Modeling and solving the dynamic user equilibrium route and departure time choice problem in network with queues. *Transportation Research Part B: Methodological , 36* (3), 253-273.

Hunt, D. (2008, 02 27). Streets Data Layer. (M. Pulis, Interviewer) Wake County.

Hunt, D. (2008, 02 27). Streets Data Layer. (M. Pulis, Interviewer) Wake County.

IBM. (2007). *IBM Spatial Offerings*. Retrieved November 29, 2007, from IBM: http://www-306.ibm.com/software/data/spatial/

IBM. (n.d.). *Informix Spatial DataBlade Module*. Retrieved November 27, 2007, from IBM: http://www-306.ibm.com/software/data/informix/blades/spatial/

Iguchi, M. (2002). A perspective on ITS deployment. *JSAE Review , 23* (2), 173-176.

Infoterra Ltd. (2008). *Spatial Solutions for Infrastructure Planning.* Leicester: Infoterra Ltd.

Ipa. (2006). Evaluation of PostgreSQL Using DBT-1. *The 4th Northeast Asia OSS Promotion Forum (Tianjin)* (p. 13). Tianjin: Ipa.

Ishak, S., & Al-Deek, H. (2002). Performance Evaluation of Short-Term Time-Series Traffic Prediction Model. (R. V.-U. American Society of Civil Engineers, Ed.) *Journal of Transportation Engineerin , 128* (6), 490-498.

Ivo. (2007). *101 ways to improve your gas-mileage.* Retrieved August 2008, from http://home.wanadoo.nl/kroone/gas/gas.html

Jackson, J., Callahan, D., Marstander, J., & Wang, P. (2007). A method for relative position tracking of vehicles using optical navigation technology. In S. Point (Ed.), *Proceedings of the 6th Conference on 6th WSEAS International Conference on Signal Processing - Volume 6. 6*, pp. 163-166. Texas: World Scientific and Engineering Academy and Society (WSEAS).

Jeff. (2007, March 27). *Making use of Assisted GPS (A-GPS) in cell phones.* Retrieved August 16, 2008, from The Warp: http://thewarp.org/blog/index.cfm/2007/3/27/Making-use-of-Assisted-GPS-AGPS-in-cell-phones

Johansson, H. (1999). *Effekter av EcoDriving pa avgasutslapp och bransleforbrukning, enforstudie.* (L. &.-F. Ericsson, Trans.) Borlange: Vagverket.

Jones, C. (2006, August 29). EU to get tough on car makers over emissions. (B. Johnson, Ed.) *The Parliament Magazine* (266).

JupiterOnlineMedia. (2008). *Dynamic Systems Development Method*. Retrieved May 23, 2008, from DevX.com : The know-how behind Application Development: http://www.devx.com/architect/Article/32836/0/page/3

Kardos, J., Benwell, G., & Moore, A. (2007). Assessing different approaches to visualise spatial and attribute uncertainty in socioeconomic data using the hexagonal or rhombus (HoR) trustree. *Computers, Environment and Urban Systems , 31* (1), 91-106.

Karl, C., & Béchervaise, N. (2003). The Driver as an Active Learner: Customising Real Time Traveller Information. In N. E (Ed.), *10th International Conference on Travel Behaviour Research*, *19.* Lucerne.

Karl, C., Charles, S., & Trayford, R. (1999). Delivery of Real-Time and Predictive Travel Time Information: Experiences from a Melbourne Trial. *Proceedings of 6th World Congress on Intelligent Transport Systems.* Toronto.

Katibah, E. (2007, May 30). Microsoft Shares Details on SQL Server 2008 Spatial Support. (D. Staff, Interviewer) Directions Magazine.

Kemp, & Goodchild. (1999). *The History of GIS.* Retrieved November 23, 2007, from What is GIS?: http://mama.indstate.edu/users/gejdg/447wk2.pdf

Khokhar, S., & Nilsson, A. (2006, August 30). Mobility Profiles and Future Location-Based Services. North Carolina, USA: Directions Magazine.

Knox, D. (2007, January 29). *Mobile Advertising.* Retrieved November 20, 2007, from European Communications: http://www.eurocomms.com/features/111562/Mobile_advertising.html

Kovel, A. E. (2000). Using GIS in Emergency Management Operations. *Journal of Urban Planning and Development , 126* (3), 136-149.

LaMance, J., Jarvinen, J., & DeSalas, J. (2002, March 1). *Assisted GPS: A Low-Infrastructure Approach.* Retrieved November 12, 2007, from GPS World: http://www.gpsworld.com/gpsworld/article/articleDetail.jsp?id=12287

Laurini, R. (2000). A Short Introduction to TeleGeoProcessing and TeleGeoMonitoring. *Second International Symposium on TeleGeoProcessing (TeleGeo 2000)*, (pp. 1-12). Nice.

Lee, J., & Lee, D. W. (2006, July - Sept). Going Open in WebGIS. *GIS Malaysia , 1* (3).

Leite, F. S., & Pereira, J. M. (2006). *Location-based services and emergency communications in IMT-2000.* Retrieved November 25, 2007, from MOBILITY – The shape of things to come: http://www.itu.int/itunews/issue/2001/07/mobility.html

Leshed, G., Velden, T., Rieger, O., Kot, B., & Sengers, P. (2008). In-Car GPS Navigation: Engagement In-Car GPS Navigation: Engagement. *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (pp. 1675-1684). Florence: ACM.

Li, J., & Kawakami, S. (2000). Predictive user equilibrium in dynamic queuing networks. *Proceedings of 2nd Asian-Pacific conference & exhibition on transportation and the environment* (pp. 1271-1278). Beijing: People's communications publishing house.

Li, J., & Liu, C. (2004). A proposed hybrid evaluation methodology for intelligent transport system deployment. *Control Conference, 2004. 5th Asian. 2*, pp. 819- 827. Guangzhou: Center for Intelligent Transp. Syst. Studies, Zhongshan Univ.

Li, Y., & McDonald, M. (2002). Link Travel Time Estimation Using Single GPS Equipped Probe Vehicle. *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on* (pp. 932-937). Southampton: Dept. of Civil & Environ. Eng., Southampton Univ., UK;.

Lime, S. (2008, April 12). *Welcome to MapServer.* Retrieved September 20, 2008, from MapServer: http://mapserver.gis.umn.edu/

Lin, F. T. (2002, March 1). An Experience Based Routing Method. *7th Inter-University Seminar on* . University of the Philippines.

Lin, H., Zito, R., & Taylor, M. (2005). A review of travel time predictions in transport and Logistics. *Proceedings of the Eastern Asia Society for Transportation Studies*, *5*, pp. 1433-1448. Adelaide.

Lloyd, B., Queen, P., & R., C. (1993). *The Basics of Geographic Information Systems.* Retrieved October 13, 2007, from University of Minnesota Extension: http://www.extension.umn.edu/distribution/naturalresources/DD5926.html

Ma, L., Priemus, P. d., & Bovy, P. d. (1999, December). Integrated environmental impact modelling for urban reight transport. *MSc. Thesis* . (T. R. Schoo, Ed.) Delft: lDelft University of Technology.

Macke, S. (2007, October 26). *Project: Topology Support for PostGIS*. Retrieved December 04, 2007, from cofundos - community innovation & funding: http://cofundos.org/project.php?id=46

Mainguenaud, M. (2000). A query rosolution engine to handle a path operator with multiple paths. *Transportation Research Part C: Emerging Technologies , 8* (1), 109-127.

MaltaStorms.com. (2008). *Latest News*. Retrieved May 21, 2008, from Accurate storm reporting in Malta: http://www.maltastorms.com/

Marchesin, E. (2003, June 23). *C#: A-Star is born.* (C. Maunder, Ed.) Retrieved May 07, 2008, from The Code Project: http://www.codeproject.com/KB/recipes/graphs_astar.aspx?fid=15953&df=90&mpp=25&noise=3&sort=Position&view=Quick&select=2387855&fr=26

Marcinkiewicz, P. (2007, November 11). Open Source GIS - Need Advice Please. (M. Pulis, Interviewer)

Marmar, S. (1999). *GIS in logistics and vehicle routing applications.* ESRI.

McDonald, K. (2002). The Modernization of GPS: Plans, New Capabilities and the Future Relationship to Galileo. *Journal of Global Positioning Systems , 1*, 1-17.

Meng, L., A. Z., & Reichenbacher, T. (2004). *Mobile Cartography: Adaptive Visualisation of Geographic Information on Mobile Devices* (1 ed.). Cambridge: Springer.

Mentor. (n.d.). *Norm's GIS Glossary*. Retrieved November 28, 2007, from Mentor Software: http://www.mentorsoftwareinc.com/resource/glossary.htm

Merrall, G. (2005). *PHP/MySQL Tutorial.* Webmonkey.com.

Michalak, S., Facelli, J., & Drew, C. (1999). Decentralized Information Technology Requires Central Coordination! *CAUSE/EFFECT , 22*.

Miller, H. a. (1996). Geographic information system design for network equilibrium-based travel demand models. *Transportation research. Part C, Emerging technologies , 4* (6), 373-389.

Mitasova, H. (2008, September 26). NC Streets Class Type. (M. Pulis, Interviewer)

Mitrović-Minić, S. (1998). *Pickup and delivery problem with time windows: A survey.* SFU CMPT TR.

Mitrović-Minić, S., & Laporte, G. (2004). Solving a Dynamic and Stochastic Vehicle Routing Problem with a Sample Scenario Hedging Heuristic. *Transportation Research Part B: Methodological , 38* (7), 635-655.

MobileIn. (2004). *Location-based Services.* Retrieved September 2008, from Mobile In a Minute: http://www.mobilein.com/location_based_services.htm

MySQL Forge. (2007, November 26). *GIS Functions*. Retrieved November 28, 2007, from GIS Functions - MySQLForge Wiki: http://forge.mysql.com/wiki/GIS_Functions

Nardi, B. A. (1996). *Context and consciousness: Activity theory and human-computer interaction.* New York: The MIT Press.

Nasr, M. R. (2007, January). Open Source Software: The use of open source GIS software and its impact on organisations. Middlesex, Middlesex University, United Kingdom.

Neis, P., & Dietze, L. (2008). *Introduction to OpenRouteService.org.* Retrieved September 22, 2008, from OpenRouteService.org: http://openrouteservice.org/

Neteler, M. (2007, December 27). *[GRASS-user] Trying to export a 3d Layer*. Retrieved from GRASS-user Mailing List: http://www.mail-archive.com/grass-user@lists.osgeo.org/msg00496.html

Neteler, M., & Mitasova, H. (2007, November 24). *Open Source GIS: A GRASS GIS Approach.* Retrieved November 28, 2007, from NC Original Data Set Sources, Metadata: http://www.grassbook.org/grasslocations/nc_spm_06_contents.html

Neufeld, K. (2007, October 26). *Status of Topology Support in PostGIS*. Retrieved December 04, 2007, from PostGIS-Users Mailing List: http://postgis.refractions.net/pipermail/postgis-users/2007-October/017532.html

Nielsen, O. A. (2004). Behavioural responses to road pricing schemes: Description of the Danish AKTA experiment. *Journal of Intelligent Transportation Systems , 8* (4), 233-251.

Noronha, V. (1999). Spatial Data Interoperability for ITS. *Presentation at Annual Meeting of the Transportation Research Board.* Washington.

Ochieng, W., & Sauer, K. (2002). Urban road transport navigation: performance of the global positioning system after selective availability. *Transportation Research Part C: Emerging Technologies , 10* (3), 171-187.

Ochieng, W., Polak, J., Noland, R., Zhao, L., Briggs, D., Gulliver, J., et al. (2002). The Development and Demonstration of a Real Time Vehicle Performance and Emissions Monitoring System. *PRESENTATION and PUBLICATION to the 82 Annual Meeting of the Transportation Research Board.* (p. 16). London: Imperial.

Oneyama, H., Oguchi, T., & Kuwahara, M. (2001). Estimation Model of Vehicle Emission Considering Variation of Running Speed. *Journal of the Eastern Asia Society for Transportation , 4* (5), 105-117.

OpenLayers. (2008). *OpenLayers: Home*. Retrieved 2008, from OpenLayers: http://openlayers.org/

Parentela, E., & Sathisan, S. (1995). *GIS-Based Allocation of Emergency Response Units along a Transportation Route.* Associates Professors of Civil Engineering. Las Vegas: UNLV Transportation Research Center.

Pearl, J. (1984). *Heuristics: Intelligent Search Strategies for Computer Problem Solving.* New York: Addison-Wesley Pub.

Penninga, F. (2004). *Oracle 10g Topology.* GISt Reporting.

Piras, A., Demontis, R., Vita, E. D., & Sanna, S. (2004). *Compact GML: merging mobile computing and mobile cartography.* Puglia: Research and Development in Sardinia.

Price, M. (2001, June). *Datums - Who Needs 'Em Anyway?* Retrieved November 10, 2007, from ESRI Gis and Mapping Software.

Pulis, M., & Attard, M. (2008). Exploring the Shortest Route Options: Applying Environmental Indicators to Calculating Shortest Route. *WICT 08.* Malta: in publication.

QGis. (2008). *Quantum GIS Documentation*. Retrieved September 27, 2008, from QGis.org: http://qgis.org/index.php?option=com_content&task=view&id=106&Itemid=79

Ramakrishna, Y., Ramakrishna, P., Lakshmanan, V., & Sivanandan, R. (2006). Bus Travel Time Prediction Using GPS Data. *Map India* .

Ramakrishnan, R., & Gehrke, J. (2003). *Database Management Systems.* McGraw-Hill Professional.

Ramser, G. B., & Dantzig, J. H. (1959). The Truck Dispatching Problem. *Management Science , 6* (1), 80-91.

Ramsey, P. (2005, October 07). *Open Source Company: Oxymoron?* Retrieved January 10, 2008, from Open source geospatial opinions and techniques, seen from the trenches and from far far above...: http://geotips.blogspot.com/2005/10/open-source-company-oxymoron.html

Ramsey, P. (2007). The State of Open Source GIS. *FOSS4G 2007* (p. 49). Victoria: Refractions Research.

Raymond, E. S. (1999, June 9). *The Magic Cauldron.* Retrieved January 10, 2008, from http://www.catb.org/~esr/writings/magic-cauldron/magic-cauldron.html

Refractions Research. (2007, August 13). *PostGIS*. Retrieved November 28, 2007, from What is PostGIS?: http://postgis.refractions.net/

Research and Markets. (2006, September). *Strategic Analysis of the European Mobile LBS Market.* Retrieved November 12, 2007, from Research and Markets: http://www.researchandmarkets.com/reportinfo.asp?report_id=350479&t=e&cat_id=

Richards, M., Ghanem, M., Osmond, M., Guo, Y., & Hassard, J. (2006, March). Grid-based analysis of air pollution data. *Ecological Modelling* , 274-286.

Rondin, E., Hailes, S., & Li, L. (2008). Load Sharing and Bandwidth Control in Mobile P2P Wireless Sensor Networks. *2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications* , 468-473.

Rosqvist, S. (2003). On the relations between driving patterns,exhaust emissions and network characteristics in urban driving. *Doctoral Thesis* . (D. o. Society, Ed.) Lund, Sweden: Lund Institue of Techonology, University of Lund.

Ryan, M. (2004, October 29). *Software Workshop Java Dijkstra's Algorithm.* Retrieved May 07, 2008, from The University of Birmingham: http://www.cs.bham.ac.uk/~mdr/teaching/modules04/java2/DijkstraAlgo.html

Saint-André, P. (2003, 04 29). *Open Source and Open Standards.* Retrieved July 15, 2008, from LAMP: The Open Source Web Platfrom: http://www.onlamp.com/pub/a/onlamp/2003/04/29/openstandardsopensource.html

Schiller, J., & Voissard, A. (2004). *Location-Based Services.* Morgan Kaufmann Title.

Schmidt, C. (2007, September). (M. Pulis, Interviewer)

Schmidt, C. (2007, September 27). *OpenLayers Routing, with a little help from my friends*. Retrieved October 15, 2007, from Technical Ramblings of a GIS Hacker: http://crschmidt.net/blog/categories/pgrouting/

Schutzberg, A. (2006, October 26). *PostGIS Certified Compliant*. Retrieved November 28, 2007, from Open GeoSpatial Consortium: http://ogcuser.opengeospatial.org/node/57

Schuurman, N., Fiedler, R. S., Grzybowski, S. C., & Grund, D. (2006). Defining rational hospital catchments for non-urban areas based on travel-time. *International Journal of Health Geographics , 5*, 43.

Schwartz, J. (2003, April). *Open Source vs Open Standards.* Retrieved July 2008, from CNet News: Perspective: Open source versus open standards: http://news.cnet.com/2010-1071-995823.html?hhTest=1

Schwarz, C. R. (1989, December). *Relation of NAD83 to WGS84.* Retrieved November 26, 2007, from North American Datum of 1983: http://www.mentorsoftwareinc.com/resource/Nad83.htm

Sester, M., Anders, K., & Walker, V. (1998). Linking objects of different spatial data sets by integration and aggregation. (P. Bergougnoux, S. Shekhar, & A. U. Frank, Eds.) *GeoInformatica* (10707), 335-358.

Shukla, R., & Magon, A. (2006). LBS, the ingredients and the alternatives. *GIS Development: The Global Geospatial Magazine* .

SmartMotorist. (2008, June). *Tips for driving in rain.* Retrieved June 2008, from SmartMotorist.com: http://www.smartmotorist.com/driving-guideline/tips-for-driving-in-rain.html

Smit, R. (2006, March). An examination of Congestion in Road Traffic Emission Models and their application to Urban Road Networks. *Doctor of Philosophy Thesis* . (S. o. Planning, Ed.) Faculty of Environmental Studies, Griffith University.

Smith, G. (2008, September). *PostgreSQL Cache*. Retrieved Septeber 2008, from PostgreSQL Admin Mailing List: http://archives.postgresql.org/pgsql-admin/

Smith, M., Goodchild, M., & Longley, P. (2008). *Geospatial Analysis - a comprehensive guide* (2 ed.). Leicester: Matador.

Sonnen, D. (2003, April 15). *Oracle's Geospatial Capabilities for Enterprise Information Systems.* Retrieved November 23, 2007, from Directions Magazine - The Worldwide Source for Geospatial Technology: http://www.directionsmag.com/article.php?article_id=350&trv=1

Spang, B. (2007, August 09). Informix Spatial Data Technology: Update and Positioning. (J. Francica, Interviewer) Directions Magazine.

Spearin, D. (2002). *Innovations in GIS Emergency Response Planning.* Canada: Public Safety and Emergency, Canada.

Sperling, J., & Sharp, S. A. (1999). A Prototype Cooperative Effort to Enhance TIGER. *Journal of the Urban and Regional Information Systems Association , 2* (11), 35-42.

SQLite. (2008, March 03). *SQLite Features:.* Retrieved May 01, 2008, from About SQLite: http://www.sqlite.org/features.html

Srivastava, L. (2004). Japan's ubiquitous mobile information society. *info , 6* (4), 234 - 251.

Steignier, S., & Bocher, E. (2008, September 5). An overview of current free and Open Source desktop GIS Developments. *Int. J. of Geographical Information Science* , 24.

Steiniger, S., Neun, M., & Edwardes, A. (2006). Foundations of Location Based Services . Zurich, Switzerland.

Stephens, R., & Plew, R. (2002). Hour 4. The Normalization Process. In R. P. Ryan Stephens, *Sams Teach Yourself SQL in 24 Hours* (3 ed., p. 504). Sams.

Stevens-Guille, M. (2001, July). *Next Generation Routing Solutions.* Retrieved May 29, 2008, from Maptuit Corporation: http://corporate.maptuit.com/about/articles/july2001.php

Team DC. (2000). *Distributed Computing's Disadvantages.* Retrieved June 3, 2008, from dc central.

Texas Transportation Institute. (2005). *Urban traffic congestion costs the USA $63 billion per annum.* Retrieved from CityMayors Transport: http://www.citymayors.com/transport/congestion_usa.html

Theobald, D. M. (2001, April - June). Understanding Topology and Shapefiles. *ArcUser* (04).

Thill, J.-C. (2000). Geographic Information Systems for Transportation in Perspective. In J.-C. Thill, *Geographic Information Systems in Transportation Research* (I ed., pp. 3-13). Buffalo, New York, USA: Pergamon.

Thirumalaivasan, D., & Guruswamy, P. V. (2005). *Optimal route analysis using GIS.* Retrieved June 23, 2008, from http://www.gisdevelopment.net/application/Utility/transport/utilitytr0004a.htm

Thomas, D., & Taylor, E. (2006). Study Designs and Tests for Comparing Resource Use and Availability II. *Journal of Wildlife Management* , 324–336.

Thong, C., & Wong, W. (1997). Using GIS to design a traffic information database for urban transport planning. *Computers, Environment and Urban Systems , 21* (6), 425-443.

TileCache. (2008). *User Manual*. Retrieved 2008, from TileCache: http://tilecache.org/

T-Mobile. (2003, July). *erman T-Mobile launches product integrating location based services, GPRS, Bluetooth and GPS.* Retrieved August 02, 2008, from Geekzone.com: http://www.geekzone.co.nz/content.asp?contentid=1125

T-Mobile. (2008). *Location Based Services.* Retrieved August 02, 2008, from T-Mobile: http://www.t-mobile.co.uk/help-and-advice/advice-for-parents/location-based-services/

Tweedie, C. (2006, January 12). *Spatial Indexing*. Retrieved December 3, 2007, from Chris Gizmo: http://chris.narx.net/index.php?s=timings

van der Voort, M., Dougherty, M., & van Maarseveen, M. (2001). A prototype fuel-efficiency support tool. *Transportation Research Part C: Emerging Technologies , 9* (4), 279-296.

Ververidis, C., & Polyzos, G. C. (2002). *Mobile Marketing using a Location Based Service.* Department of Informatics, Mobile Multimedia Laboratory. Athens: Athens University of Economics and Business.

Vonderohe, A. P., Travis, L., Smith, R. L., & Tsai, V. (1993). *Adaptation of Geographic Information System for Transportation.* Washington, USA: National Academy Press : Transportation Research Board, National Research Council.

Vreeker, R., Nijkamp, P., & Ter Welle, C. (2002). Transportation Research Part D: Transport and Environment. *7* (1), 27-42.

W. P. Carey School of Business. (2007, October 10). To Pay or Not To Pay: The World of Office Suites Opens Up. (E. Farquhar, Ed.) *Knowledge@W. P. Carey , Managing Technology* (1483).

Werbach, K. (2002, October). Open Spectrum: The New Wireless Paradigm. *Spectrum Series Working , 6*.

Wiles, F. (2008, September). *Why isn't PostgreSQL using my index?* (RevSys) Retrieved September 2008, from Revolutional Systems: http://blog.revsys.com/2008/09/why-isnt-postgr.html

Winter, H., & Goasduff, L. (2008, February). Gartner Forecasts Worldwide Location-Based Services to Grow Nearly 170 Per Cent in 2008. Egham, United Kingdom: Gartner.

Wolf, J., Hallmark, S., Oliveira, M., Guenslar, R., & Sarasua, W. (1999). Accuracy issues with route choice data collection by using global positioning system. *Transportation research record* , 66-74.

Wootton, I. A., & Spainhour, L. K. (2007). Examining Deficiencies in Florida Pedestrian Crash Data. *Transportation Research Record , 2002 / 2007*, 31-38.

Zhou, L. (2003). *NAC Geographic Products Launches Universal Address System For 26 Countries.* NAC Geographic Products Inc. Toronto: GISCafe.