This is a post-review, pre-copyedit version of a paper included in the 10th World Congress on ITS, the definitive publisher version is available as part of the conference proceedings CD-ROM

The full citation for this paper is...

McCullough F, Jenkins, M, Hermans F, Haskins D & Waterson B (2003) “ProbeIT: Map Information Management Framework for a Hybrid Navigation System Using Externally Sourced Data”, in proceedings of the 10th World Congress on Intelligent Transport Systems and Services, Madrid.

PROBEIT – MAP INFORMATION MANAGEMENT FRAMEWORK FOR A HYBRID NAVIGATION SYSTEM USING EXTERNALLY SOURCED DATA

Francis McCullough – Jaguar Cars Ltd - fmccull2@jaguar.com
Marcus Jenkins – Navigation Technologies - mjenki@navtech.nl
Dr Filip Hermans – Atkins Transport Systems – filip.hermans@atkinsglobal.com
David Haskins – Kingston University - haskins@drum.u-net.com
Ben Waterson – University of Southampton - B.J.Waterson@soton.ac.uk

SUMMARY

Electronic maps are a key component of the latest in-vehicle navigation systems, but correctly attributed and working in conjunction with telematics technology they would allow for other applications, like hybrid navigation and advice on speed and other regulatory information.

This paper outlines the design and implementation of a prototype in-vehicle navigation system, working off 100% downloadable data developed as part of the ProbeIT project. The system uses a caching strategy in order to optimise the quality of service, bandwidth usage, and to support operation when wireless communication is unavailable. The software components and data structures developed are described and discussed together with the advantages of this approach in relation to other navigation and telematics solutions.

INTRODUCTION

The main function of in vehicle navigation systems is to support the automation of route planning and guidance activities, which were previously performed by drivers using paper based maps. Currently some navigation units, when combined with a variety of traffic information services, also provide the driver with relevant road network status information and can even suggest re-routing to avoid traffic jams. In the UK, real-time traffic information is commercially available from TrafficMaster and ITIS, and for example the information used by Toyota’s Traffic Avoidance product, to deliver integrated dynamic navigation functionality.

These are highly useful applications. However, electronic maps correctly attributed and working in conjunction with telematics technology would allow the realisation of a number of additional in-vehicle applications; including advice on speed and other regulatory information. To enable these applications a suitable data exchange and service provision framework is required. An important factor of the framework is how enforceable traffic regulation information is, which is authorised by Traffic Authorities, when integrated within the process. The development and demonstration of such a framework is the objective of ProbeIT.

ProbeIT is a ForesightVehicle project, which is 50% funded by the Highways Agency (HA) and supported by Essex County Council. The research is performed by a UK consortium, which includes Atkins, Jaguar Cars Ltd, Navigation Technologies, Kingston University and the University of Southampton. Besides funding, Essex County Council provides views on potential benefits of such a system and how traffic regulation data can be integrated. ProbeIT is a three year project which started in January 2001.

The vision of the project is to develop an end-to-end process for the sourcing and exchange of geo-referenced information between traffic management systems, a uniform data source and vehicles utilisation of the Travel Information Highway (TIH) and cellular communication. The objective of this process is to provide timely and accurate location-based information in the vehicle, such that Advanced Driver Assistance Systems (ADAS) are possible. The aspects of ADAS which are considered in the project are: dynamic navigation, traffic regulation advice and floating vehicle dynamic data.

The objective of this paper is to put the need for a data management framework into context, define in more detail the applications the framework could underpin, to identify the key requirements and to demonstrate how the in-vehicle application has been designed to overcome these issues. Issues regarding the server design and data acquisition are discussed in an additional paper entitled 'ProbeIT – Vehicle Server and Data Integration'.

CONTEXT

The need for the development of a framework for the exchange of geo-referenced information, between a centralised uniform data source and in-vehicle systems, arose from both vehicle manufacturers and highway authorities.

The vehicle manufacturers, representing the interests of individual users, identified through focus groups that drivers of vehicles are very interested in systems which improve their driver comfort and safety. On the other hand, highways authorities, representing societal interest, identified that tools to improve road safety and network management could be enhanced though the better exploitation of telematics.

Driver comfort relates to reducing the effort required in the driving task and facilitation on route decision making. It is believed that timely and accurate map information, integrated with real-time traffic and travel information, will enable higher degrees of automation and personalised advice in vehicle.

Although full automation, if implemented correctly, could clearly lead to improved safety, the initial benefits are expected to arise from better compliance with traffic regulations resulting from warning the driver of dangerous traffic or driving conditions. A system supported by a framework, which can deliver regulatory information to the driver in a timely and secure fashion, will clearly be very valuable.

Equally valuable is the potential to use the vehicles themselves as probes, in order to learn information about network conditions, as part of the data exchange mechanism. Improvements in network management can then be expected both for achieving a better understanding of the network status and also from understanding compliance by drivers to instructions, such as speed and lane restrictions, given as part of the network management strategies.

REQUIREMENTS

Although there are many stakeholders who could benefit from applications enabled by a ProbeIT framework, the effectiveness and success will depend on how the drivers interact with the ADAS applications. This means that all requirements for the ProbeIT framework and prototype should be driven by what the vehicle users expect from the system.

Vehicle System Requirements

Most journeys consist of a small distance on minor roads, before moving on to major roads for the bulk of the journey. The usage of a navigation system is biased toward journeys in which the user (driver) is unfamiliar. The integration of real-time traffic and regulatory information makes it much more likely that the driver will use the system on a daily basis even if he is familiar with route. There is significant evidence that drivers are not aware of the current speed limit when driving and in areas of high enforcement would benefit from knowing the speed limit.

ADAS can be defined as those vehicle applications which are aware of the environment that the vehicle operates in. There are two main ways of acquiring this awareness. The 'autonomous' approach, in which vehicles rely on on-board sensors, usually based on radar, laser and video technology, to detect objects and the layout of the road infrastructure. The alternative is a more 'cooperative' approach in which systems on board of the vehicle obtain information communicated from the roadside using telematics. The potential of this 'cooperative' approach is what the ProbeIT framework is trying to address.

Most of the existing cooperative systems on the market fall in to one of two categories, either a thick or thin client. Thick clients typical contain all the functionality and a have a large static data store, although this may be supplemented with dynamic data such as with Toyota's Traffic Avoidance System. The thin clients on the other hand rely heavily on a remote server and have very little functionality without communication with the server, as well as having little persistent data storage. Thick clients have lower communication costs, but the platforms are usually more expensive, and the reverse is true for thin clients.

Because each approach has advantages and disadvantages, the ProbeIT project tries to get the best of both worlds by implementing a hybrid solution, that would support most of the functionality of standard navigation solutions without wireless communications, but provides greater functionality when communications are available. This means the base requirement of the project is to implement a prototype hybrid navigation system using 100% downloadable data, whilst providing functionality and usability comparable to current CD and DVD-ROM based systems. The system design goes beyond the capability of these products so as to enable two-way exchange of dynamic information as well as providing advice based on the downloaded dynamic data synchronised with the route guidance instructions. Applications supported are traffic regulation and speed advice. All this functionality is implemented on a system with limited persistent memory and CPU performance. The delivery of downloaded data is targeted at 2.5G mobile phone data links using standard Internet protocols.

Data Requirements

Due to possible safety and legal implications connected to traffic regulation and speed advice, there is a requirement for the data to be accurate and timely. Because the data will reside between three different locations, i.e. publishers, service provider and individual vehicles, the key elements of the ProbeIT data exchange framework are the management processes and mechanisms. The management processes and mechanisms make sure that the data used in the vehicle, to inform the driver or to be integrated into the vehicle, is synchronised with the most up-to-date information available from the publisher. Therefore, the focus of the ProbeIT project is on the development of a framework and not the applications themselves.

Data delivered to the vehicle from the server is derived from a number of sources. Basic map data including road geometry, cartographic features and most static road attributes (including geocoding information) is sourced from Navigation Technologies. Dynamic data is obtained from three main sources: QMISS [1] data is acquired via the UK Travel Information Highway (TIH) [2] infrastructure, local traffic order information is obtained from various databases from Essex County Council and the floating car data mechanism.

Having a floating car data mechanism as an integral data source means that the vehicles themselves can be used as probes, in order to learn information about network conditions. Improvements in network management can then be expected both for achieving a better understanding of the network status and also from obtaining greater compliance by drivers to instructions (such as speed and lane restrictions, given as part of the network management strategies).

The need for multiple sources is required to enable the various ADAS applications to be available with the required functionality and geographical coverage. Depending on the source and type of the data, the format and quality will not necessarily be the same. In order for the application to be able to operate seamlessly between areas, all data sources need to be integrated to approved quality. This integration corresponds both to location referencing i.e. matching geo-referencing systems and data attributes.

Sourcing and Distribution Requirements

Cooperative ADAS applications can be provided through proprietary mechanisms. However, it is considered that a commonly agreed architecture is needed. This is because of the range of different stakeholders (which include central and local government authorities, potentially from different countries, and the vehicle and component manufacturers), and the increasing demand for global and interoperable solutions. There is also a great deal of fundamental communality between the characteristics of each system: wireless data link, geo-referenced data, need for data accuracy, timeliness and traceability, which all should be exploited. The realisation of these applications, using many common components, will also lead to economies of scale and so reduce the cost, and in turn increase the uptake rate and so offer greater potential for spin-off applications which have not yet been considered.

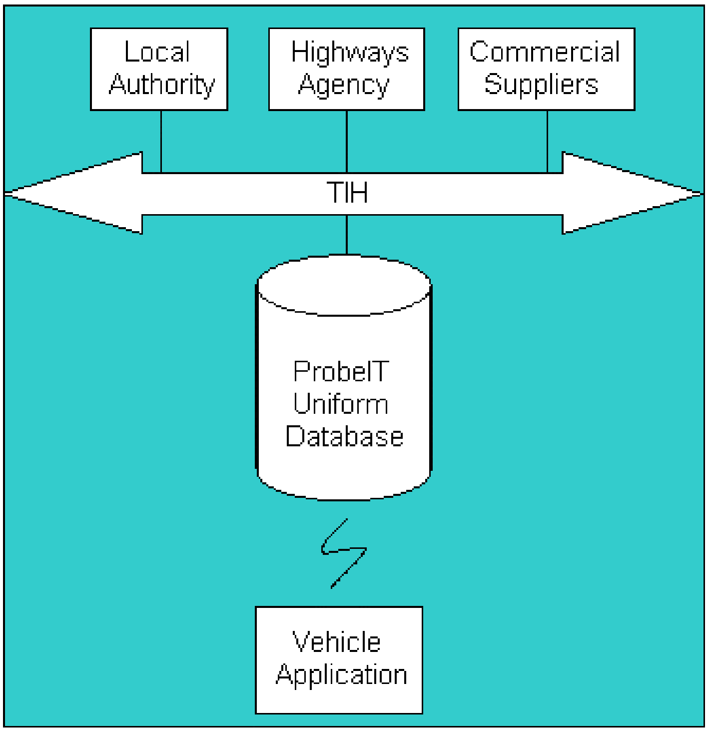

Figure 1: Top Level Architecture

The top level architecture of the project is shown in Figure 1. The external information is integrated into the ProbeIT database to establish a uniform data source, which is used to serve spatially referenced information to the vehicle. Serving information to the vehicle is done using a specially designed interface which serves, on demand, a set of spatial objects to the vehicle. These objects are managed by the vehicle application to support dynamic navigation and traffic regulation advice. The vehicle uses the same architecture to communicate information back to the server to enable advanced probe vehicle applications.

The data management mechanism relies on wireless communication between vehicle and server. At the outset of the ProbeIT project cellular communication was selected. The main reasons for this are that no special infrastructure is required, which reduces rollout costs and increases rollout speed, and enables communications being instigated by the application. Because the infrastructure is provided by a communication service provider, the exchange of each set of data will cost. This means that data exchange needs to be optimised and kept to a minimum.

The architecture is considered open because each function within the framework is abstracted behind a tightly defined set of interfaces. Because this has been done extensively throughout the system, modification to detailed implementations, new sources of information and new functionality of vehicle applications can be easily accommodated without system redesign.

VEHICLE ARCHITECTURE

Hybrid functionality

Based on the above stated requirements, the project team decided to try and get the best of both thin and thick clients by implementing a hybrid solution that would support most of the functionality of standard navigation solutions without wireless communications, but provide greater functionality when communication is available.

Standard navigation functionality, such as destination entry (geocoding), moving map display, vehicle positioning, route calculation and route guidance all need to be provided by this system to support the existing market expectation. These core functionalities, and the way in which they are implemented in this system, are described in more detail later in this paper.

The hybrid solution is driven by the fact that the capacity and quality of service which can be expected from the mobile telecommunication network is variable from region to region, and cannot be guaranteed, especially in rural areas. This impacts on the storage and management of the data on the vehicle. The caching of data becomes critical as it will dramatically improve the performance of the system, as well as minimise communication charges and latency in the user interface.

A minimum level of functionality without vehicle-server communication was decided upon and this defined that a set of data needed to be maintained in a cache database in the vehicle system. The functionality thought to be essential is that any town name can be entered and geocoded and the vehicle can calculate a route using high-level roads only. The detailed geocoding and route calculation is performed when stationary on route. To support the geocoding functionality, at minimum, all of the city/town/village name and location information are cached in the vehicle. To calculate the high-level road route, the complete high-level road network, in routable form, is also cached in the vehicle. High-level roads comprise motorways and trunk roads and form a relatively small part of the total volume of the navigation, and can be contained at insert volume. In order to provide guidance on high-level roads detailed geometry for the high-level road network is also cached in the vehicle. Caching is also essential for rapid rendering of large scale maps (such as showing a complete country on a screen) generalised map geometry and cartography. A smaller scale map comprising detail of the route on the high-level road network can also be rendered from the cached data.

The software architecture

The functional decomposition of a navigation system naturally highlights a number of functions, which form self-contained components. When dealing with complex software systems an object orientated approach offers significant advantages, allowing the very complex implementation to be encapsulated behind a single class. The use of CORBA [3] middleware further enhances the concept by allowing components written in different languages and running on different platforms to communicate and work together. The operating-system and programming language agnostic nature of CORBA also means that the design of the system would be platform independent. The discipline of defining the interfaces to the components in IDL formed a significant part of the design process and formed a contract between the parties writing the software components

Using CORBA also provides a good mechanism for experimenting with distributing some components on the server and some on the vehicle. The most likely components for such distributed processing are route calculation and geocoding. It is unlikely that sufficient time is available during this project to implement this, but it could facilitate future work in this direction.

Components design

The in-vehicle software is designed on a component and interface level such that subsystems can be packaged and deployed either within different systems in the vehicle or even in the server. In the prototype system, CORBA is used to implement this inter-subsystem communication. The component design is shown on the diagram in Figure 2.

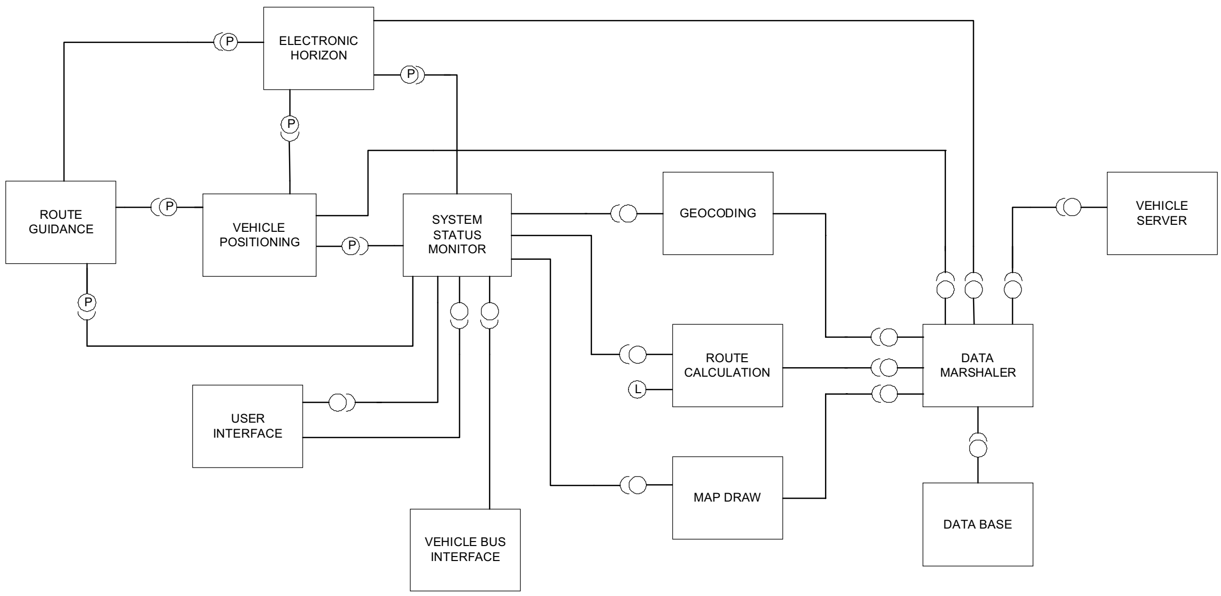

Figure 2: Vehicle Component Diagram

The system status monitor is the main control component. It contains a high quantity of application logic and controls the communication between the other components. The system status monitor also maintains the state of various parameters for the operation of the system, such as user preferences, current and recent routes and the current state of the HMI (Human-Machine Interface).

The vehicle-positioning component uses sensor data from GPS (Global Positioning System), yaw rate gyro and wheel speed to identify the most likely position on the map. This information is provided by Navigation Technologies' prototype sensor box (see Figure 3). GPS is the primary source of position and vector information from the vehicle. However, GPS has a variable accuracy dependent on visibility of satellites, multipath errors and whether Selective Availability is enabled by the US government. Therefore, inertial sensors augment the positioning against the map. The solid-state gyroscope provides reliable indication when sharp turns are made by the vehicle, although subtle turns, as made when exiting a motorway on a slip road, are not reliably detectable by this sensor. The wheel rotation sensor (‘speed pulse input’), once calibrated for a vehicle, provides reliable information about the distance travelled along a road. These three physical sensors are fused with the map geometry itself to provide software messages every second with the latest vehicle position. A commercially available CAN [4] device was used generate a square wave speed pulse from CAN data.

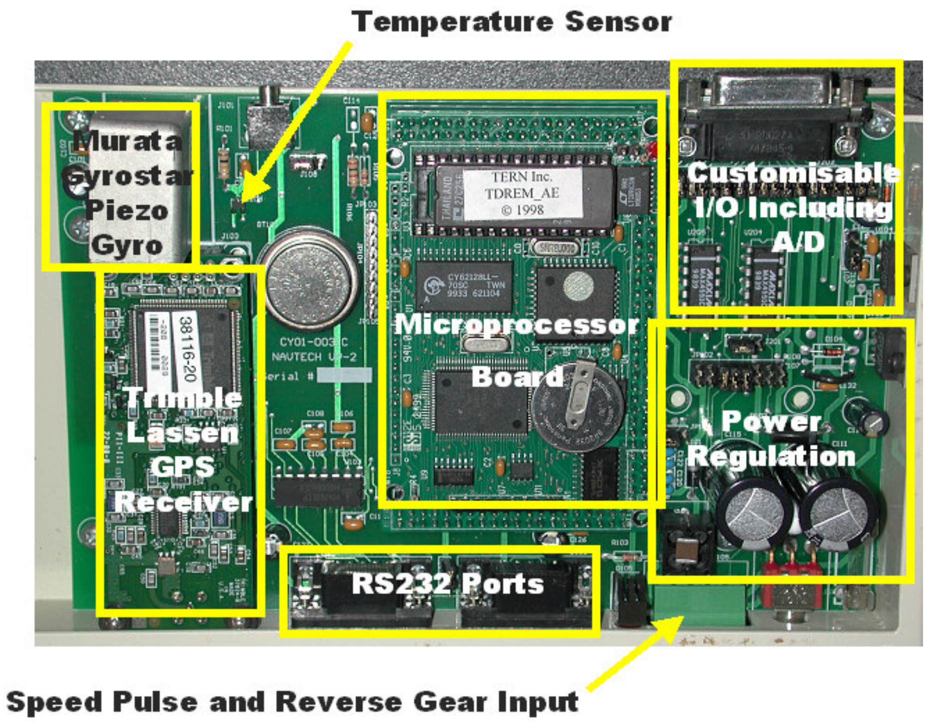

Figure 3: Sensor Box

The vehicle position module is used by a number of components in the vehicle and implemented using an observer pattern, to which the other components subscribe. Each time a new vehicle position is calculated, the vehicle positioning service informs asynchronously the other components regarding the new position.

The electronic horizon is an easily assessable view of the road ahead of the vehicle, with a configurable distance. The electronic horizon includes the map and associated dynamic data to enable the delivery of enhanced advice to the driver in a timely fashion. This data is distributed in the same way as the vehicle positioning by subscribing to an electronic horizon service.

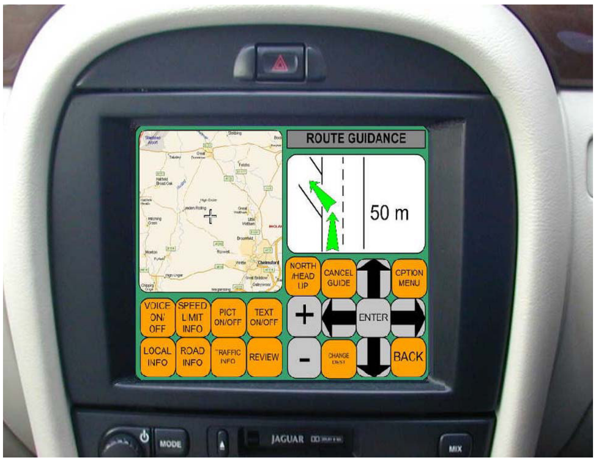

Figure 4: Guidance HMI

Figure 5: Speed Advice HMI

The user interface allows the user to operate the system. It features both touch screen and voice recognition. Typical layout of the graphical user interface is shown in Figure 4 and Figure 5. The design of this part of the system is not optimised as HMI design optimisation is a specialist field and not the main focus of this project.

The map drawing component renders maps, which are displayed by the user interface via the system status monitor. To render maps at all scales it is necessary to include generalised and non-road geometry. Extra features required include rivers and area boundaries. The maps are requested based on an area and zoom level.

The aim of the geocoding component is to identify the location on the map of a place from a partial address, which the user has entered. The data is represented in a 'gazetteer'. Gazetteers were extracted from the map database in an administrative hierarchy to enable the discrete downloading of street data and town/city/village data. Textural information shared amongst all of these is shared via a separate dictionary, where text items are represented as a numerical value. This dictionary entity is referred to as the 'lexicon'. This allows a two level approach to be made, where an address could be geocoded down to village level, when no communication with the server is available. To geocode down to street level would normally require the download of a gazetteer, unless that is already present in the vehicle from a previous usage.

The route calculation component has two outputs. Firstly it calculates a route which is a list of segments which the vehicle will transverse, between its current position and the destination. Secondly, the route calculation component also generates a manoeuvre list which indicates at which points on a route it is necessary to give guidance and the type of guidance that should be given.

The route guidance component generates advice on the manoeuvre route and dynamic advice from the electronic horizon. To do this successfully it is necessary to prioritise certain information higher than other information to avoid overloading the driver. To achieve this, an advice hierarchy was developed with only the highest priority advice given to the driver. Route guidance instructions have the highest priority, followed by regulatory advice, then flow information, followed by any other advice. When flow information is received, the driver is given the option for recalculating the route based in the received information.

The data marshaller is the data access point for the other components and controls the storage lifetime of the data in the in vehicle cache database. To process a request for data, a check is made to see if the data is in the database. If it is then the data object required is populated from the data in the database. If the data is not present in the database, a request is made to the in vehicle server which downloads the appropriate data from the uniform source.

The platform used to prove the ProbeIT concept is based on an industrial PC with a number of peripherals to provide a better level of integration with the vehicle. A Pentium III, 700 MHz industrial PC provides the core computer platform. This provides a similar amount of computing power to the navigation platform, but has a greater level of storage capacity. The existing navigation display was replaced by an 8.5 inch SVGA touch screen monitor and forms the main means of interacting with the system. The system interfaces to the vehicle through both the CAN and D2B bus. An Optolizer is used to communicate on the D2B bus and allows the system to mute the audio system to output instructions and also to allow voice commands to be issued clearly. The CAN interface uses a Softting card and allows messages to be read from the bus, some of which will be used to generate dynamic advice. These physical interfaces are hidden behind the vehicle bus interface component's IDL. The wireless link is provided by an option globetrotter card fitted with an external antenna, which supports a class 10 GPRS connection. A GPRS connection allows the system to operate much more efficiently, as it is not necessary to dial up each time a download is required.

System operation

To use the system the user starts by entering an address i.e. city/town/village followed by the house number and street name. Once the city/town/village name has been identified from data in the on-board cache, street information for that place can start to be downloaded (if it is not already present in the vehicle's cache). The street name and house number is then matched against the street data for that city/town/village. The user confirms the destination and then the system can start to calculate the route. During the route calculation process, the system may download tiles of routing and geometry data from the server, if that data is not already available from the vehicle's cache. There is a high probability that there is sufficient data in the cache to route the vehicle to the high-level road network without the need for communication with the server. Once the route has been calculated, the system starts to generate manoeuvres from the route. The route guidance subsystem in the vehicle is responsible for integrating the manoeuvre and downloaded dynamic information in real-time and delivering this integrated information to the user interface. As the vehicle moves along the road network, the physical sensor data (GPS, gyroscope and road wheel speed) is fused with the map data and route information to provide a prioritised locus of points for the vehicle position. The electronic horizon is continually regenerated, based on updated vehicle position information, and integrates relevant downloaded dynamic information with road geometry. Instructions and guidance are issued to the user by a touch screen display mounted in the central console. The system recognises voice commands and issues audible instructions to the user over the vehicle audio system using a text-to-speech engine. The display shows a moving map, as well as pictograms and textural information. The user interface contains soft keys for a limited set of controls appropriate for use whilst driving, as is common in current navigation systems.

Data structures

In order to support the in-vehicle functions and operation as described above, a data partitioning strategy was designed to minimise the amount of duplicated downloaded data and to allow for a clear distinction between long-term cached data and shorter-term cached data. Several types of data were identified, including: geocoding data containing place names and locations, road geometry data and associated attributes, cartographic-specific data (such as lakes, railways and generalised geometry) and dynamic data (such as speed limits, temporary traffic orders and traffic information). This geocoding data is represented in the previously described gazetteers and lexicon objects. Road geometry and attribute data (for attributes that have a lifetime of the same order as the road geometry) is partitioned on a geographic area tile basis and further by layers of road category and referred to a 'base geometry tile'. Cartographic data is also partitioned on a tile basis and layered by zoom level. Dynamic data is partitioned by geographic area tile and layered by data lifetime. In this way a temporary traffic order, such as electronic motorway warning signs, can be requested and delivered separately from static speed limit data for a road segment.

Each vehicle navigation application requires a particular type of information. For the three applications considered in ProbeIT, this has resulted in six different types of tile information. They are MapDrawingTile, Gazetteer, Lexicon, RoutingTile, BaseGeometryTile and DynamicTile. The definition of the exact format of each type of the tile is specified through its object definition (see Figure 3). This definition is a direct result of the data requirements of the applications.

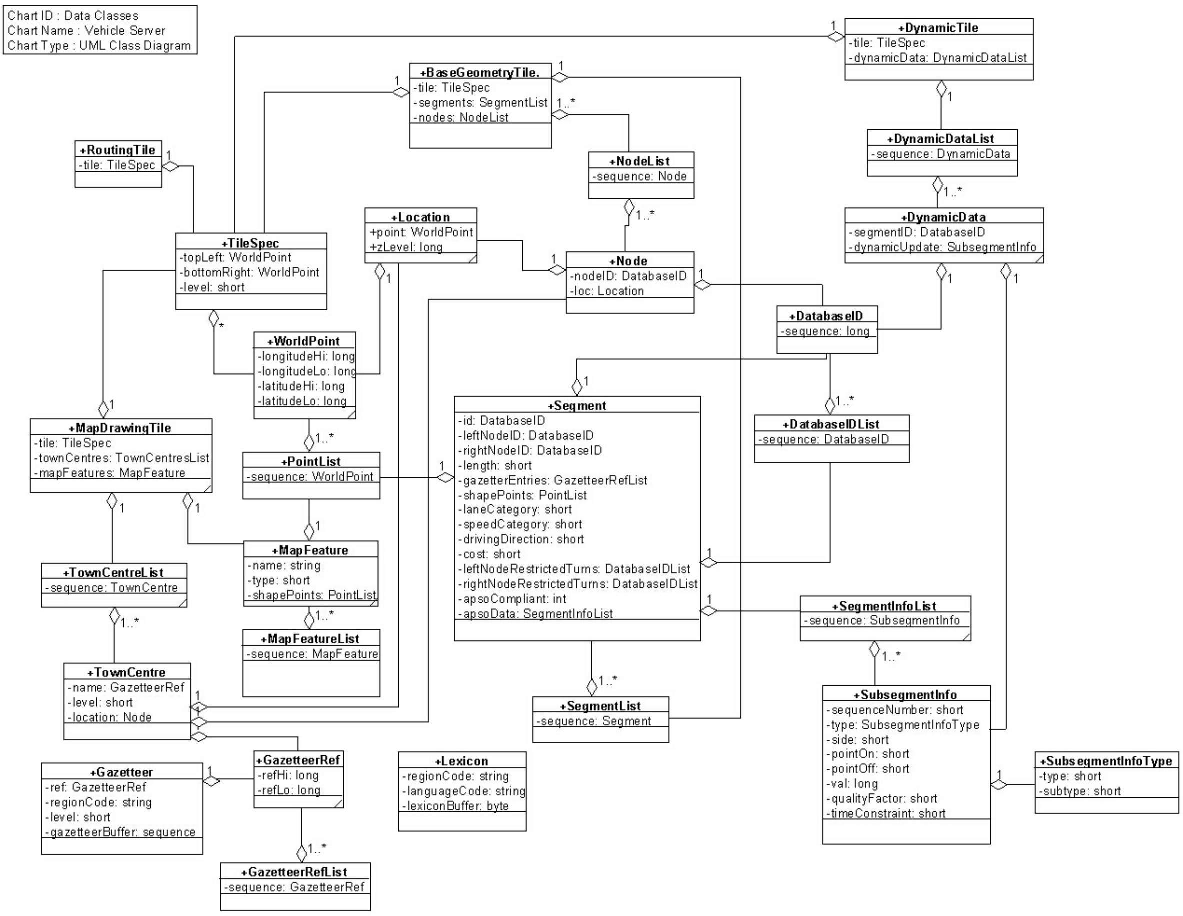

Figure 3: Vehicle-Server Interface Model

Dynamic data strategy

Managing dynamic data on the vehicle is the most important aspect of the project. Each of the dynamic data used in this project has dramatically different dynamicity. For example, variable speed limits on the M25 can be considered to have a life time of 2 minutes, but traffic order road works may have a life time of several months. To get around this problem a number of dynamic levels have been defined each relating to a different lifetime of the data on the vehicle.

Floating car data

Floating car data is also used as a 'dynamic' data source in the framework. One of the difficulties with using floating car data is the huge volume of data from all the vehicles fitted with the system. The key idea behind the strategy used is to perform exception reporting only. For example, an incident is reported by the vehicle if the vehicle hasn't been informed of that incident already. This Floating Car Data (FCD) mechanism minimises data exchange between vehicles and server and the computational load on the server in generating FCD based dynamic information. On the vehicle side an exception is triggered if the measured parameter is outside the threshold band of the dynamic attribute. An example of this is a delay due to an accident; the first few vehicles passing by will experience a much longer link time compared to the time received from the server, and therefore will inform the server. After a server has received the data from a few vehicles it will add a dynamic data entry to update the link with an additional time. Newly informed vehicles will then experience a link time comparable to dynamic information and will not tell the server of any discrepancies. Once the incident has cleared vehicles will experience a shorter link time and consequently report this to the server with the result that server information is updated with the real network status. Besides real-time adaptation, the FCD can be used in the long run as historical data to adjust the static weight of the link segments and provide the basis for predictive adjustments of link times at peak periods.

SERVER ARCHITECTURE

The objective for the server is to generate a uniform information resource which can be served to the vehicle to enable ADAS. This means the server should be designed such that different sources of geo-referenced information can be integrated and handled efficiently and that the vehicle-server interface can be populated in real-time.

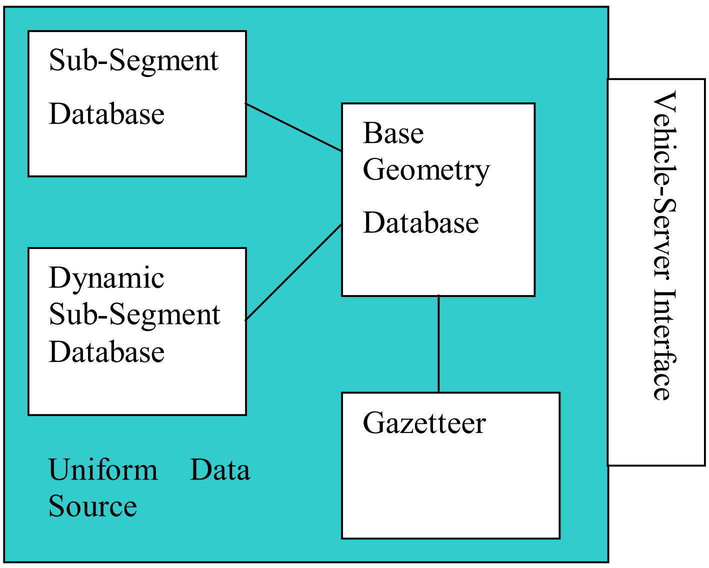

Figure 4: Server Architecture

This has resulted in the following features. The core element of the uniform data source is based on standard navigation database. In the ProbeIT case this is Navigation Technologies' NAVTECH data. This data, and more specifically the basic geometry information, is the underlying network to which all other information is referenced. For the purpose of the project, test areas have been augmented with shape points, which specify detailed road layout. The architecture of the server is illustrated in Figure 4.

Referenced to the underlying geometry are three sets of sub-segment information, these describe 'static' and 'dynamic' information, and the gazetteer. The 'static' sub-segment information corresponds to data which varies on an infrequent basis and is managed through a strict update process. The information includes fixed mandatory speed limit information, traffic regulations and any other fixed attributes of the road. The information is the result of a data integration process, combining many data sources into one. The data is stored in a database separate to the base geometry database, and is called sub-segment information. The 'dynamic' sub-segment information corresponds to highly variable information and includes variable speed limits, accidents and road network status, which is again integrated from many sources into the dynamic sub-segment information database.

The gazetteer information is pre-processed before it is stored in the gazetteer database. The main reason for the pre-processing is to improve performance and in particular the speed of response. The result of pre-processing produces a gazetteer format which is tightly coupled to destination entry and map display functions on the vehicle. The data is 'static' in nature and only updated infrequently.

Detailed design and implementation is provided within the paper entitled 'ProbeIT – Vehicle Server and Data Integration'.

CONCLUSION

The results above show that the design and partial implementation of an integrated system to realise vehicle applications, like dynamic navigation, speed advice and traffic regulation advice, has been achieved within the ProbeIT project. The key outputs of the ProbeIT project are:

- that vehicle applications, relying on geo-referenced data, can be designed using off-board data, sourced on demand from a central server; and,
- that requirements driven by the vehicle applications can be accommodated in the design of interface and server.

For possible future work, a safety critical analysis of the data exchange mechanism is required to determine if the approach is suitable for next generation ADAS applications. Included in this further research would be the task to determine the requirements on satellite positioning to allow integration of map based data into these ADAS systems.

REFERENCES

[1] QMISS website: http://www.qmiss.org.uk

[2] Travel Information Highway (TIH) website http://www.tih.org.uk

[3] Common Object Request Broker Architecture (CORBA) website http://www.corba.com and www.org.org

[4] Controller area network (CAN) website http://www.can.bosch.com