Qmulus – A Cloud Driven GPS Based Tracking System for Real-Time Traffic Routing

Niyati Parameswaran, Bharathi Muthu, Madiajagan Muthaiyan

Abstract**—**This paper presents Qmulus- a Cloud Based GPS Model. Qmulus is designed to compute the best possible route which would lead the driver to the specified destination in the shortest time while taking into account real-time constraints. Intelligence incorporated to Qmulus's design makes it capable of generating and assigning priorities to a list of optimal routes through customizable dynamic updates. The goal of this design is to minimize travel and cost overheads, maintain reliability and consistency, and implement scalability and flexibility. The model proposed focuses on reducing the bridge between a Client Application and a Cloud service so as to render seamless operations. Qmulus's system model is closely integrated and its concept has the potential to be extended into several other integrated applications making it capable of adapting to different media and resources.

Keywords**—**Cloud Services, GPS, Real-Time Constraints, Shortest Path, System Management and Traffic Routing

Acronyms**—**CBGPS: Cloud Based GPS Model
GeoDB: Geographical Database
GPS: Global Positioning System
SysDB: System Database

I. Introduction

THIS paper presents a CBGPS Model-Qmulus which serves to provide the best route to the destination amongst the possible paths generated by the System. The model uses dynamic computation for travelling time, implements real-time constraints and presents constant trafficking updates all of which would aid in generating optimal routes.

The model has been designed using UML [1] as it provides an efficient system which would be easily adaptable, will permit additions specific to new developers and allows ease of communication.

The operations would be implemented using cloud deployment system [2] in order to boost scalability, flexibility, manageability, customizability and provide access to a potentially much larger bracket of people.

Niyati Parameswaran is a student of Birla Institute of Technology & Science Pilani, Dubai Campus, DIAC, Dubai, UAE (email: niyati.15@ gmail.com).

Bharathi Muthu is a student of Birla Institute of Technology & Science Pilani, Dubai Campus, DIAC, Dubai, UAE (email: bharathi.smm@ gmail.com).

Madiajagan Muthaiyan is an Assistant Prof. with Birla Institute of Technology & Science Pilani, Dubai Campus, DIAC, Dubai, UAE (email: jagan@bits-dubai.ac.ae).

II. Problem Formulation

One of the pressing day-to-day problems people in metropolitan cities encounter is getting caught in traffic snarls. As a result they end up spending a lot of time on travel. The constantly rising amount of vehicular traffic not only adds to creating large overheads in terms of time lost but also adds to expenses in terms of fuel cost. The Air pollution Barometer also witnesses a rise. Moreover, as commute forms a vital requisite for a large number of people and almost every major city is linked by an extensive roadways system a populous sample space is getting affected reinstating why this area requires more focus.

The current methodologies and technologies supported by transportation authorities and other government regulated bodies are more infrastructure dependent. As a result are not very scalable or flexible. These limitations culminate in time and costs overheads. Currently, the maps are not updated in real-time for all models and GPS devices provide route priorities based on static information which can be misleading for drivers. Moreover the routes are created on the basis of distance as a factor alone which may prove to be expensive if real-time trafficking is ignored. Since traffic levels are a variant of real-time, it becomes important to adapt practices which would implement dynamic principles and be able to provide insight about surroundings based on current scenarios readings with feasibility for real-time modeling and updating.

There has been a tremendous research and development in fields related to the creation of autonomous vehicles. The huge leaps and successes made in this area re-emphasize the importance of a surveillance system which would be able to track down an efficient yet safe path from the source to the destination. The routes projected need constant updates about change in environments-be it due to pedestrians or other obstructions. Also, the system should model suggested routes by taking into account the possible occurrence of an unexpected event such as an accident, signal malfunction or a chaos triggered roadblock. This surge in development further promotes why real-time routing and dynamically updating

SysDB is crucial and currently requires focus.

III. Problem Solution

An efficient traffic management system which would be able to provide accurate intuition to the drivers in terms of choice of routes while taking into consideration several factors related to traffic management would be able to assess the situation and deal with this Urban Challenge [3].

This paper presents a design which would compute the optimum route(s) that a driver could select from on the basis of the destination specified. The system modeled comprises the Client and the Cloud which communicate on the basis of a Request and a Response.

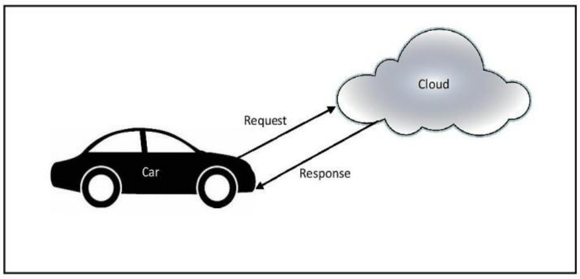

Fig. 1 Schematic block diagram for CBGPS-Qmulus

The GPS application is hosted by the Client and the functionality of this GPS application gets deployed on the Cloud. The System encompasses both the Client application and the Cloud services thereby executing the architecture of the artifacts it represents and integrating these elements.

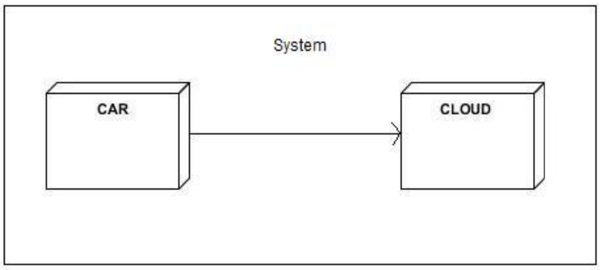

Fig. 2 UML Deployment diagram for CBGPS- Qmulus

Fig.4 illustrates the System Model using packages as entities. The Car Console and the Cloud Service form the primary namespaces in this representation.

The Car Console is composed of the Car Display, the Clock and the GPS Tracker. These entities lie in the same inheritance hierarchy, are intrinsically related and also collaborate with each other. The Car Console imports these entities as per requirement and binds them semantically thus providing better structure to the Console model.

The Cloud service is composed of the Map Reader, the Path Finder, the System Manager and the Route Finder. These namespaces exhibit dependencies and import features from each other to implement their specific functionality. Once the Map Reader has created a cartographic representation of the area under study, the Path Finder will derive from it. The System Manager imports the conclusions drawn by the PathFinder and works on it to provide a broader, detailed perspective while incorporating intelligence through a sense of real-time routing. Finally, the Route Finder amalgamates and merges these chunks of data and adds realization to these packages by implementing their logical specifications

As depicted in Fig. 5, the packages have been further broken down into components which are capable of being deployed and re-deployed repeatedly and independently. Each component represents a modular part of the system which encapsulates the System's content. The component model thus proposed is defined by both provided and required interfaces thereby encompassing both static and dynamic semantics.

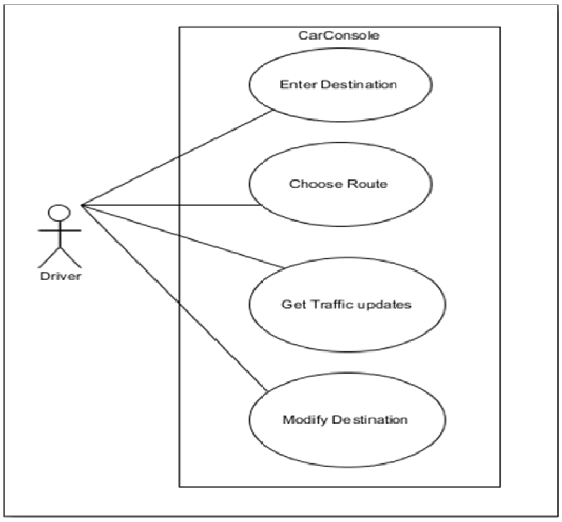

Fig. 3 Use case diagram for Car Console

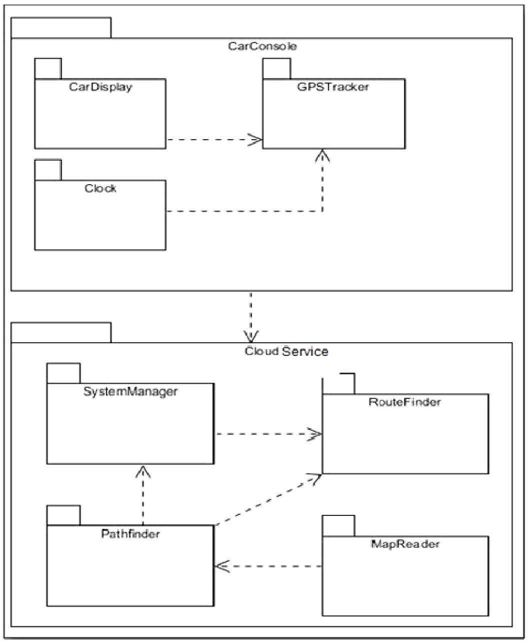

Fig. 4 Package diagram for CBGPS-Qmulus

The component model thus proposed is defined by both provided and required interfaces thereby encompassing both static and dynamic semantics.

The two primary components are:

- Car Console
- GPS System

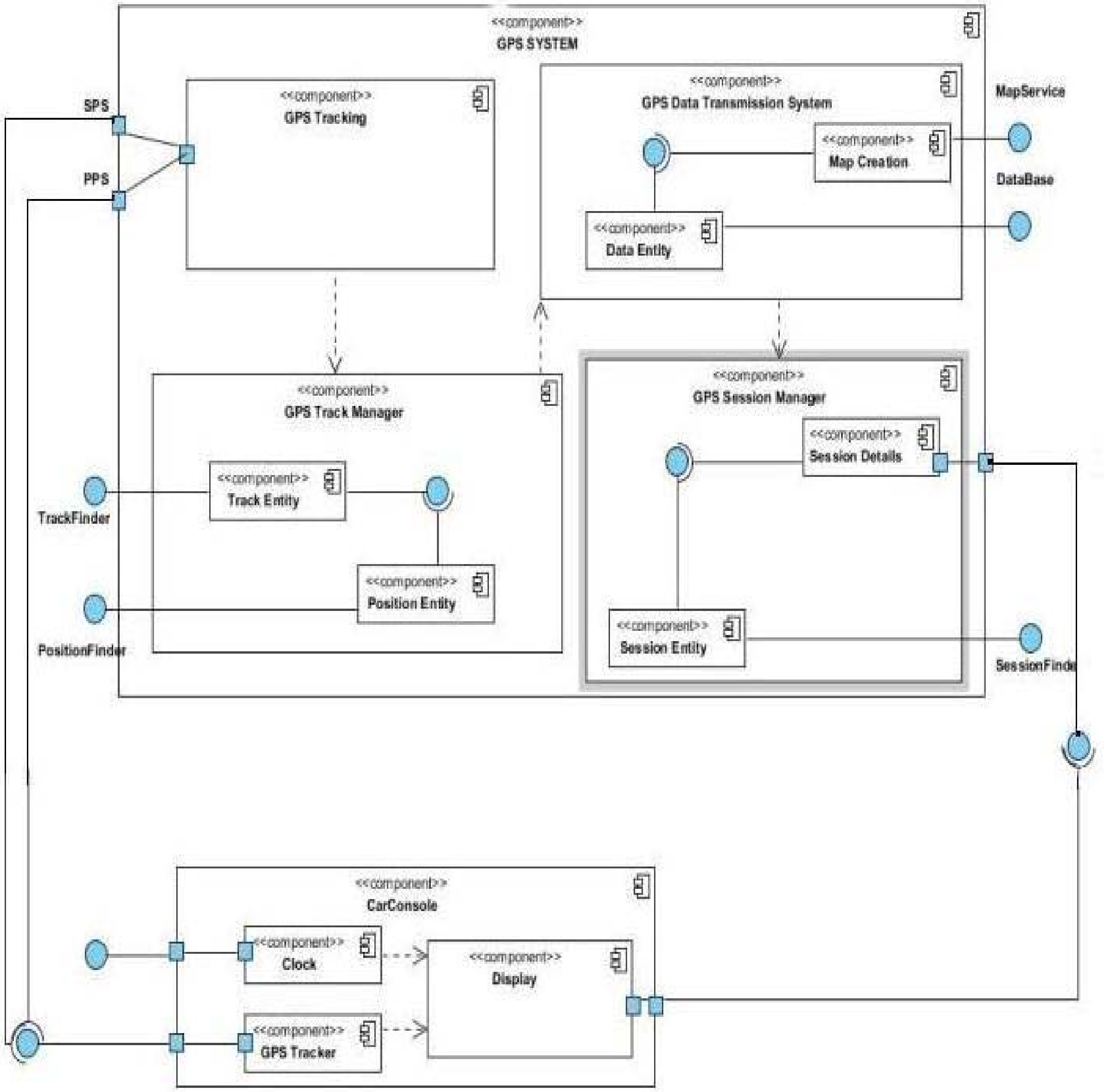

Fig. 5 Component diagram for CBGPS-Qmulus

The Car Console is composed of a Clock, GPS Tracker and a Display Component. The GPS System is composed of GPS Data Transmission, GPS Tracking, GPS Track Manager and a GPS Session Manager Component.

The functionality of these various components and sub-components has been illustrated in the following.

A. Car Console

The Car Console forms part of the Car System. Our assumption here is that every car has a functioning GPS System installed. The sub-components of the Car Console are:

Clock:

The Clock component will keep track of the current time and will be linked with the Display component in the car.

GPS Tracker:

The GPS Tracker component will notify the GPS System on its activation and will work closely with the GPS System component.

Display:

The Display component will be responsible for taking in inputs relating to the destination and will also constantly *update* the car display as per the driver's current location after having provided the possible routes that the driver could take.

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:7, No:1, 2013

The display component will be customized to suit the driver's needs.

B. GPS System

The GPS System on being switched on will take in the inputs corresponding to the source and the destination from the driver and in turn would trigger working of its various other components. It will be responsible for sending the request to the Cloud Component. The services entailed to the GPS System would all be performed on the Cloud. The subcomponents of the GPS System are:

GPS Data Transmission:

The GPS Data Transmission component will utilize the locations inputted to render a map service plotting the source and destination on a map which would be obtained from an online database. A local map would get downloaded and plotted in accordance with the source and destination mentioned.

GPS Tracking:

The GPS Tracking component will make use of the coordinates based on the map rendered and will apply the Standard Tracking and Precision Tracking Services. These services will constantly update the real-time routing system on the basis of the current location of the car as per the map generated.

GPS Track Manager:

The GPS Track Manager component will be responsible for keeping a log of current and new tracks as well as getting and posting current and new positions.

GPS Sessions Manager:

The GPS Sessions Manager will provide session details and other session related information. It will also permit the driver to terminate a currently running session in case the driver decides to change destination. The Sessions Manager in association with the Track Manager will be responsible for generating updates ensuring the selection of a route which will result in reaching the destination in shortest time.

Fig. 6 illustrates the Entity Diagram for CBGPS-Qmulus. The following will provide intuition on the functioning of the various classes that form part of Qmulus's design.

On switching on the GPS System in the Car, the GPS System's components will invoke operations that would be deployed on the Cloud. These operations will help track the current co-ordinates of the Client in accordance with the GPS. By providing cars with a GPS Systems, the cars effectively could utilize a wireless communication system by the creation of an ad-hoc network.

On acquiring the co-ordinates of the current location and fixing the co-ordinate for the destination, the Map Reader class gets called. This class renders a map creation service by generating a map using the online GeoDB hosted on the Cloud. The various landmarks are mapped into a 2D pictorial representation which can be further interpreted for more intuition on the location by utilizing a number of vector and raster formats, cartographic symbol depiction and built in collaboration updates.

Once the map has been generated, the Path Finder class uses this map to create possible paths to the destination. It works on both the spatial and non spatial data sets [4] and takes into consideration a real-time system which would permit full fledged vector editing with respect to pathways generated. The algorithm will first discover adjacent nodes by implementing neighbor discovery technique based on the computed distance of each path. Once the nearest neighbors have been discovered, the path discovery will generate possible paths that could be chosen and Djikstra's algorithm [5] will be used to compute the shortest path while also assigning priorities to these paths.

After the possible paths have been identified the Manager System begins its execution. The purpose of the Manager System is to constantly evaluate the traffic level scenarios on the paths proposed. This system is closely integrated with the online traffic portals and is tied up in a high speed request response editable network community. The Manager System primary purpose is to maintain real-time routing and implement real-time constraints. Various traffic management parameters such as Jam Level, Congestion Level [6], Bottleneck areas [7] are investigated and the notifications are recorded. Intelligence is incorporated to this system by means of which it has the potential to draw conclusions on peak time trafficking in the roads based on 24/7 surveillance. The Manager will secure updates from the GeoDB on a frequency that is flexible in accordance with the necessity and the number of path options so far generated. Further, the Manager System will have the potential to deal with unforeseen situations like an accident or an oil spill or a signal malfunction by predicting possible delays in these situations.

Using both supervised and unsupervised algorithms the Manager System's performance can be augmented. Using the notifications provided by the Manager System and the paths identified by the Path Finder, the Route Finder will compile time estimates for the various paths. It will take into consideration the parameters associated with trafficking, regulations on the specific proposed roadway stated by the Local Transport Authority, road type, total distance and flexibility to change path in case of uncalculated incidents. It would then present routes along with their priorities. These routes will be plotted on the screen of the Car System and the driver could make the choice he/she wants on the basis of the recommendations made by the Cloud Component. Once the response from the Cloud has reached the Client (here, the Car System) and the choice of route has been made by the driver, the system would loop back the current location co -ordinates and feed it to the GPS Tracker so that the functioning System would now acquire a real-time intuition with the driver being constantly aware of the route being travelled on. The driver will also be notified appropriately whenever a modification to the choice made is necessary.

The various classes in this model serve to holistically represent and systematically utilize the information that is handled by the entire system while associating behaviors to represent a conceptual modeling of the CBGPS-Qmulus's framework.

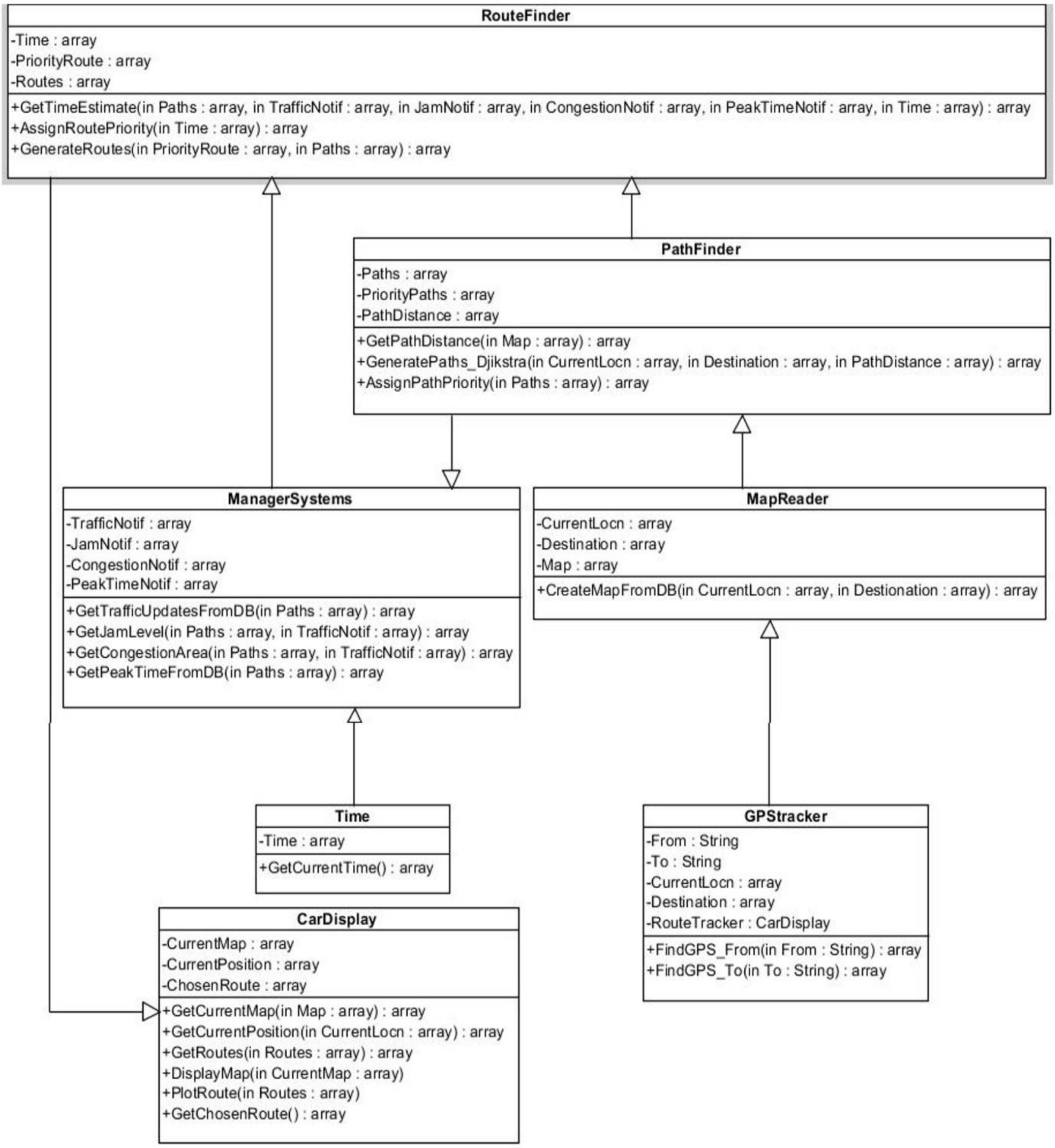

Fig. 6 Entity diagram for CBGPS-Qmulus

IV. Conclusion

A Cloud based GPS System incorporates several features which re-instate its suitability for usage. These applications can be created, tested, implemented and debugged easily. Since there is no requirement for new installation or new downloadable updates, every new feature introduced to the model is immediately made available to the customer bracket. There is no restriction in terms of access as a cloud driven application will secure access everywhere. The System's usability can be enhanced by using more media, by incorporating Saas [8]. Lastly, the integration opportunities associated with cloud deployed applications are immense as these applications have been built with the sole purpose of grounding up from the grassroots level.

The constraints which need to be dealt are certain issues that might crop up with the implementation of the service on the cloud. Retrieving our maps which have been stored on the

Cloud could affect the aesthetic appeal as the refresh and update rates of showing the map might get a slower. The Rerouting speed [9] needs to be dealt with very strictly in the implementation as the intelligence must recognize that a track has been changed as early as possible and should not always need to wait for the completion of a network cycle to notify the driver. Finally providing an offline alternative to the project will be an apt extension as it will ensure seamless operations even if the connections get dropped due to lack of network availability for a small period of time such as in cases of cloud outage.

REFERENCES

[1] Fatma Corut Ergin. “What is UML.” mimoza.marmara.2011. Available:http://mimoza.marmara.edu.tr/~fatma.ergin/courses/cse344/PS/UML.pdf

[2] Tarun Karnwal, T. Sivakumar, G. Aghila, Cloud Services in Different Cloud Deployment Models. An Overview,International Journal of Computer Applications (0975 – 8887) Volume 34– No.8, November 2011.

[3] Chen, Wei and D. Levinson (2006), Traffic Management System Performance Using Regression Analysis: A case study of Mn/DOT’s Traffic Management Systems, California PATH Working Paper for California Department of Transportation UCB-ITS-PWP-2006-5.

[4] Mohammad A. Rob, Some challenges of integrating spatial and non spatial datasets using a geographical information system, Information Technology for Development 10 (2002) 1–8 ITD92.

[5] Juan Chamero,”Dijkstra's Algorithm As a Dynamic Programming strategy”, intag.org, April2006, Available: http://www.intag.org/downloads/ds_006.pdf.

[6] Thammasak Thianniwet, Satidchoke Phosaard, Wasan Pattara-Atikom, Classification of Road Traffic Congestion Levels from GPS Data using a Decision Tree Algorithm and Sliding Windows, Proceedings of the World Congress on Engineering 2009 Vol IWCE 2009, July 1 - 3, 2009.

[7] Fondazione Eni Enrico Mattei & Scuola Superiore Enrico Mattei, Externalities in the Urban Transport: Assessing and Reducing the Impacts, International Symposium on Technological, Energy and Environmental Topics in Transport, October 27-29, 1998.

[8] Dillon, T., Cloud Computing: Issues and Challenges.

[9] On-Board VS Cloud Maps for Apps, gpsreview.net. Available: http://www.gpsreview.net/on-board-vs-cloud-maps-for-apps/.W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

Niyati Parameswaran is pursuing her final year in Computer Science & Engineering at Birla Institute of Technology & Science Pilani, Dubai Campus. She intends to pursue specialization and do substantial research. She is proficient in C, C++, JAVA, VB, Octave, Perl and jQuery. Her fields of interest include Machine Learning, Computational Intelligence, Neural Interface Systems, Web Security, Crypt Analysis, Cloud Computing and Literature.

Bharathi Muthu is pursuing her final year in Computer Science & Engineering. at Birla Institute of Technology & Science Pilani, Dubai Campus. She intends to pursue Operations Management. She is proficient in C, C++, JAVA, XML and UML. Her fields of interest include Data Mining, Cloud Computing, Digital Electronics and Computer Organizations, Artificial Intelligence and Neural Interface Systems.

Madiajagan Muthaiyan holds a PhD in Component based software Development from Birla Institute of Technology & Science, Pilani, India and a M.S. in Software Systems from Birla Institute of Technology & Science, Pilani, India. He has 18 years of College / University teaching experience and 2 years of experience as System Analyst in a Blue Chip Software Company. Presently, he is working as Assistant Professor of Computer Science at BITS Pilani, Dubai Campus, U.A.E. His areas of interest include Cloud Computing, Component Based Software.Engineering, Distributed Engineering, Distributed Database Systems, Software Architecture, and Theory of Computation. He is a Professional member of Professional bodies ACM, World Enformatica Society and Computer Society of India.