Visualization and modeling of traffic congestion in urban environments

Authors: Ben Alexander Wuest and Darka Mioc, Department of Geodesy and Geomatics, University of New Brunswick, Canada. P.O. Box 4400, Fredericton NB Canada E3B 5A3,
E-mail: bwuest@nbnet.nb.ca, dmioc@unb.ca

ABSTRACT

Routing has become a common feature or accessory in today's modern vehicles. Determining the most efficient route to a destination is attractive to consumers and very important to emergency response services. These products employ road maps that may or may not be up-to-date with the current geographic street information. Companies are emerging to provide current road information, restrictions, and points of interest (POIs) to smart vehicular routing systems. As this technology expands, event driven systems that integrate with real time information are emerging. Real time information, such as street congestion monitoring and traffic restrictions, provides a non-static dimension to geographic street data. This opens the door for dynamic routing systems which provide real time path information that is adjusted according to dynamic monitors and network information.

This paper presents dynamic routing using Dynamic Shortest Paths *(DSP)* in Internet based Geographic Information Systems. The technology base for this paper is Oracle 10g, its associated Internet mapping technologies, network data modeling, and shortest path implementations. We explore the network data model of Oracle 10g and the process of building a dynamic street network. In this paper we present an algorithm, the implementation and results for a DSP solution with traffic congestion. We present the prototype developed for this paper and we detail how the dynamic information simulated in this prototype could be integrated with emerging GPS technologies and dynamic monitoring devices. The major contribution of this paper is to provide a real time, Internet based, demonstration of a vehicle routing DSP solution in Oracle 10g based on traffic data and GPS simulation.

INTRODUCTION

This paper presents a prototype for visualizing and modeling traffic congestion in urban environment and applied dynamic routing using Dynamic Shortest Paths *(DSP)* in Internet Geographic Information Systems. The spatial database technology used in this research is Oracle 10g, its associated Internet mapping functionalities, network data modeling, and shortest path implementations. We explore the network data model of Oracle 10g and the process of building a dynamic urban street network, which updates according to simulated street congestion.

In this paper we present a solution in the framework of a taxi dispatching service. The ability to dispatch a taxi from starting point to its destination is implemented. The Taxi Route is simulated according to current traffic patterns in the urban environment. The prototype dynamically visualizes adjusted taxi paths according to traffic congestion through a DSP solution within the Oracle 10g database.

An algorithm is developed and results are given for a DSP solution with traffic congestion. This algorithm is based on Dijkstra's algorithm as implemented in the Oracle 10g database. The database design and prototype implementation are presented. We can also demonstrate how traffic congestion affects a dispatched taxi and visualize this response in our prototype. Applications of this medium are important to any kind of routing system; more particularly in the domain of emergency response services. This visualization could be applied as an in-car service to help drivers visualize the most effective route in an urban environment. The major contribution of this paper is to visualize the effects of traffic congestion in an urban environment through a DSP solution in an Internet GIS.

DATA ACQUISITION AND MODELING

The data presented in this paper was all prepared using the Oracle 10g database. This section outlines the different models employed by the prototype. This paper will not go into the finer details of the Oracle 10g database. The basic model for maps within Oracle is represented in Figure 1.

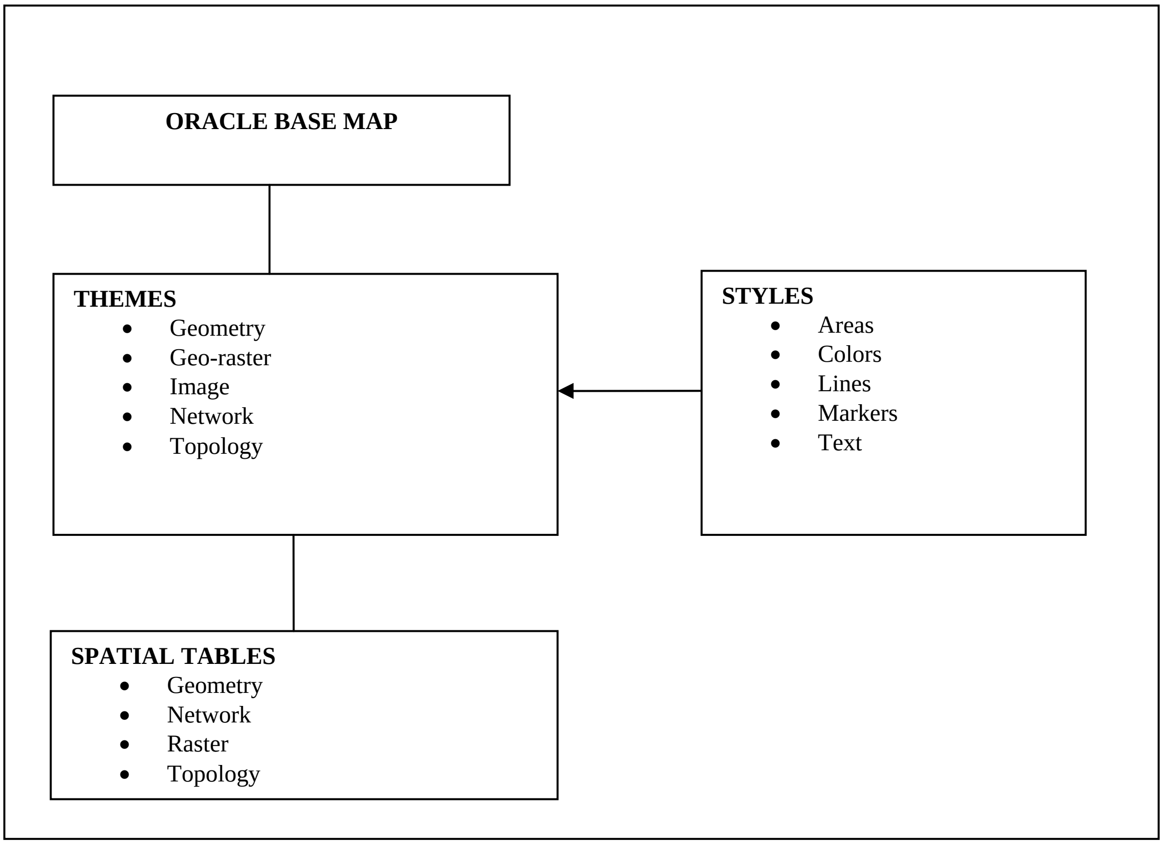


Figure 1: Oracle Map Data Model

URBAN TRAFFIC CONGESTION

Traffic congestion in urban environments is a regular phenomenon. Major city arteries become congested as people leave their offices at the same time. Other restrictions such as traffic accidents and street works cause sections of streets to slow and sometimes obstruct navigation completely. These occurrences cause navigation to sometimes be tedious and can disturb the flow of emergency response services in which time is crucial. Visualization of traffic congestion and automated response to changing traffic patterns would be a powerful feature of any urban GIS in today’s world.

A small entity relationship model was designed and entered into the database (see Figure 2). This entity relationship is a sort of street topology over the street geometry data. The nodes for this model were extracted using the Oracle Network Data Editor.

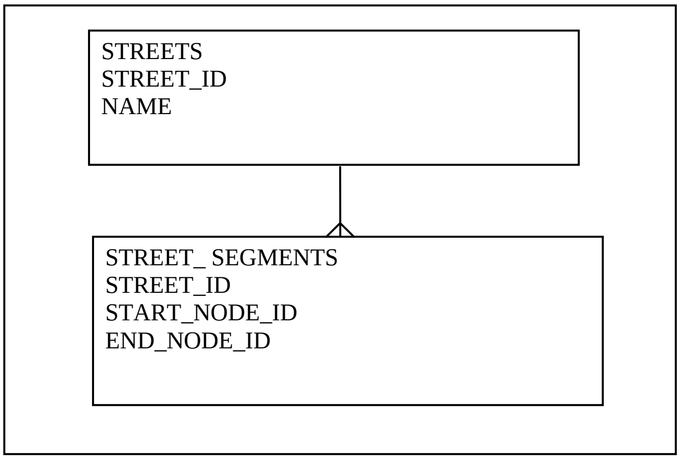


Figure 2: Street Entity Model

Network Model Construction in Oracle 10g

Oracle 10g was used to construct a network model and to design a spatial database. In this version of Oracle a solution for building networks for geographic data is provided. The Oracle 10g model provides a schema, called SDO_NET for analyzing data in network model (Kothuri et al, 2004).

For this research, data was acquired for the prototype to generate a street network of a city. The data was obtained from the City of Fredericton, New Brunswick. This data was acquired from a subset of an ESRI shape file. The data (presented in Figure 3) was imported into the Oracle 10g Map model via Oracle Map Builder.

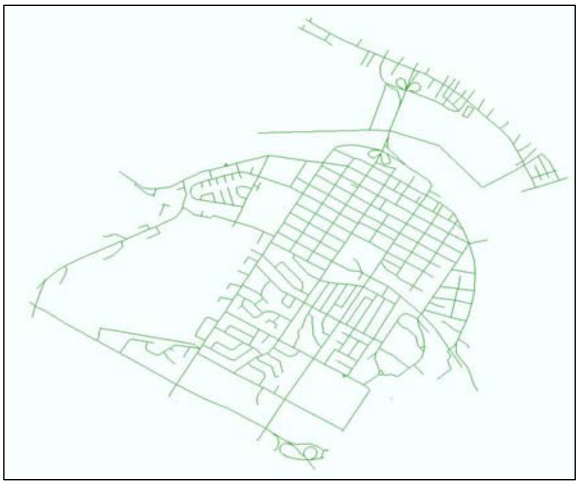


Figure 3: Street Network – City of Fredericton

Within the created network model, additional attributes were created to identify the following data for given street segments:

- Average Speed *(km/h)*
- Length *(km).*

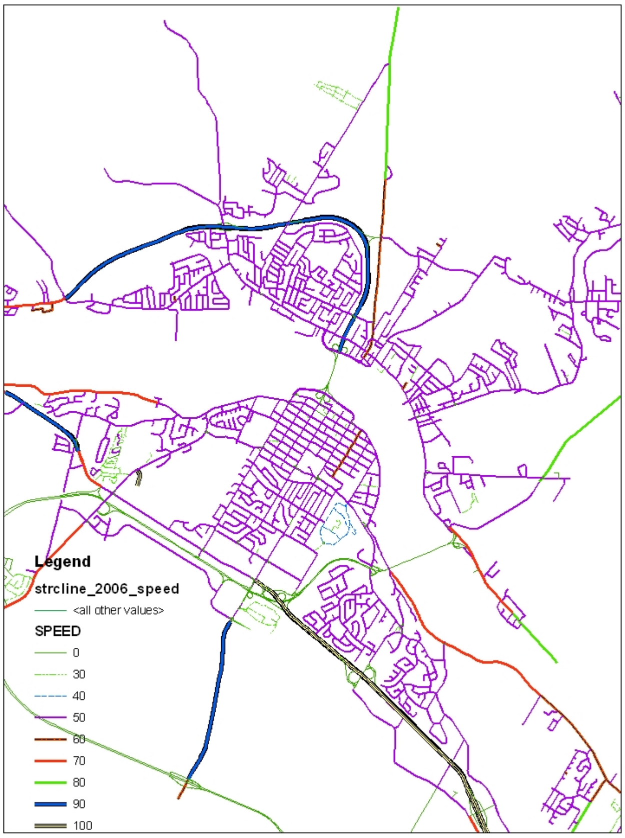


Figure 4: The average traffic velocity in the City of Fredericton

It is necessary for the application to be able to set street speed data on streets within the downtown Fredericton area. Every street segment is assigned an average speed (see Figure 4). The data containing the observations of average speeds was obtained from the Transportation Department, City of Fredericton. We are using this average speed data for a simulation of a street monitors in a real time system. The application can set an average speed for a given street using this model, but within applications we can change the street speed due to observed traffic congestion. This relationship allows for the prototype to simulate traffic congestion on major urban arteries. This simulation is of course only one type of traffic congestion that can make street networks dynamic.

The data was processed into the Oracle Network Model using PL/SQL programming. The network data was created manually as a View into the imported street network. The implementation of the network as a view provides the ability for dynamic speed changes of street segments to affect the cost of traveling across the network with respect to time.

THE ALGORITHM

This section presents the algorithm developed for our prototype. We employed the Dijkstra version of the shortest path implementation that exists in the SDO_NET component of the Oracle 10g database. This provided us with the ability to calculate a static path at a particular time.

Dynamic Shortest Paths

The Dynamic Shortest Path problem is an abstraction of the Shortest Path Solution. A shortest path is one that is calculated based on a static network (Grier et al., 2002). It is assumed that the contents of the network will not change. However, in most real world situations, networks are not static. Road information may be static but new technology is emerging to monitor street speeds and report traffic restrictions. This layer of event driven data is adding a non-static dimension to traditional GIS network data.

To outline the problem of a Dynamic Shortest Path *(DSP)* we will take for example a network model *NM*,

$$NM = (N, L)$$ where

N – set of nodes
L – set of links.

In the following scenario we will assume that the shortest path SP_0 between the nodes A and B has been calculated on network model *NM* using the cost function of time. In other words, the algorithm has found the optimal path in the network to achieve the shortest travel time from node *A* to node *B.* The route initiation time from node *A* is t_0. The arrival time at node *B* is represented by t_a. Route estimation is based on the current status of the network at t_0. This calculation will be valid for $t_i : i = \{0, a\}$ when network model *NM* is a static network. It will not be valid for $t_i : i = \{0, a\}$ when the network model *NM* is subject to change. Many factors are responsible for changes in the network. In this paper traffic congestion represents the variable responsible for our dynamic network model.

The DSP solution developed is an iterative approach of successive shortest path analysis on the database. The successive calls are made by the visualization in the prototype from each node passed on a given path.

We derived the following function for price based on the following variables:

i) Distance - $d(SP_0)$ in Kilometers.
ii) Number of Passengers – *np* and
iii) Base Price - *bp* set to $2.50, a simple formula was devised and price was formulated.

$$price(SP_0) = bp + 1.0 * np + 0.5 * d(SP_0) \quad (1)$$

The algorithm is an implementation of combining our price model with iterative calls to the Oracle 10g shortest path query. The following points describe our algorithm:

- The shortest path SP_0 from node *A* to node *B* at t_0 is calculated.
- The nodes *N* of SP_0 using the speed information on the streets table as an indicator of distance traveled on the given path SP_0 are traversed.

- For each node reached on the path the current distance $d_i(SP_0)$, time $t_i(SP_0)$, and the current path $SP_0^{'}$ that represents the part of SP_0 already traveled is calculated.
- The new shortest path SP_i from node *A'* (*current node)* to node *B* at t_i is calculated.
- SP_0 is updated such that $SP_0 = SP_0^{'} + SP_i$.
- The current time for SP_0 such that $t(SP_o) = t_i(SP_o) + t(SP_i)$.
- The current distance for SP_0 is updated such that $d(SP_o) = d_i(SP_0) + d(SP_i)$.
- Repeat the algorithm for the updated network until all nodes of the path have been traversed.

This process allows for dynamic changes to the network to affect the path chosen and thus implement a version of the Dynamic Shortest Path algorithm using the Oracle Database. This algorithm is limited in the sense that the granularity of decision making is done as each node is traversed. It is a practical solution and provides a successful visualization for the prototype presented by this paper. In our future work we plan to change the granularity of decision making to make it more sensitive to the changes within the street network model.

THE IMPLEMENTATION

The prototype is implemented through Oracle Application Server MapViewer and its embedded Java API. In addition to the Java API, Oracle provides a JSP and XML request layer (Kothuri et al, 2004). These technologies were not used because the Java API provides more flexibility. We developed a number of PL/SQL procedures.

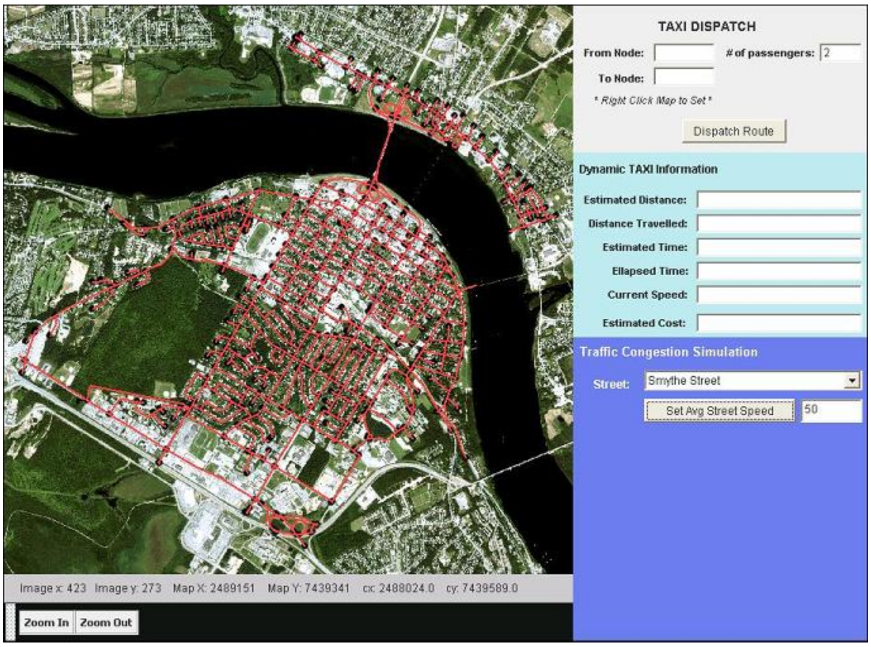
To implement the prototype a number of PL/SQL procedures were developed to interact with the data models described in previous sections of this paper. These programs were developed to ease the client processing and overcome limitations in the current MapViewer application server functionality. This section will outline the PL/SQL procedures designed and implemented.
These are database embedded scripts that can manipulate data or communicate with connecting clients through a JDBC interface. Not all of these procedures are detailed as there were far too many.

The Applet

The first part of the implementation involved designing an interface to allow for:

a) The user to dispatch a taxi route,
b) The user to view (see Figure 5) the current dynamic taxi information *(current speed, distance traveled, & elapsed time)*,
c) The estimated distance and time to the destination,
d) The current congestion simulation in the urban environment and
e) The user to be able to manipulate the map using simple map functions such as pan and zoom during the simulation.

This was developed using Java swing and AWT components in JDeveloper.



# of passengers: 2

Street: Smythe Street

Set Avg Street Speed: 50

Image x: 423 Image y: 273 Map X: 2489151 Map Y: 7439341 cx: 2488024.0 cy: 7439589.0

Figure 5: JAVA Applet Application Interface

Shortest Path Generation

In analyzing the capabilities of the Java API provided by the Oracle MapViewer service, a limitation was found. It was discovered that although the server provides the ability to calculate the shortest path from one node to another on a given network, it does not store the results of this path persistently in the network results tables. Therefore, an Applet does not have the ability, through a MapServer connection, to retrieve the geometric data for a shortest path calculated by the MapServer. The only information that can be retrieved is a dynamic image layer of the resulting shortest path. To overcome this problem a PL/SQL program was designed and implemented. In summary, this program interacts with the SDO_NET PL/SQL package and performs the following actions:

i) Receives the calling client id, client path index, graph start node, and graph end node as parameters.

ii) Loads the FREDERICTON_STREET_NETWORK using SDO_NET routines.

iii) Executes a Dijkstra shortest path query on the given nodes using SDO_NET routines. By performing this, the results are stored in the NET_FREDERICTON_PATHS and NET_FREDERICTON_PATH_LINKS tables associated with the street network.

iv) Retrieves the calculated path id and time for the results for the resulting path.

v) Makes an entry in the client analysis table *(route_table)* using the given client id, client path index, calculated path id and time. This is done so the client can make a subsequent JDBC request to retrieve the details of the calculated path.

Taxi Thread JAVA Class

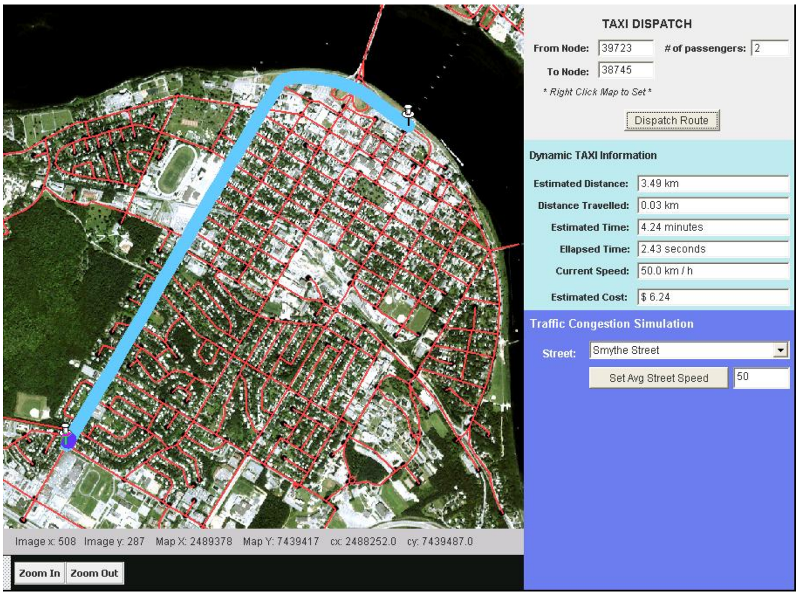
For the dynamic visualization of the moving vehicles additional application was developed. A thread class was implemented to display the motion of a Taxi on the current map display. This thread interacts through a JDBC to retrieve the current taxi speed from the speed information currently set on the network link it is navigating. In a real time system this would be received via GPS. In this example, this is not feasible, and it is achieved through simulation. The thread also implements the Dynamic Shortest Path algorithm described earlier by performing new shortest path calculations when the Taxi reaches a new node on the given route. At every position on the route this thread interacts with the main display of the applet to update the dynamic taxi information.

Traffic Congestion Simulation

To simulate traffic congestion, the lower right panel of the Applet was added to provide an interaction with the congestion model tables. This panel selects the street names from the street topology table. The user is provided with the ability to change the street speed parameters for that particular street. The streets geometry theme is set up to color code streets according to whether the traffic is moving at high or low speed rates. Because, in our implementation the network is a view on the geometry layer and the cost is a function of the speed and length fields of this table, the network is being dynamically updated. This provides the taxi thread with the ability to select a more optimal route from its current location if it is traveling down a street that suddenly becomes congested with traffic.

RESULTS

The result of this research is an implementation of the prototype that can dynamically display the velocity of the traffic in the street network for a whole City (see Figure 5). Furthermore, using the developed prototype, individual vehicles (taxis, buses, etc.) can access the dynamic network database to obtain updated optimal path information as the traffic situation changes. The following images depict the dynamic route selection provided by the application implementation.



TAXI DISPATCH

From Node: 39723 # of passengers: 2

To Node: 38745

* Right Click Map to Set *

Dynamic TAXI Information

Estimated Distance: 3.49 km

Distance Travelled: 0.03 km

Estimated Time: 4.24 minutes

Ellapsed Time: 2.43 seconds

Current Speed: 50.0 km / h

Estimated Cost: $ 6.24

Traffic Congestion Simulation

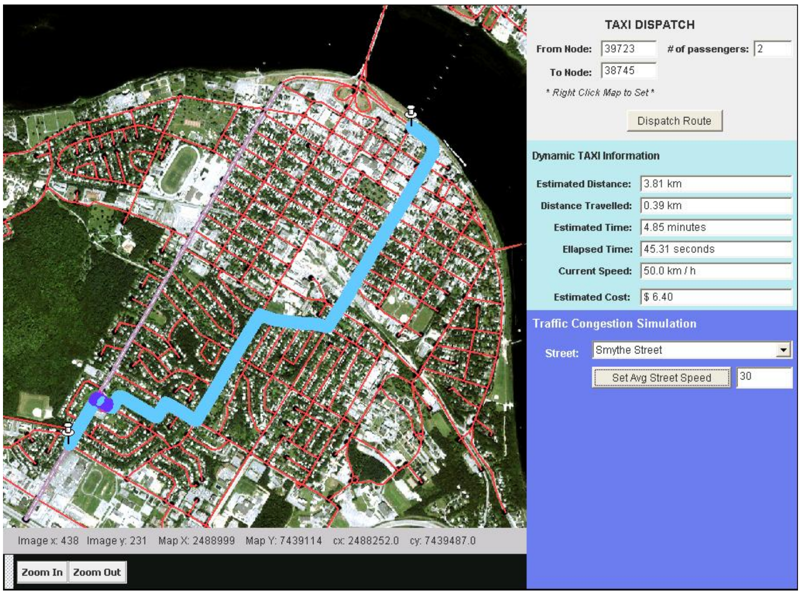
Street: Smythe Street

Set Avg Street Speed: 50

Image x: 508 Image y: 287 Map X: 2489378 Map Y: 7439417 cx: 2488252.0 cy: 7439487.0

Figure 6: Initial Taxi Route

In Figure 6, a taxi that has been dispatched down Smythe Street of Fredericton, New Brunswick has been shown. The estimated distance is 3.49 km, estimated time is 5.24 minutes, and the estimated cost is $ 6.24. This initial estimate was made without any traffic congestion on Smythe Street. To simulate congestion while the taxi is on route the congestion panel is used to set the current speed on Smyth Street from 50 km/h to 30km/h. The taxi thread adjusts when it reaches the next node on the route and a new Dynamic Shortest Path is resolved as depicted in the following image (shown on Figure 7).


TAXI DISPATCH

From Node: 39723 # of passengers: 2

To Node: 38745

* Right Click Map to Set *

Dynamic TAXI Information

Estimated Distance: 3.81 km

Distance Travelled: 0.39 km

Estimated Time: 4.85 minutes

Ellapsed Time: 45.31 seconds

Current Speed: 50.0 km / h

Estimated Cost: $ 6.40

Traffic Congestion Simulation

Street: Smythe Street

Set Avg Street Speed 30

Image x: 438 Image y: 231 Map X: 2488999 Map Y: 7439114 cx: 2488252.0 cy: 7439487.0

Figure 7: Adjusted Taxi Route

As depicted by image shown in Figure 7, the route has been adjusted along with the route particulars. The DSP implementation has found that it is more efficient to leave Smythe Street and navigate other streets in downtown Fredericton to reach the destination.

CONCLUSIONS

In this paper we presented the methodology for dynamic visualization of traffic velocity in urban areas. We developed a spatial data model that contains information about current velocity of traffic in urban transportation network. This spatial database can be used to dynamically visualize the changes of traffic within the urban environment. The prototype we developed can also be used to model traffic congestion problems. We have developed a solution of Dynamic Shortest Paths *(DSP)* within the framework of a Web Geographic Information System. The implementation, as explained earlier, is not one performed by reoptimization but one performed with the mechanisms available through the Oracle 10g database and could prove to be very useful if integrated with real time technologies *(i.e. GPS tracking and Traffic Monitoring devices).*

This solution has been presented in the form of a Taxi Dispatching system. It is noted that this sort of routing algorithm is important to all domains of urban transportation systems that are interested in dynamically tracking the shortest time to a destination based on traffic and other travel restrictions.

Bibliography

Grier N., Chabini L., 2002. A new approach to compute minimum time path trees in FIFO time dependent networks. In *Intelligent Transportation Systems, 2002.* Proceedings of the IEEE 5th International Conference, 485-490. IEEE 2002.

Kothuri R., Godfriend A., Beinat E., Pro Oracle Spatial, 2004. Apress, ISBN: 1-59059-3839.

Oracle Database 10g: Oracle Spatial Network Data Model. 2005. An Oracle Technical White Paper. May 2005.

Aloisio K., Cruz R., Marcus L., Whitlow H., 2002. Accessing Multi-Modal Transportation Accessibility Using ArcView Network analyst, ESRI User Conference 2002, http://gis.esri.com/library/userconf/proc02/pap0185/p0185.htm.

Sobek A. D., Miller H. J., 2006. U-Access: a web based system for routing pedestrians of different abilities, J Geograph Syst, 2006, 8: 269-287.

Nielsen O. A., Israelsen T., Nielsen E. R., 1997. GIS-based method for establishing the data foundation for traffic models, ESRI User conference 1997, http://gis.esri.com/library/userconf/ proc97/proc97/to600/pap573/p573.htm.