Economical LBS for Public Transport: Real-time Monitoring and Dynamic Scheduling Service

Jianghua Zheng[1,3,4], A.C. Winstanley[2,4], Lei Yan[3], A. Stewart Fotheringham[4]

1. College of Resources and Environment Science, Xinjiang University, Oasis Ecology Key Lab of National Education Bureau, Urumqi 830046, Xinjiang, China, itslbs@126.com

2. Department of Computer Science, National University of Ireland, Maynooth, Co.Kildare, Ireland.

3. Institute of RS&GIS, Peking University, China, Beijing 100871, China

4.National Center for Geocomputation, National University of Ireland, Maynooth, Co.Kildare, Ireland.

***Abstract*—Public transportation plays an increasingly important role in building sustainable cities. However, a major deterrent to using public transport is the uncertainty of the length of waiting times at stops. This research suggests a solution based on location tracking based on an example of the use of tram transportation in Blackpool, north-west England. It utilises two mobile services: one involving real-time monitoring; the other involving dynamic scheduling. The former is intended for use by public transport managers. The later is intended for use by the general public. Both provide information on the current location of the next tram and its estimated arrival time. The system, TramInfo, was built by PHP and AJAX and the embedded application software was developed by J2ME based on Symbian OS. Future work is also discussed.**

***Index Terms*—location based services, J2ME; public transport, dynamic real-time scheduling**

I. INTRODUCTION

Public transportation is being given increasing attention by transportation administrators and by travellers due to the increasing volumes of traffic in metropolitan area and the high cost of gasoline. However, long waiting times at public transportation stops, such as a bus or tram stop, is a common deterrent to the use of public transit in many cases. A standard solution is for transit operators to provide a timetable of services at each stop but this has the obvious disadvantage of not being reliable in the event of traffic disruptions or heavy congestion. It would be highly beneficial to the travelling public if they could receive dynamic scheduling information for any stop on the route. This paper explains how this can be done in an efficient manner based on a prototype set up on trams in Blackpool, England. Although similar systems have been developed elsewhere, we argue that our architecture has advantages over previous systems.

II. SYSTEM STRUCTURE

The server is developed with PHP and AJAX, the latter being widely used since the advent of Google Earth. It provides seamless web services, especially the display of remotely sensed images and other spatial data. The application also provides a web monitoring service (see http://www.cs.nuim.ie/~adamw/tramtracker). The system structure is showed as Fig.1.

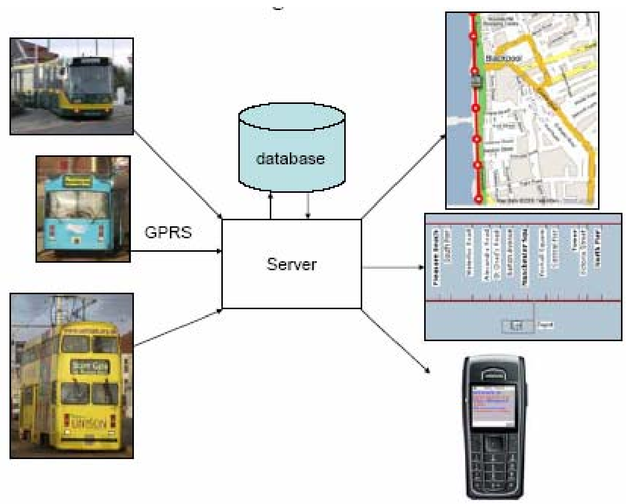

Fig.1 System structure

In the paper, the mobile service part is mainly discussed. The system has three levels. The process procedure is shown in Fig.2.

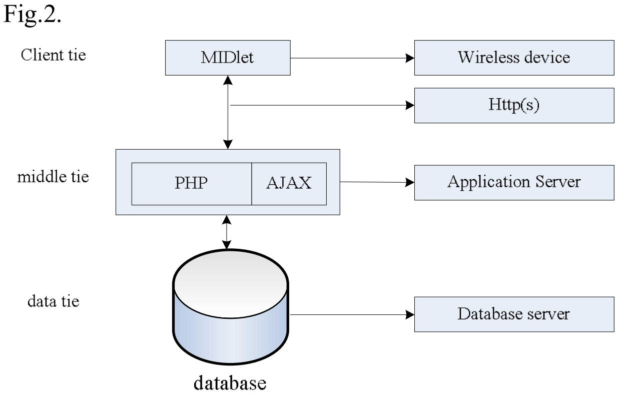

Fig.2 software architecture

III. MOBILE OPERATING ENVIRONMENT

Various mobile devices have emerged recently with the advent of advanced geotechnologies. Mobile devices are competing equally with PC's with the availability of miniaturized hardware and compatible software. In the mobile computing environment, any operating system suitable to the

978-0-7695-3177-9/08 $25.00 © 2008 IEEE
DOI 10.1109/GPC.WORKSHOPS.2008.47

mobile terminals must be compact with low memory usage and must at the same time support an efficient power management framework along with real time communications and other telephony protocols. There are many mobile operating systems which meet the above requirements. Some of these are Symbian OS, Windows Mobile, Linux and Palm OS. Symbian was chosen as the operating system for this application because of its open architecture, extensive documentation, and excellent developer support through Software Development Kits (SDKs).

A. Symbian OS & Series 40TM Platform

Symbian OS is a 32 bit open architecture operating system which meets the requirements of this project in terms of size (memory), low power consumptions and low cost. Even with the minimum requirements it provides very good API (Application Programming Interfaces) for programmers along with integration to peripheral hardware, thereby allowing programmers to achieve innovative and unique designs.

Symbian OS has several versions for different application goals. Fig.3 shows its classifications by application goals.

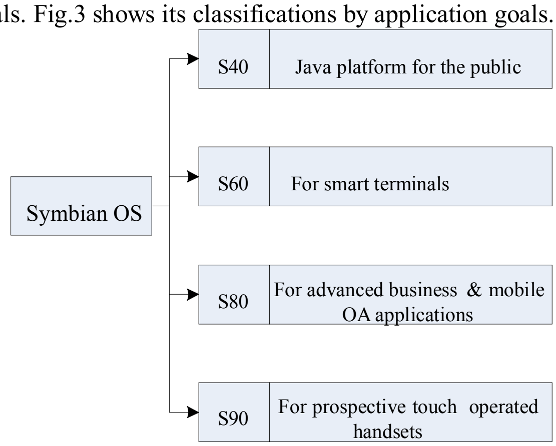

Fig.3 versions of Symbian OS (S is instead of Series.For example, S40 is instead of Series 40TM)

S90 is relative new. S40, S60 and S80 have their upgrade versions. Since one of the two goals of TramInfo is to provide tram information service for the public, S40 was selected as the embedded OS for mobile terminals to serve more people with low cost hardware. Nokia once defined S40 as the JAVA mobile phone platform for the public. Up to now, S40 has four versions. The new version is the 5th released in 2007. It is interesting that a 4th version does not exist. S40 supports 65536 screen colors and WAP2.0. It also provides support for JAVA MIDP2.0 (Mobile Information Device Profile), CLDC1.1 (Connected limited Device Configuration), and XHTML (Extensible Hypertext Markup Language). It can meet the needs of mobile communication. It also has an enhanced 3D image engine supports screen resolution to QVGA (Quarter VGA, 240×320). It is suitable for LBS applications.

B. Software Development Kits (SDKs)

The SDKs consists of all the tools required to develop real-world applications that include an emulator, target compiler, example applications, and extensive documentation. The S40 platform has a large number of resources including SDKs, white papers and discussion groups. NokiaTM provides SDKs that are compatible with popular Integrated Development Environments (IDE) of JAVA, such as Borland Jbuilder and Eclipse. The selection of an SDK and associated programming environment is dependant on the target model of the S40 phone.

For TramInfo, we selected Nokia S40 v3.0 SDK and Borland Jbuilder 2005 on the emulator to develop our application prior to downloading and testing on the Nokia 6070. The SDK can be obtained from the Nokia Forum site [3], and provides detailed installation instructions. The SDK also requires the JAVA Runtime, and it must be installed and tested prior to installation of the SDK. We have used jdk1.5 as the JAVA Runtime System. The SDK is a complete development environment and its key features include a cell phone emulator, example applications, extensive documentation, tools to compile projects for the emulator and also installation files required by a real cell phone. We started by building and running example applications provided with the SDK to ensure the capabilities of the emulator.

C. Emulator implementation

The emulator provided by the SDK is S40 compatible and closely emulates the operation of a real cell phone. The PC mouse is used to interact with the emulator whereas the cursor keys are used to navigate through the displays. The emulator very closely simulates the operation of a real mobile phone up to the point where an entire application may be designed, coded and tested without an actual mobile device. The code written in JAVA is debugged and compiled in the IDE. After finishing debugging and compiling, we obtain a MIDlet application program. This still needs to be obfuscated for code security before release on a mobile phone. The file's postfix is jar, which also reduces the size of the project. The application can be installed on the real mobile phone by wireless or cable download.

IV. MOBILE INFORMATION SERVICE SYSTEM

A. *system structure for mobile information service system*

A mobile information service system provides real-time and dynamic transport positioning & scheduling services for the public. In this expanding field, attention has been paid to several key aspects, such as the system architecture and data collection for travel-time computing.

What kind of system architecture is most suitable for a LBS application? There is no definite answer. Some researchers summarized various LBS solution and divided the applications into two types by functions of mobile terminals.: *thin* or *fat* client and server. Both of them can use the same physical structure as in figure 4.

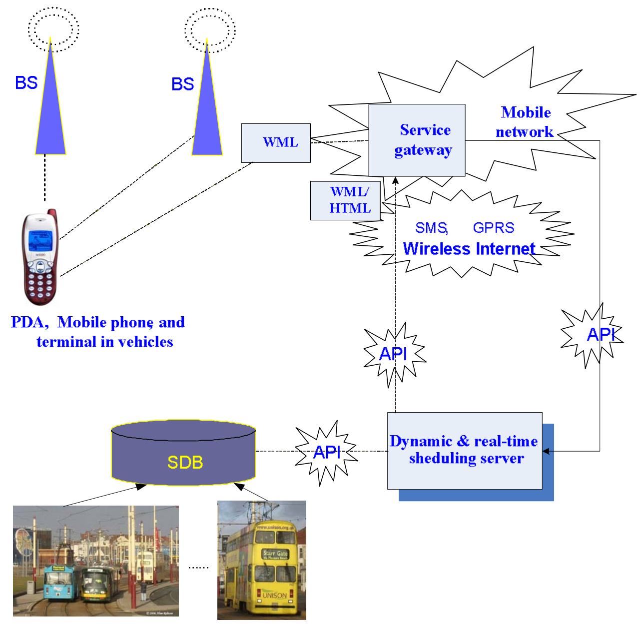

Fig.4 system physical structure

For thin client implementations, the functions of mobile terminals divide into two aspects: the interface for the user to send a request and the interface for the display of results. All the data and the core computing are on the server. This type can be categorized as being in centralized service mode.

The second type has a fat client, meaning the core computation is performed in the terminal. Most of the data is also held in the terminal. The server is used only to collect real-time data and broadcast it to the mobile terminals. This type is also characterised as self-service mode by some researchers.

Table below just show some features of the two types.

TABLE I
FEATURES OF SELF-SERVICE MODE AND CENTRALIZED SERVICE MODE [2]

items	self-service mode	centralized service mode
place of data storage	local storage at terminal	centralized storage at server
place of core computing	at terminal	at remote server,
comuting capacity of terminal	high requirement	low requirement
data maintenance & update	inconvenient for user	convenient, user doesn't take part in
safty of data	poor	good
place of realt-time processing	at terminal	at server
cost of real-time processing	higher	lower
long of response period	very fast	fast

Table I show some features of the two types and that each has its own shortcomings and advantages. Choosing which to use depends on the applications' functions and special requirements. For example, China Unicom put forward NaviStar, a personal navigation service in 2004. It adopted a centralized service mode for three major reasons: it is easy to support dynamic navigation based on real-time information of traffic flow, it is easy to carry out data maintenance and update, and lastly (but not the least) is the Chinese government's restrictions on users obtaining high-resolution spatial data. In this case, the mobile terminals just function to send their requests to the server and receive and display the result on the terminals. For location-based services for public transport, it is also suitable to adopt a centralized service mode for the system structure. The following case study shows how we formed an economical LBS system for a public transport system.

B. *Case Study: TramInfo for tram system in Blackpool UK*

We implemented a prototype mobile information service system, *TramInfo*, for making available dynamic tram information for the public in Blackpool UK. It satisfies major requirements of public transport travellers. These can be expressed as:

- Where is the nearest tram at present?
- How long will it take the nearest tram to reach my stop?
- How long will the tram take to get to my destination?

The system needs a lot of data, especially longitudinal data on vehicle travel times, to obtain results. We selected a centralized structure for the system, so the core computing and data storage are on the server.

a) positioning equipment on tram   b)tram in Blackpool UK   c) terminal for mobile services

Fig.5 system hardware

Fig.5 shows the hardware used by the LBS system. Here the server is not counted as LBS system hardware, as it also is part of the service system without the mobile information services. The equipment in (a) costs approximately 300Euros. Figure (c) shows the Nokia 6070 cell phone used as a general mobile terminal. It is inexpensive with powerful functions. The cable is just used for program transfer and testing. The system is an economical LBS solution for both developers. and users.

The program is developed based on Jbuilder2005, MIDP2.0 (J2ME) and the jdk1.5 environment. It operates on Symbian Series40 III (S40) operating system. The major functions are implemented in two different modes: with textual and graphical results.

Table II gives some detailed information on system modules listed in the left part of the above window. Each module's function can be seen in the right column of the table below.

TABLE Ⅱ
SYSTEM MODULES & FUNCTIONS

No.	name of class	function
1	Copyscreen.java	show copyright screen
2	Displayable1.java	Canvas
3	HttpDownload.java	connect to web & send requests
4	Mainscreen.java	main screen for user
5	MIDlet1.java	Main program
6	Parser.java	parser response stream
7	Queryscreen.java	for user to input information for query
8	resultscreen.java	to show the query result
9	Stopselect.java	stop list for selection
10	txtTask.java	thread to call HttpDownload

The work flow of the system is as in figure 6. MIDlet1.java starts Mainscreen.java. Queryscreen.java will call stopselct.java to let user choose two names of tram stops as start and destination. After this, it will activate txtTask.java to connect the remote server and send the request parameters. When the results are received by the mobile terminal, it will make the results readable by Parser.java and return to txtTask.java. Then it will start a new result screen by resultscreen.java to show the result, so a complete query cycle is complete. When we close the result screen, it returns to the Queryscreen.

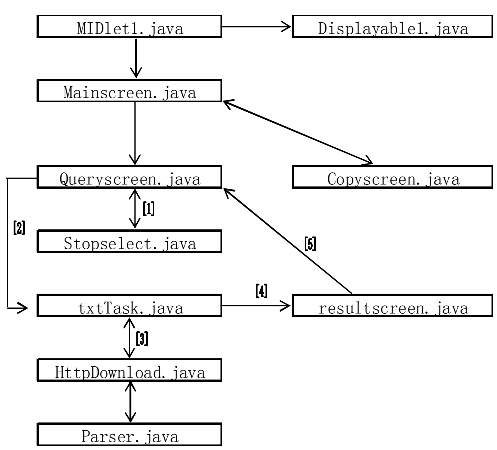

Fig.6 work flow of the system

The Fig.7 below shows the procedure on emulator. When we start the mainscreen, it provides three items, StartQuery, Copyright and ExitService. If the first item is selected, it will show you the queryscreen, which let users to choice their start stop and destination by using stopselect screen. After the choosing and with a user's confirmation, system will ask you to ensure your entering network service. If the user says yes, the result will display on the resultscreen in the next several seconds. It shows the useful information about the location of next tram, the distance between the tram and user's start stop and the time passenger should waiting for the tram. The time is calculated by the distance and average speed of tram.

Communication costs of the system are very inexpensive. Users only input the names of start and destination stops. These parameters are sent to sever by the mobile network (GPRS). The results are also simple text strings. The size of the two transfers of information is very small and so inexpensive for the users if the system is charged according to data transfer amounts.

Fig.7 real running

V. CONCLUSIONS

The research focuses on providing an economical LBS solution for public transport. Its main hardware is a positioning terminal and mobile phone. However, it provides powerful services. The server provides two types of services. One is wired-web service. The other is for wireless terminals. Both types provide not only public transport control service for the transport administrations, but also dynamic timetable information for the public. The integrated architecture follows the trend of intelligent transportation system (ITS). It also adopts the third-party service provider model using the mobile phone network. This reduces operational costs and so the scheme is a successful economical real-time LBS solution. As some users may need maps to find their way to the tram stop, the next step of the research is to realizing map representing for mobile users. This will make the system more attractive and effective.

ACKNOWLEDGMENT

The research is partly supported by a Special Postdoctoral Fellowship from National University of Ireland, Maynooth and the PhD Start-up Fund of Xinjiang University (ID: 070282)

REFERENCES

[1] Jianghua Zheng. Key Technologies of Dynamic Navigation for Vehicles under the Circumstances of Real-time Traffic Information [D]. Peking University, 2006 (in Chinese)

[2] ZHENG Jianghua, YAN Lei, LIU Yuefeng, etc. Study on Data Organization of Personal Navigation Services [J]. Computer Engineering, 2006, 32(24): 41-47 (in Chinese)
[3] http://www.forum.nokia.com/
[4] S Hoh, S Gillies and M R Gardner. Device personalization — where content meets device [J]. BT Technology Journal, 2003, Vol.21 No.1 pp67-74
[5] Cristian Darie, etc.. AJAX and PHP Building Responsive Web Applications [M]. Post & Telecom Press, 2007 (in Chinese)
[6] Sebastian Herden, Arman Mkrtchyan, etc.. Personal Information Guide – A Platform with Location Based Services for Mobile Powered E-commerce, Proceedings of the 14th International Workshop on Database and Expert Systems Applications, 2003
[7] Jae-Chul Kim, Tae-Wook Heo, Ju-Wan Kim. Ubiquitous Location Based Service, Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems, 2005
[8] Hartigan, P.F.. Transport and processing of perishable information: Essential infrastructure for location based services, Proceedings of 3rd International Conference on: 3G Mobile Communication Technologies, 2002