Personal Navigator for a Public Transport System using RFID Ticketing

Ana Aguiar and Francisco Nunes and Manuel Silva and Dirk Elias
Fraunhofer Portugal Research Centre AICOS
Porto, Portugal
Email: {ana.aguiar, francisco.nunes, manuel.silva, dirk.elias}@fraunhofer.pt

Abstract

RFID-based public transport ticketing systems rely on widespread networks of RFID readers that locate the user within the transport network in real time to be able to verify whether he can travel at that time with the ticket he holds. This paper presents a system that uses that same RFID-based location information to give the user navigation indications depending on his current location, provided that the user has indicated beforehand the places he intends to visit.

The system was designed to be cost-effectively deployable on the short term but open for easy extension. A fully working demonstrator was developed and runs in our premises. The added value of the development is the identification and integration of the necessary components into a system that delivers a valuable enhanced service using an existing infrastructure at low extra-cost and in a user-friendly way. This paper presents the details of all system components and relevant practical issues that have to be accounted for in similar enterpreneurships.

1. Introduction

The use of RFID ticketing systems for public transport networks (PTN) is growing rapidly worldwide and enjoys good acceptance among users. London, Helsinki, Shanghai, Istanbul, Moscow, Porto are some of the cities that have decided for its use. France and Holland are using RFID-enabled fares in their railway systems.

A generic architecture of common public transport RFID-ticketing systems can be seen in Figure 1. Such ticketing systems usually consist in a network of RFID readers placed at the entrance to each transport, be it underground, tram, bus or train. The users carry tickets that are passive RFID cards and can be loaded with trips or a budget at public kiosks distributed thoughout the PTN. A user passes his card on a reader at the entrance to each transport to validate his trip. The system verifies in real-time whether that RFID ticket is charged with the necessary amount of money or with the correct fare type to travel from that point.

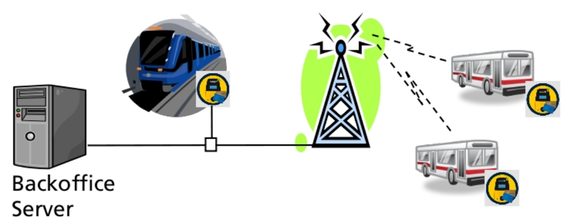

Figure 1. Generic architecture of an RFID-based public transport ticketing system.

The availability of such a geographically widespread network with real-time connectivity opens perspectives of deployment of a plethora of enhanced services running on that same infrastructure. Since each user of the PTN must carry an RFID ticket and validate it each time he enters a transport, the system knows where the carrier of a certain card is at a certain moment. So, a public transport system with RFID-based ticketing inherently can provide the real-time localisation of a card in the PTN, without requiring any change in the behaviour of a user. This paper describes a personal navigation systems that exploits this real-time geographic localisation to navigate the user in the public transport network, giving him real-time indications of how to get to his desired destination(s).

It was recognised by users in [1] that more information would greatly increase usage of public transports and route planning and real-time information were among the most desired services. These results support the relevance of this work for offering an improved mobility experience, making public transports more attractive.

Navitime [2] is a personal navigator deployed in Tokyo that considers all modes of transportations, including walking inside and outside buildings, and the whole traveling activity. However, the system requires high-end devices and broadband wireless connections that are currently not available to the majority of the population outside of Japan. In [3], a navigator to guide users during transfers in multimodal transport systems is presented, but it focuses on in-building navigation and location for transfer buildings. Furthermore it also uses high-end devices and broadband wireless connections. None of these systems is an option under the conditions that drove the design and development of the system presented in this paper.

The next section explains the specific service requirements that influenced system design and set it apart from other solutions. Section 3 explains the details of the system's hardware and software architecture and specify the interfaces between the different modules. Sections 4 and 5 describe in detail relevant parts of the system, namely the route calculation and the user interfaces. Finally, Section 6 concludes the paper.

2. System Design Requirements

The design of the navigation system was driven by a set of premises that distinguish it from other navigation solutions.

- The service should be deployable on short term, and not in a far future.
- Deployment cost for the service provider should be efficient.
- Usage cost should be low considering currently common european communication costs.
- Service should be easily adaptable and extendable to a fast changing reality.

These design premises were mapped to system requirements that drove design decisions throughout the design and implementation of the system demonstrator.

- Low deployment cost implies using as much as the existing infrastructure as possible and keeping changes and additions to the existing system as low as possible. It also means using as interface a device that users already carry around on them.
- Low usage costs means, currently and regarding the current costs of mobile Internet, avoiding the use of mobile data connections in the basic service version. This is especially important for roaming customers, as visitors to the city are one target user group.
- Flexibility of deployment and extension expresses itself as requirement on the software architecture. The software should be highly modular with clearly defined and well-known interfaces. This is very relevant for enabling extensions to the system and enabling the outsourcing of whole functionalities to different service providers.

Next to deployability under current conditions, another concern had a major influence in system design: user friendliness. The system should require as little interaction from the user as possible and little change of his habits. Furthermore, all interfaces between user and navigation system should be carefully designed so that users unfamiliar with the system or with technology in general feel comfortable using it. Finally, being user-friendly also implies using technologies that most users are already familiar with, so that they do not have to acquire any new device or learn to handle one.

A navigation system complying to these design requirements enhances the experience of users of a PTN in a cost effective way. Visitors and sporadic users are target user groups, as the system is especially helpful for people unfamiliar with the PTN. But also normal users would profit, being guided to destinations out of their current and known parts of the transport network, for example visiting a place for the first time in an area they do not usually use. In this way, the navigation system enhances the urban mobility experience and makes using the PTN more attractive to people unfamiliar with it.

3. System Overview

We start by presenting the typical usage scenario and then explain how it can be technically achieved.

3.1. Usage Scenario

The typical usage scenario is that the user tells the navigator service where he wants to go through the web interface. This can be done either by clicking on a map, by entering an address, or by choosing the destinations from a list of relevant points of interest, e.g. touristic attractions, public offices, universities, hospitals. Then, the user inputs his mobile phone number and passes his RFID ticket on a reader. After that, the user starts on his journey.

Whenever the user validates his ticket at the entrance to a transport, he receives an SMS on his mobile phone with indications on how to get to one or more destinations. Actually, the user always receives enough information to reach the next ticket validation machine. The information is relevant in the context of the public transport network, i. e. the user is told on which station to get off the current transport, which destinations he can reach from there, in which station to catch the next transport, and which transport to take next. An example SMS can be in Figure 2.

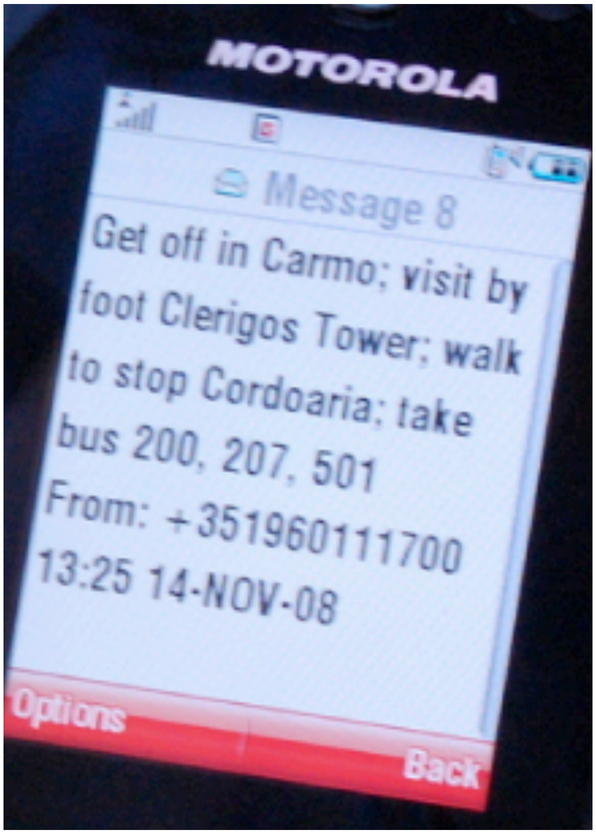

Figure 2. Example of an SMS: where to get off the bus, what to visit there by foot, the next bus stop to go, the next bus to catch.

3.2. System Architecture

The system designed to provide the service described is depicted in Figure 3. Although the navigator is an enhanced service on top of the PTN with RFID ticketing, we opted for an open system architecture that enables different necessary services to be provided by different providers and, ultimately, even the navigation service can be operated independently of the PTN. In the latter case, the PTN operator must agree to make the necessary information available. Currently, some PTN operators have made APIs available on the Internet to enable developers to access not only transport routes and schedules but also real time information on the transport, like expected arrival time at a certain stop [4]–[6], so this is not unreasonable.

Besides information on the bus schedules, it is also necessary to have a geographic information system

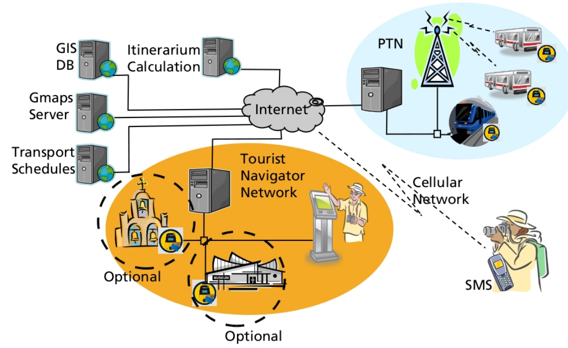

Figure 3. Architecture of the navigator service for a public transport network (PTN).

(GIS). Only geographically indexed points of interest or streets are supported destinations, so it is necessary to have the geographic coordinates of the PTN stations and reverse geo-coding for addresses. However, this information must not belong to the navigator service operator. It can be provided by specialised service providers that offer those services online through a known (or agreed) interface. The itinerary calculation, a core module of the service, must also not be implemented by the navigator service provider. Finally, even the maps used on the graphical user interface (GUI) can be provided by an external entity. In the prototype developed, the Google Maps API from Google [7] is used for geo-coding, maps and display of transport routes. The itinerary calculation module was developed as part of the demonstrator and will be described in Section 4.

3.3. Software Architecture

The software architecture for the service is shown in Figure 4. The use of Web Services Description Language (WSDL) [8] with SOAP provides an open system architecture with clear, well-defined interfaces between modules. This enables the functionality of the modules to be provided by different specialised external entitities and also for easy replacement or extension of each module. Furthermore, the use of a software architecture centred on web services offers easy platform portability. In this way, the graphical user interface developed for the kiosk is also available on the web, on a PC or even on a mobile device—a very relevant issue for user-friendliness.

The kiosk runs a web browser to render the route planning interface. This would be enough if it were not for the need to read the id of the user ticket card to associate it with the list of destinations that the user

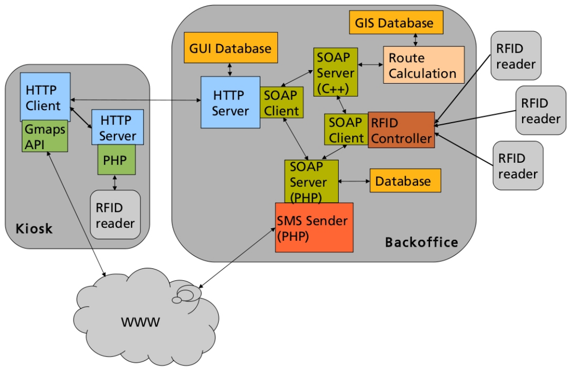

Figure 4. Software architecture of the navigator service prototype.

wishes to visit. This requires an HTTP server to run on the kiosk so that the card id number retrieved by the driver of the RFID reader can be passed to the Asynchronous Javascript and XML (AJAX) interface through a PHP script. Since Javascript does enable access to hardware and PHP scripts are server-side scripts, the installation of the HTTP server on the kiosk was unavoidable. Nevertheless, the kiosk runs only very few software modules so that the maintenance effort on the distributed infrastructure is limited.

The backoffice inter-connects the different modules necessary to provide the navigator service and remote procedures are handled by SOAP calls. The route calculation service is offered through a C++ SOAP server that implements the algorithms described later in Section 4. The itinerary calculation requires data from a GIS system, namely walking distances between points to visit and closest PTN stations, which were obtained from the Google Maps API for the demonstrator, as an example for an external service provider. In the demonstrator, the information about bus schedules was embedded in the database by hand, however this would not be the case in a real world scenario. In that case, a new module would be required at the Backoffice that would gather all the necessary schedule and GIS information, also preferably over SOAP interfaces, and put it in the format required by the route calculation. This preparation consists in transforming the GIS data and bus schedules into a weighted graph on which the best way for the user to visit his desired destinations can be calculated.

3.4. RFID Ticketing-Based Localisation

As mentioned previously, localisation is an intrinsic feature of RFID ticketing for PTN. The readings from the network of RFID readers distributed throughout the PTN are transmitted to the navigator system through a SOAP module. Everytime a card is validated on a transport, the RFID controller sends the card id read and the identification of the reader where it was validated to the RFID controller in the backoffice. These calls are the actual positioning system, as they locate the user inside the PTN network. Upon reception of a card reading, the controller calls the service that calculates the next steps of the user, automatically generates the corresponding SMS text and then calls the SMS sender service.

If the service is offered by a provider external to the PTN operator, the latter must make the card reads available. I. e. the PTN operator must expose the card id and the location where it was read to the navigation service operator, at best through a SOAP remote call.

4. Itinerary Calculation

The calculation of the best path between the destinations the user wants to visit is a core functionality of the navigator service. Currently available itinerary services of PTN operators require the user to list the stations of the PTN he wants to visit and then shows the user the best path between every two of those stations. However, this is not a friendly interface, because users often do not know which station is closest to where they want to go and also because often the user wants to visit more than one destination. So, the system we propose takes another approach. The user enters the actual destinations he wants to visit and the system calculates the best itinerary to visit them all, including the corresponding stations, the order in which they can best be visited and the sequence of transports between them. This section deals with the problem of finding the best path to visit several points of interest in the network, considering both the public transports and the walking distances between the stations and the actual desired targets to visit.

The stations of the PTN are nodes in a weighted graph, with edges representing the possible connections between them whose weight is the travel time. The points of interest are also mapped as nodes in the graph, connected to station nodes and between them by edges whose weight is the corresponding walking time. Average travel times between nodes (stations) of the PTN were obtained from the PTN. Walking times between geographically indexed nodes were supplied by an external entity, e. g. Google Maps API in our case. Average waiting times for each transport connection were calculated from amount of transports and their frequencies between every two nodes. This can be calculated offline, as it does not change frequently, and is considered a constant in the calculation of the fastest path.

At first glance, the problem of finding the fastest route that travels through a subset of nodes in a graph might be solved in two steps. First, the best paths between every two nodes in the subset is calculated. Then, a graph with only those nodes is formed and the best route found. However, the large number of nodes in a network limits the applicability of most theoretical approaches to the case of a PTN. So another solution had to be found that enables the calculation of the all pairs shortest paths in real time. We took an hierarchical approach, as is suggested in [9], which enables optimal shortest paths to be calculated in real-time. The total graph is divided in zones and the shortest paths between all nodes in each zone are calculated, followed by the shortest paths between the boundary nodes of all zones. Since the nodes of a PTN are fixed, these optimal paths are pre-calculated offline using the A* [10] algorithm instead of Dijskstra [10] for better performance. The destinations chosen by the user are the only nodes of the graph that are not known in advance and the shortest paths between them and the PTN nodes must be calculated in real time.

After the desired subset of nodes to be visited is known, the shortest paths between all of them are calculated. These paths consist of one or more edges of the original graph, passing by one or more stations, with or without transport changes, so one path can consist of more than one transport. Our algorithm takes this into account and adds a fixed waiting time each time that a transport change takes place, so that paths where a single transport is taken are preferred to paths with changes of transport.

After calculating the shortest paths between the nodes to be visited, a new graph can be built containing only the nodes that the user wants to visit and the edges are the best paths between them (calculated in the previous step). The problem now is to find the best way to visit all the nodes in this new graph, which is equivalent to finding the shortest spanning tree [10], i. e. the minimum weight tree that contains all the nodes in the graph of the nodes the user wants to visit. Because the calculation of the shortest spanning tree was taking too long for a real-time service, a branch and bound algorithm was implemented to reduced the computation time.

The procedures described here guarantee that the shortest paths between the nodes to be visited can be calculated online within times that are acceptable to users as real-time, guaranteeing the usability of the system.

5. User Interface

The user interacts with the systems in two ways: initially he communicates the destinations he intends to visit to the system, and during his trip he receives the navigation indications. The first occurs over the Internet on the user's computer or mobile device, or on kiosks placed at PTN stations or other points in the city; the latter occurs per SMS, with text messages being sent by the system to the user to guide him in real time.

5.1. Kiosk and Web Interface

AJAX is used to provide an intuitive interactive interface that is platform independent and, thus, offers the same visual aspect on the kiosk and web. The only differences in the available possibilities are due to the fact that the hardware available at a kiosk cannot be assumed to be available at home.

The kiosk has an RFID reader and a touchscreen as only peripherals. The interface is developed especially for the touchscreen, having big buttons with clear functions placed consistently, making them easy to use even for newbies. There is a set of pre-defined routes with different durations for tourists that want to check out the city highlights fast. But there is also the possibility to personalize the destinations. The user is then prompted for his RFID ticket card and for the phone number and receives the first indications after a final confirmation screen.

At home, the user can program a route in a similar way, but he can only associate it with his phone number. The exit screen directs him to a kiosk to associate his route with his RFID ticket. At the kiosk, the user can load his pre-programed route by entering his phone number, he can edit it if necessary, and finally is prompted to associate the route with his RFID ticket.

Both interfaces have been tested by several people by now, most of them fully unfamiliar with the project, and has been found intuitive and easy to use. Of course, the current interface is already the result of several improvements motivated by user suggestions to usability improvement. Unfortunately, due to the limited space, it is not possible to show here images of the web interface developed, but it can be visited at www.naviporto.projects.fraunhofer.pt.

5.2. Navigation Intructions per SMS

Although the ubiquitous availability of broadband wireless communications would enable a more sofist-icated interface, SMS were chosen because they are familiar to most people and represent no additional cost even for roamers. These facts are important since city visitors are a main target group of the system.

Each time the user passes his RFID ticket in a reader at the entrance of a transport, the answer SMS is automatically calculated by a service adjacent to route calculation. First, it is verified whether the user is where he is expected to be and the next destination to be visited is determined. If the user is where the system expects him to be, the route to the next unvisited destination is calculated. If the user has lost his way, the whole route is re-calculated with the current station as starting point and all unvisited destinies as list of destinations. Then, the user is guided as if he were not lost.

Each time, after the system knows the route to the next destination, the SMS text is automatically built, containing the necessary information to guide the user to the next ticket validation. That information is:

1) the station to get off the current transport;
2) the destinations to visit by foot next to that station, if any;
3) the station to take the next transport, if different from the one to get off;
4) the next transport to take.

With this information, the user knows what he needs to visit the places he desires and reach the next RFID reader, that functions as the "positioning" device and triggers the next navigation information.

6. Conclusions and Outlook

This paper presents a novel enhanced navigator service for public transport networks that use RFID ticketing. The service presented takes advantage of a side-effect of RFID-based ticketing: a positining system within the transport network. To our knowledge there is no other service taking advantage of this intrinsic hidden feature of RFID ticketing systems. The navigator service described is an added value service that can be deployed at low extra cost on top of an existing infrastructure, thus supplying a further argument for the case of RFID ticketing in transport networks.

The value added by the real-time navigation in the PTN can motivate non-users or sporadic users to use public transports more often, helping to induce mode shifts from other modes, like using the own vehicle. Furthermore, increasing the number of PTN users leads to increased revenues of the PTN operator.

Several issues remain to be studied or improved, among others replacing the need for the web server in the kiosk (e.g. with Google's Native Client), development of an enhanced version of the navigation indications for more demanding users, integration of the system with other means of transport or integration of real-time transport schedules and expected waiting times.

References

[1] B. Caulfield and M. O'Mahoney, "An examination of the public transport information requirements of users," *IEEE Trans. on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 21–30, Mar 2007.

[2] M. Arikawa, S. Konomi, and K. Ohnishi, "Navitime: Supporting pedestrian navigation in the real world," *IEEE Pervasive Computing*, pp. 21–29, Jul 2007.

[3] K. Rehrl, S. Bruntsch, and H.-J. Mentz, "Assisting multimodal travelers: Design and prototypical implementation of a personal travel companion," *IEEE Trans. on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 31–42, Mar 2007.

[4] H. Reed, "Cta bus tracker api - an api for the chicago transit bus system." [Online]. Available: http://ctabusapi.jottit.com/

[5] TriMet, "Transit tracking service for portland, oregon." [Online]. Available: http://developer.trimet.org/

[6] B. A. R. Transit, "Transit tracking service for san francisco, california." [Online]. Available: http://bart.gov/schedules/developers/index.aspx

[7] Google, "Google maps api." [Online]. Available: http://code.google.com/apis/maps/

[8] W3C, "Web services description language (wsdl)." [Online]. Available: http://www.w3.org/TR/wsdl/

[9] N. Jing, Y. wu Huang, and E. A. Rundensteiner, "Hierarchical optimization of optimal path finding for transportation applications," in *In Proc. Of ACM Conference on Information and Knowledge Management*, 1996, pp. 261–268.

[10] A. Drozdek, *Data Structures and Algorithms in C++*, 3rd ed. Thomson Course Technology, 2005.