NAPA : Nearest Available Parking lot Application*

Hae Don Chon Divyakant Agrawal Amr El Abbadi
Department of Computer Science
University of California, Santa Barbara
{hdchon,agrawal,amr}@cs.ucsb.edu

Abstract

With the advances in wireless communications and technologies of mobile devices, location-based applications or services will become an essential part of future applications. We have developed a location-based application, NAPA, that assists users to find a nearest parking space on campus. NAPA is an example of such applications, which combines a number of new features such as location-based, wireless communication, and a directory service like LDAP.

1 Introduction

A location-based application is an application that functions differently depending on the current location of its users. For example, when a user wants to find a restaurant, he is not concerned with restaurants that are far away from his current location, rather he wants to choose one from several restaurants near his current location. In this case, the application should get the user's current location information in some way and provide it to the server that would return several candidate restaurants. A crucial part of this location-based application is locating users' current location. Global Positioning System (GPS) is a widely used technology for this purpose and it is constantly being improved: The U.S. Department of Defense has recently released a new performance standard for GPS, which will increase accuracy for civilian users from 100 meters to 36 meters of horizontal positioning accuracy. By 2008, we will get better accuracy from Galileo, a project from the European Space Agency. Galileo will guarantee 4 meters of accuracy. Even better will be from BlackJack, a refinement of GPS by the NASA, which will deliver accuracy of 2 to 3 centimeters. Mobile devices such as PDA or cell phones or combination of both are getting more popular and according to Nokia (www.nokia.com), the number of mobile devices by the year 2003 will be more than 1 billion. With the advances in GPS and wireless communications technology and the growing popularity of mobile devices, the need for location-based applications has gained significant attentions.

As a part of the digital campus project at University of California, Santa Barbara, we have developed a location-based application, named NAPA: Nearest Available Parking lot Application. Finding an empty space from multiple parking lots scattered around a campus or some area like a city or airport, can be problematic if the covered area is large and many lots are almost full. Furthermore, if a professor is running late for a class and is driving from off-campus, then it is important to find a parking space as near as possible to the class room. In addition to this feature, the NAPA server can make the process of charging parking fees automatic by contacting a directory server that the University maintains. This would reduce the amount of work being done at the campus entrance, which is often a time consuming process in itself.

2 Overview

A NAPA server maintains information about parking lots and buildings on UCSB campus. The information about a parking lot includes the maximum capacity, the number of vehicles currently parked, the type of parking permits allowed to park, and so on. There are different types of parking permit depending on the user's status: faculty, staff, student, visitor, and so on.

The users of a NAPA client application are typically at one of the entrances of UCSB and submit queries to find a nearest available parking lot. There are two options to query a nearest parking lot: one is the nearest parking lot to a destination and the other is the nearest parking lot from the current location. Users choose the first option when they have an appointment or a class at a specific building, say Building A. Naturally, they want to park at a parking lot as

*This research was partially supported by the NSF under grant numbers EIA98-18320, IIS98-17432, and IIS99-70700 and the Caltrans-Testbed Center For Interoperability (TCFI).

Proceedings of the 18th International Conference on Data Engineering (ICDE'02)
1063-6382/02 $17.00 © 2002 IEEE

close as possible to Building A regardless of their current location. They choose the other option, when they just want to park, then they might as well do it at the nearest parking lot.

The NAPA server associates buildings on the campus with parking lots in the order of distances to the building. Although it could do this on the fly, this is not necessary since the associated rank would not change often. Therefore, we precompute the distances between the buildings and the parking lots in the covered area and rank the parking lots accordingly, so that each building has a list of parking lots from the nearest to the farthest. When a user queries a nearest parking to a destination providing his or her destination and an identification, the NAPA server looks up the parking lot list associated with the specified building and finds a nearest available parking lot, in terms of the distance to the building, and the lot should be appropriate for the client's parking permit type.

With the identification the client provides, the NAPA server tries to authenticate and retrieve the type of permit the client holds by communicating with a directory server maintained by the University. If it fails to authenticate, it assumes that the client is a visitor, thus assigning a visitor permit. In this demonstration, we assume the directory server to be an LDAP(Lightweight Directory Access Protocol) server.

When a user queries a nearest parking lot from the current location providing his or her current location and identification, the NAPA server finds a nearest available parking lot in terms of the distance from the current location and the lot should be appropriate for the client's parking permit type. The client program is assumed to get the location information from the GPS receiver attached to the PDA. At this stage of the development, however, we simulate the GPS feature by letting the user touch the location on the campus map displayed on the PDA screen. This simulation is necessary for the development, otherwise we have to physically move around the campus to test the system. At the end of the development, we plan to add a GPS receiver to a PDA and get the real location information. Figure 1 shows the architecture of the NAPA.

After finding out the nearest available parking lot, the user sends the NAPA server a message that he or she has parked. Then the server updates the information about the lot accordingly. When the user leaves the parking lot, the NAPA server can automatically charge the appropriate parking fee if it is necessary. To make the application more interesting, we add a reservation feature that lets the users reserve a parking space only when they are near the campus, for instance, at most 5 miles from the campus.

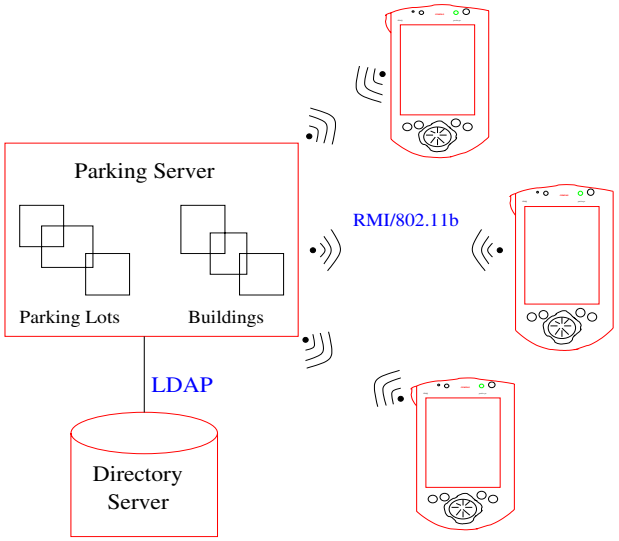

Figure 1. NAPA Architecture

3 Demonstration

The demonstration involves three components: a NAPA server, an LDAP server, and a couple of PDAs (iPAQ 3650 from COMPAQ). The two servers will reside in a notebook and the PDAs will run a client program which communicates with the NAPA server. IEEE 802.11b is the choice of wireless communication protocol between PDAs and the notebook. For demonstration purpose, we made the client program change its identity. In addition to this, the client program has features that allow its users to change their destination from a list, change their current location by letting them touch the screen on a displayed map, park and leave, query the nearest parking lot to the destination or from the current location, and finally, query for information such as the id, destination, etc. In this demonstration, we will illustrate these features as well as several other scenarios such as what happens when a lot is full or what happens when a user tries to reserve a parking space remotely. Figure 2 shows a snapshot of a client program running on an iPAQ.

Figure 2. Snapshot of running a client

Proceedings of the 18th International Conference on Data Engineering (ICDE'02)
1063-6382/02 $17.00 © 2002 **IEEE**