NFC-Forum Global Competition 2009 – **Research Track**

Project Description

EasyRide

Make your ridesharing easier !

An Ecological, Ethical and Social Ride-Sharing NFC-Enabled System

http://ricmnfc.free.fr/videos/

Université **J**oseph **F**ourier Team (France)

Abstract

Car traffic will continue being a main environmental concern for the upcoming decades. For major cities and suburbs, ride-sharing is a great way to reduce traffic density and improve social links in a population by leveraging existing cars, drivers and infrastructure. Unfortunately, it is still only marginally used. *EasyRide* is a new, dynamic ride-sharing system based on NFC technologies. It provides software infrastructure to implement the system on a metropolitan level. It has to act in concert with local public transport networks and local private taxi companies in order to ensure a minimal Quality of Service. The system is privacy friendly and will be hosted by the open-source project *OW2 Aspire RFID* (http://wiki.aspire.objectweb.org/).

Team Presentation

The project is realized by 6 Computer Engineering students of Polytech'Grenoble, helped by two professors of Université Joseph Fourier (UJF), France. It was realized in collaboration with the ADELE team of the LIG Laboratory.

- ***Student Staff:***
  Audrey COLBRANT – PROJECT LEADER (contact: audrey.colbrant@gmail.com)
  Clément COLLIN
  Elodie NOUGUIER
  Ludovic PLANTIN
  Samy SIDOTMANE
  Khalid ZAIDA
- ***Supervisory Staff:***
  Didier DONSEZ, Professor at UJF, LIG Laboratory, ADELE team (link: http://www.liglab.fr/)
  Pierre-Yves GIBELLO, Assistant-Professor at UJF, CTO of ExperLog (link: http://www.experlog.com/)

1. Motivations

The primary goal of the system is to promote, facilitate and secure car-pooling in a city area and its suburbs. Many commuters use their car for daily rides and travel alone. Most people could easily carry 3 or 4 more passengers[1]. If they did, it could dramatically reduce the quantity of cars in circulation. In many European countries, ride-sharing is largely under-used[2].

The above-stated purpose is guided by more global considerations:

- ***Sustainable development***
  - It reduces fuel consumption: fossil fuel is not an eternal energy. There will be a time when fossil resources will dry up. Limiting car rides also limits fuel consumption.
  - It reduces greenhouse gas emissions: CO2 emissions are a major global issue. By increasing the number of passengers by car, we reduce the number of cars on the road and this decreases CO2 global emissions[3].

[1] A 2008 study shows that 77% of Americans workers drive to work alone. See Burnham J. & al., *Market Analysis for RidePartner*, December 2008, http://dynamicridesharing.org/~dynami11/wiki/images/4/40/TeamForesight_Market_Analysis_v14.pdf

[2] Less than 1% of the population uses such a means of transport in France, for example. Cf. Razemon O., *Développer le covoiturage*, in *Le Monde*, September 2 2008, http://www.lemonde.fr/cgi-bin/ACHATS/acheter.cgi?offre=ARCHIVES&type_item=ART_ARCH_30J&objet_id=1049344

[3] See Kenworthy Jr., *Transport Energy Use and Greenhouse Gases in Urban Passenger Transport Systems: A Study of 84 Global Cities*, in *Third Conference of the Regional Government Network for Sustainable Development*, Notre Dame University, Fremantle, Western Australia, September 17-19, 2003, http://cst.uwinnipeg.ca/documents/Transport_Greenhouse.pdf

- ***Security / Comfort***

Decreasing traffic on motorways is a good way to make roads safer. It reduces noise, which is particularly significant in city centers. It also means fewer, smaller roads.

- ***Social links***

Many people drive alone because they simply don't know anyone they can give a ride to. Taking part in car-sharing is a good way to meet people and make contact, especially if the system provides an accurate profiling of each user. You can put people with similar interests in touch. Individuals cooperate with others to accomplish the same commendable purpose[4].

2. Definition of the approach

Carpooling is a main preoccupation in a lot of major cities in the world. In some, San Francisco for example, a car has to carry a minimum number of passengers if you want to go into the city center during rush hour. A lot of projects have been developed around this theme. Some of them are perfectly mature. Basically, their purpose is to match ride offers with ride requests. There are two major categories in these systems:

- ***Ride Planning systems:***

A website where you first subscribe and then indicate the rides you want. You have to specify the origin, the destination and your dates. It can be occasional or regular. They are often provided by institutions or firms for their employees.

- ***Dynamic Ride Sharing systems:***

These are based on the use of mobile phones. You contact a server (by SMS, phone call, 3G) to indicate you are offering / requesting a ride. When the server finds a match (passenger / ride), it calls back and gives you data about the other user. The most effective project of this kind is *Piggyback* (http://www.piggybackmobile.com/), which uses Android and provides direct matching using GPS and 3G capacities of the Google Phone.

- ***Some projects combine both approaches:***

They allow instantaneous and postponed rides. Helsinki's *Ecolane* (http://www.ecolane.com/) is the greatest example. It manages 2000 rides per day and began in 2005.

- **EasyRide*'s features:***

Keeping in mind the context described above, our project has the following features:

  - The system is a dynamic one. It aims to facilitate improvised rides in order to promote car-sharing as a common means of transportation, just like a taxi or bus.
  - The system is to work in a metropolitan context, where the issue of carpooling is more critical. The system has to be based on an existing city infrastructure in order to provide a minimal Quality of Service (taxi companies, public transport).

- ***Economical integration:***

Such a system must provide a mechanism to compensate and/or motivate both drivers and passengers. The details of that mechanism have to be handled by the entity which manages *EasyRide*. It can consist of tax reductions, savings on car insurance, free tickets for public transportation, and so on. The attachment **[A_EasyRide-Flyer.pdf]** shows an example of commercial advertisement for institutions.

[4] The Company *RideSearch.com* uses this social dimension as a marketing argument: *Less pollution, lower gas expenses, less traffic and more friends*. Cf. http://ridesearch.com/ .

3. Innovativeness of the approach

- ***Rich RFID content***

The system uses different *NFC Record Type Definitions* (RTD) and proposes new ones (external RTD). Our tags contain:

  - Local Taxi Phone number, Local Public Transportation SMS Request
  - Point Of Interest data (address, name, geographical coordinates of the markers)
  - Identifier used to launch the correct application (DriverMIDlet or PassengerMIDlet)

- ***Easy to use:***
  - **Auto-launching** of phone application: we use *PushRegistry* to start our application (the application is automatically triggered when the user touches one of our RFID terminals).
  - **No need** to specify the **origin** of the ride, only the destination. The origin is included in the tag.

- ***Privacy-friendly:***

RFID-based applications are more and more closely associated to tracking technologies. With the increase in registration forms, video surveillance devices, network logging, etc., people are becoming more and more wary of being spied on. For more transparency and trust, the entire code of our application will be open-source. Algorithms are thus accessible. There are no hidden mechanisms. Software can easily be audited and certified by third parties. *EasyRide*'s software will be hosted by the open-source project *OW2 Aspire RFID* (http://wiki.aspire.objectweb.org/) under the LGPL license. The availability of our source code allows it to be amended, which at least guarantees the opportunity to fix it when exposed to vulnerabilities.

- ***Reliability:***

Drivers are ensured that awaiting passengers are physically present (the passenger application starts when a passenger puts his NFC phone in contact with an RFID geo-localized marker).

- ***Social networking:***

EasyRide provides an accurate profiling system. Ride requests are based on users' profiles: users can fill out a form that will allow them to prioritize features when looking for drivers, allowing them to indicate the most important qualities to them (minimal distance, driving skills, language...). When soliciting a ride, the list of potential drivers will be sorted according to these priorities, speeding up and refining the selection process.

Double ranking of users: Our ranking is based on both user and system evaluation. Each passenger has to give an evaluation of the driver after a ride (driving skills, conversation…). The driver evaluates his passengers. This evaluation can be made through the *EasyRide* WebPortal. When users try to cheat the system (a passenger has accepted a ride but doesn't wait for the driver, a driver doesn't take a passenger who waited for him…), *EasyRide* detects the problem and penalize bad users (nature of the penalties is to be defined by the entity which manages *EasyRide*).

4. Process description

Below is a nominal scenario (normal situation). A video-clip shows *EasyRide* at work at http://ricmnfc.free.fr/videos/ .

- ***Step 0***:

Users (drivers and passengers) must register through our web site in order to download our phone application.

- ***Step 1***:

When a driver wants to provide a ride, he simply swipes his NFC phone in front of a RFID tag sticker on his car dashboard before he starts his car.

The application installed on his phone automatically activates and asks him to enter his destination. He can possibly select one among a list of favorite/previous rides.

The driver can then hook the phone onto a dashboard holder and begin his ride.

- ***Step 2***:

If someone wants to carpool as a passenger, he swipes his phone in front of a RFID located at the edge of a road, generally at a bus stop. Then he selects the place he wants to go to.

The system will return a list of drivers with a matching ride, sorted according to the user's preferences. He chooses the one he prefers among the list.

- ***Step 3***:

Once a driver has been selected, the system informs this driver that someone wants to be picked up. He can see details of this request (position of the passenger, passenger's profile, additional time required for his itinerary). Then he tells the system if he accepts or not.

If he declines, an updated list of drivers will be sent to the potential passenger, or else the matching step is finished.

- ***Step 4***:

When the passenger gets into the car, he swipes his phone in front of the dashboard sticker, and repeats this action when he gets out.

5. Technical solution

EasyRide is based on Java technologies. It uses JavaEE and J2ME. The attachment **[B_EasyRide-Architecture-Diagram.jpg]** shows its architecture. The users interact with the carpooling system via different components:

- ***WebPortal***

The main goal of this component is to allow users to see their rides' history and to rate other users they have carpooled with. It also enables them to subscribe to the system, to download the dedicated application to their mobile phone, and to set/modify their profile and preferences. This *WebPortal* will be accessible both via computer or mobile phone browsers.

Technologies: JSP/Servlet, HTML, CSS, Javascript.

- ***PassengerMIDlet and DriverMIDlet***

The passenger and the driver client are two parts of the application located on users' mobile phones. They

will be run by the users to find someone to carpool with, whether they are drivers or passengers.

DriverMIDlet contains a Bluetooth communication module for GPS trackers. The GPS could be directly embedded in the NFC phone in the future. Passengers don't need GPS because they use NFC markers to be localized.

Screenshots of the midlets are available in attachment **[E_EasyRide-Midlets.pdf]**.

Midlets were tested with this hardware: *Nokia 6131 NFC Phone, Royaltek RBT 2300 GPS.*

Technologies: MIDP 2.0, CLDC 1.1, JSR 82 (Bluetooth), KSOAP2 (WebServices), JSR 257 (Contactless).

- ***EasyRide Kernel***

Main part of the system which contains these components (each component can be hosted on a separate computer to balance the charge):

  - **PassengersHandler and DriversHandler**

They handle communications between the Kernel and the NFC phones through WebServices (based on JAX-WS). They manage lists of connected users and have knowledge of their current states (awaiting a ride, driving alone, driving with a passenger…).

  - **DataManager:** It manages communications with our database. Information is stored about *EasyRide* users, and can be used by a retribution external program. It uses JavaEE 5 (EJB3, JPA, EJB-QL). The structure of the database is represented in the attachment **[F_EasyRide-Beans.jpg]**.
  - **DemandSupplier:** It does the matching between supply (drivers pool) and demand (passengers pool). It sends to each passenger who asks for a ride a list of 5 possible drivers. A driver is selected if his destination is near the passenger's one. The list is sorted according to passenger's preferences.
  - **Geocomputer:** Module in charge of geographical calculation. It can be linked to a dedicated commercial WebService to do complex operations. Provide results to the *DemandSupplier*.

- ***The EasyRide NFC tags***

To have an idea of the tags' appearance, please see the attachment **[D_EasyRide-Design.pdf]**.

  - **Meeting points**: these markers are spread around the city. They contain data which define Points of Interest (latitude, longitude, name, address) and several additional data such as useful phone numbers (taxi, public transport).
  - **Cars**: each vehicle used for ride-sharing has a RFID tag. It contains vehicle's registration and a identifier of the *DriverMIDlet* to launch when a *EasyRide* enabled phone is put in contact.

The attachment **[C_EasyRide-Process-Diagram.jpg]** is an illustration of the system at work when a passenger requests for a ride.

6. Attachments

These are the attachments available in the ZIP-file:

- **A_EasyRide-Flyer.pdf**: Flyer which shortly presents the project.
- **B_EasyRide-Architecture-Diagram.jpg**: UML Class-Diagram which details *EasyRide* architecture.
- **C_EasyRide-Process-Diagram.jpg**: UML Sequence-Diagram which resumes the process of finding a ride.
- **D_EasyRide-Design.pdf**: A view of some elements designed for *EasyRide*.
- **E_EasyRide-Midlets.pdf**: Screenshots of the NFC-phones midlets.
- **F_EasyRide-Beans.jpg**: Structure of the database used by *EasyRide*.