UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

Smart Ride Seeker (SRS) - An Introductory Plan

Sameh Abdel-Naby and Paolo Giorgini

April 2006

Technical Report # DIT-06-022

Smart Ride Seeker (SRS) An Introductory Plan [1]

Sameh Abdel-Naby [a,2] and Paolo Giorgini [a]

[a] *University of Trento, Department of Informatics and Telecommunications (DIT)*

Abstract. A diverse range of architectures and concepts was proposed by scholars within the theme of Car Pooling Problem solving. Most of these scholars have attempted to put together two major elements: the need for people to move from a place to another, and the resources used to accomplish this action. Based on the use of location and available car seats, Car sharing systems allowed a substantial number of people to share car rides. These systems would, among other advantages, rationalize energy consumption, save money, and decrease traffic jams and human stress, and eventually make a significant improvement in human life. This paper proposes the initial phase of Smart Ride Seeker (SRS), which is a Car Pooling-*like* technique for distributing resources among a community that shares the same goals. This SRS technique will be developed through a *mobile-based* application that allows the mapping of ride seekers' locations along with the locations of available cars on a graphical interface/map, giving the possibility to calculate the fastest and simplest path for both the *ride giver* and seeker to meet and fulfill their demand.

Keywords. Car Pooling, Agents, Mobile Communications, Pollution

1. Introduction

Obstacles to vehicle purchasing are vanishing, thanks to a wide range of manufacturers contributions and productions and to the vast financial support offers given by financial institutions. Also, driving as well as roads navigation are two common old difficulties that are getting increasingly tackled by technological applications. Vehicle maintenance services are currently well-offered to societies and smartly managed. Nevertheless, what remains to be unsolved thus far are the problems of pollution (with its all variants) caused by the vehicles' emissions, traffic jams, parking, human stress, and finally the depletion of non-renewable energy resources.

Car sharing is a method to reduce the usage of cars in a specific town or territory, reducing car usage helps in turn to decrease pollution and prevent some other related problems. This usually takes place by having 1) a car owner who uses his/her car to move from a place to another, and 2) another person who is interested to go somewhere along the car owner's path to destination, and at the same time the *ride seeker* is willing to share the ride cost with the car owner.

[1] Use IOS-Book-Article.tmpl file as a template.

[2] Correspondence to: Sameh Abdel-Naby, via Sommarive 14, I-38050 Povo, Italy. Tel.: +39 0461 881508-2020; Fax: +39 0461 883964; E-mail: sameh@dit.unitn.it.

This car-sharing interactions constitute a reliable means of transportation for many people in an increasing number of countries, and it is usually managed and organized through a third party website that uses a web-based technique to match requests. Indeed, different research activities were carried out by governments in different parts of the world to encourage mobility in general and car pooling services in particular [5,6]. The motivation behind this service is usually saving the cost of transportation that is often higher than other solutions.

Car Pooling Problem (CPP) can be shown in different scenarios [1]. For example, an organization could encourage its employees to share their cars, so that available cars, drivers and destinations are known in advance by all the employees approaching a ride. However, a variant of the latter scenario is when the *ride giver* is not identified in advance and accordingly the ride seekers. Another scenario, which was briefly pointed out to in the previous literature, is to have ride seekers interested to have a commuting ride and others interested in only one way direction trip: a case in which the car may be underused for half of the trip.

Variations within each scenario raise the importance of breaking down the ride management architecture into separate independent techniques to ensure optimal distribution of resources. Moreover, several changing conditions are involved in handling CPP related implementations. Consider, for example, having cars with different capacities, some cars may take four or seven passengers but others may take just two. Given all of that, an SRS application development should not be designed depending upon fixed assumptions.

Mobile phones can provide quite sufficient image processing services, and that will facilitate the achievement of our proposal concerning the SRS. In particular, a basic framework can be obtained from previous relevant literature, ToothAgent [3]. A value-added service can also be included, and to be built on top of these systems so it can encourage people to act as both car seekers and givers (i.e., credits system). The paper core reflection is concerned with implementing such a system on mobile phones throughout Agents-oriented programming, so that it can be of use for people on the run, without the need for a person at certain situation to obtain any extra equipment to participate.

2. Illustration of the Problem

Car Pooling Problem is NP-Hard [2,3], and because vehicle routing algorithm for a certain unit customer demand is similarly applied in our case. Thus, for purposes of being simple and at the same time practical, places for ride seekers to meet ride givers are assumed to be fixed. Initially, in a single city, the system will recognize only 3 stops, each of which acts as a pickup point as well as a dropping point (similar to bus stops at which buses stop either to pick up or drop passengers). Therefore, this assumption is close to real life and, in the long term, a probability to think of the same model for larger territory is applicable.

For example, in Trento, one pickup/dropping stop (PDS) will be located in front of the University welcome office in *Via Verdi*, another in front of Engineering Faculty main entrance at *Mesiano*, and the third stop in *Povo*, ahead of Science Faculty main entrance. Accordingly, for a *ride seeker* (RS) to move from downtown to Povo, s/he should walk to Via Verdi, which we expect to be convenient for him/he; and for a *ride giver* (RG) it is necessary to stopover in the proper PDS on his/her way.

Intuitively, we assume a flexibility on the part of the *ride giver* to drop the *ride seeker* anywhere along the path. Based on the prior assumptions, we could allocate graphically the pickup/dropping stops (PDS) as image or map top layer That map would reflect the actual distances between these points and, consequently, that would give us the possibility to calculate precisely the shortest and fastest pathway for RS and RG to fulfill their demands. This method can be slightly customized and modified so that it can be applied on a greater scale (i.e., between cities).

2.1. SRS Main Argument

In the Pervasive Car Pooling architecture [7], as explained in the previous literature, pervasive objects (PO), such as key chains, are tasked with gathering data about the exact trip-path taken by the system participants, and match these data among them. Building upon this step, the system shifts to the second phase, which is route the ride sharing requests properly. Absorbing the exact path taken by the *ride giver* is critical to the system process, because it follows that the *ride seeker*'s routing message is processed, a step which is quite fundamental in designing a Car Pooling architecture.

In our framework, we assume that the system will automatically assign one of the pre-defined fixed pickup/drop stops (PDS) to ride givers (RG) or *ride seeker* (RS) according to their home or work locations. For example, the user will enter his/her location during the new-user registration phase. If it is Trento, then Trento PDS will be automatically assigned to him/her. And the possibility for a user to enter more than one location (e.g. work and home) will be given and automatically assigned to the surrounding PDSs. Moreover, the ease with which a user locates the preferred PDS will be growing up with the increase of available PDSs; hence the SRS reaches flexible and reliable architecture.

2.2. Graphics complexity concern

A common problem for routing the requests within a Car Pooling system is the difficulty to handle or understand a multipoint trip. Calculating a distance between two points is mostly constrained by the time limit. In some previously introduced architectures, ride seekers were asked to put the maximum time to wait and the preferred time to take during the trip, so finally the system avoids having any ride givers that are going to take more than the *ride seeker*'s preferred time. But sometimes the opposite takes place, which will let us understand that the *ride giver* is taking different and longer path or maybe stopping in-between the two points.

In SRS, we introduce the notion of mapping all the PDSs on a graphical interface that acts as a territory map, reflecting the real world actual distances, so the system can run a simple technique to calculate the distance between the picking up point and the delivery point and, accordingly, it recognizes whether the *ride giver* is taking the appropriate path to reach the destination. And in the second phase, the system recommends a certain request to a specific situation. On a later stage of application, proper routing messages can be reached by linking the system to maps or addresses database that would help us giving up the use of fixed PDSs gradually and rely completely on mapping users on graphical interface and calculating the distances needed to accomplish more reliable requests routing.

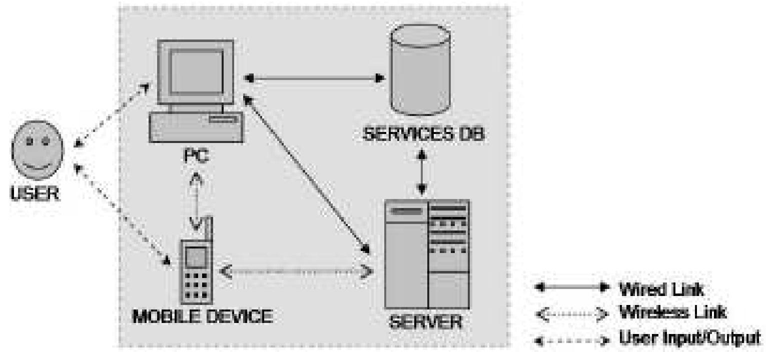

Figure 1. ToothAgents Interactions of System's Components

3. The Model

Ease-of-use is seen only in using mobile based applications that has significant influence in our daily life, a research group [8] have succeeded to combine between the returns coming out of several ride sharing techniques, demonstrating a mobile infrastructure integrations scenario, relying on GPS-based navigation, to finally form a car pooling-like scenario but public (cars does not belong to any pooling agencies) and more cost effective. But again destinations, distances, time-to-wait and time-to-spend are left freely to the negotiation process of agents. In contrast with our foreseen architecture, we are having a fixed pickup/delivery stops and standard estimate time that work as system constraints, and the user is informed with such a rule.

The idea of car sharing is originally coming out of the hitchhiking concept, as car drivers maybe pulling-over sometimes on the way home or traveling to pickup a hitchhiker and then to drop him somewhere along the way, that condition will form a car sharing situation. Two problems will arise: 1) the *ride giver* is not aware whether the hitchhiker is trustworthy or not, these are people that suddenly meet in the street without any pre-organization, and both took the associating decision just on-the-fly. 2) the cost of the ride is yet not well-defined, because the hitchhiker started his trip somewhere along the way and he will be dropped the same mode, given that the *ride giver* will not be able to calculate the entire trip cost and then divide it, and that will lead to large confusion.

At the SRS initial phase, system participants are well determined, two system actors will be located in the centralized focus of the scheme: the *ride seeker* and the *ride giver*. A car sharing system will be basically concerned about fulfilling these two entities demands as well as processing their requests. System inputs and outputs will be given and sent, generally exchanged, among these two actors, in turn, the connection between system, seeker and rider should be well-established and robust. A mobile based application was chosen for these type of human interactions, thus, standard mobile phones relations were taken as a method of communication, nevertheless, autonomous agents will be negotiating the ride details and finally communicating the final agreement with the mobile user. Similar architecture - *selling used books for student* - was described by scholars [4]. also, see Figure. 1.

Autonomous agents that take part of a Multi-agents environment for mobile based applications are well-know with its capabilities to achieve difficult organization and ne-

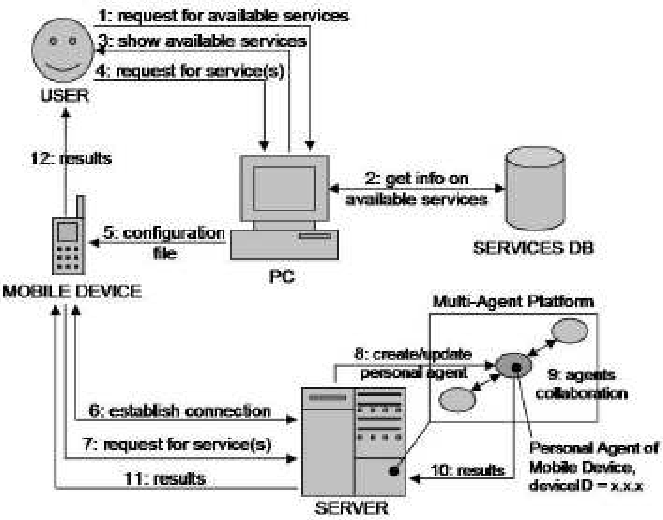

Figure 2. ToothAgents Interactions of System's Components

gotiation tasks within a certain community. Consequently, the use of autonomous agents in SRS is essential, our suggested scheme will be using an any-time algorithm suggested by scholars [9] that is expected to help us in solving the problem of tasks allocation (care ride requests routing) within a community of autonomous agents, and as a result, agents are expected to shape an alliance to better perform a certain car *ride seeker* request between multiple points.

4. SRS Scenario

In this part, we look at the SRS from three different perspectives: the ride seeker, the ride giver, and the SRS application. For each of these standpoints, we try to produce a scenario for the foreseen system activities, and finally, we presume a real life circumstances so we could convey our perception.

The RS will be going through an initial phase to select the application interface language and the action to be taken (for example, seeking a ride or offering one). By selecting the "seeking ride" option, the system will turn the RS to the next step: selection of the nearest PDS as well as the PDS at the final destination. The last step will be selecting the ride date and time within a flexible time range. All these selections will be saved in the mobile-based interface, and then sent to a managing server, either by short message service (SMS) or by Bluetooth when the user is within the coverage area. Then the user will wait for a reply containing the ride details. Notably, these same actions can be done through a computer-based application.

The RG will be going through two possible situations: the first is when s/he would like to enter to the system and offer a new ride service within a specific time range, and the second is when s/he is asked about the possibility to give a ride. In the first situation, upon entering his/her offer, the status of the RG will be always Pending during this time and s/he will be receiving messages directly from the system, through email, SMS or

Bluetooth transaction. In the second situation, when a ride seeker requests a ride at a certain time and between two PDSs that were common to be offered by a certain RG, the system would automatically send this request to the RG regardless of his/her status (for example, Pending) in the system. Of course, for referral-agent this function, the SRS has to maintain a logging and communication history, and save the users' destination and time records. At this point, a technique similar to Agents Cloning [10] can be used to facilitate recommendations of requests and referral routing.

The SRS will be supervising four major tasks. First, it will manage the request routing process and demand matching. Second, it will run a reputation system by asking a feedback from ride takers and, in turn, use this feedback to recommend future ride givers. Third, it will manage a crediting system, in which both ride seeker and ride giver will be encouraged to exchange roles. Upon registration in the crediting system, a ride seeker or a ride giver will get, presumably, five credits for free; to move from a place to another, the seeker will donate two credits to the ride giver; finally, at a certain time the ride seeker will have to choose either to start to offer rides to others so that s/he could collect credits in return or to buy credits. Fourth, the SRS will be responsible for all means and methods of communication between system agents and actors. (Can be applied similar to ToothAgent architecture, see Figure. 2)

5. Future Work

We expect to start the phase of actual development and implementation by July 2006. We see a considerable potential in integrating SRS with other services, like Public Transportations Info portals and online maps viewers, which will increase reliability and usability. That is because the system will be giving alternative transportation solutions based on the given ride PDS.

6. Conclusion

In this paper, we presented the introductory plan of Smart Ride Seeker (SRS), a car sharing system scheme, based on mobile phone application and autonomous agents. We have elaborated on a model for sharing car rides within a certain community, suggesting that both ride givers (RG) and ride seekers (RS) can interact actively with the application; an interaction that will make SRS a self-organized scheme. In addition to the above-mentioned advantages (rationalizing energy consumption, saving costs, and decreasing traffic jams and human stress) this scheme would increase Car Sharing security and service quality by: 1) using mobile phones as the base of communication between system users, 2) managing a reputation system that will act as a ride recommender for ride seekers, and 3) managing a crediting system that will encourage ride givers to share rides with other passengers.

Acknowledgements

We first thank the University of Trento - BlueAgents Research Group for the efforts it made to develop several types of mobile-based applications. In particular, we thank

ToothAgent research team and ArsLogica s.r.l. for the unabated cooperation and support given to innovative and creative ideas.

References

[1] R. Baldacci, V. Maniezzo, A. Mingozzi. 2004. An Exact Method for the Car Pooling Problem Based on Lagrangian Column Generation. *Operations Research*, 52(3), 422-439.

[2] Araque, J. R., T. L. Morin, J. F. Pekny. 1994. A branch and cut algorithm for the vehicle routing problem. *Ann. Oper. Res*. 50 37-59

[3] Letchford, A. N., R. W. Eglese, J. Lysgaard. 2002. Multistars, partial multistars and the capacitated vehicle routing problem. *Math. Programming* 94 21-40.

[4] Bryl, V., Giorgini, P., Fante, S. 2005. Toothagent: a Multi-Agent System for Virtual Communities Support. , *Informatics. Telecomunication*. Technical Report DIT-05-064

[5] D. Hirasuna, "Congestion policy: a literature review," Minnesota House of Representatives Research Dept., 1999.

[6] "Assessment of the Seattle Smart Traveler," U.S. Dept. of Transportation - Federal Transit Administration, 1996.

[7] Montessoro, P, L., DiGiusto, S., Pierattoni, D. "Designing a pervasive architecture for car pooling services"

[8] Rothkugel, S., Sturm, P. "CarPActities: Distributed Car Pool Agencies in Mobile Networks". D-54286, Trier, Germany.

[9] Shehory, O., Kraus, S. "Task allocation via coalition formation among autonomous agents". Univ. of Maryland.

[10] Shehory, O., Sycara, K., Chalasani, P,. and Somesh Jha. "Agent cloning". The Robotics Institute, Carnegie Mellon University, Pittsburgh.