RideBuddies - Multi Agent System for Ride Sharing/Carpooling

Harini Sirisena

Faculty of Information Technology, University of Moratuwa, Sri Lanka

harini.sirisena@gmail.com

***Abstract*— Carpooling or ride-sharing is the sharing of car journeys so that more than one person travels in a car. Carpooling reduces each person's travel costs such as fuel costs, tolls, and the stress of driving. This research focuses on implementing a ride-sharing solution in Sri Lanka using a multi-agent system through the communication, negotiation and coordination of agents.**

***Keywords*— Carpool, multi-agent system, route matching, journey intersection, transportation.**

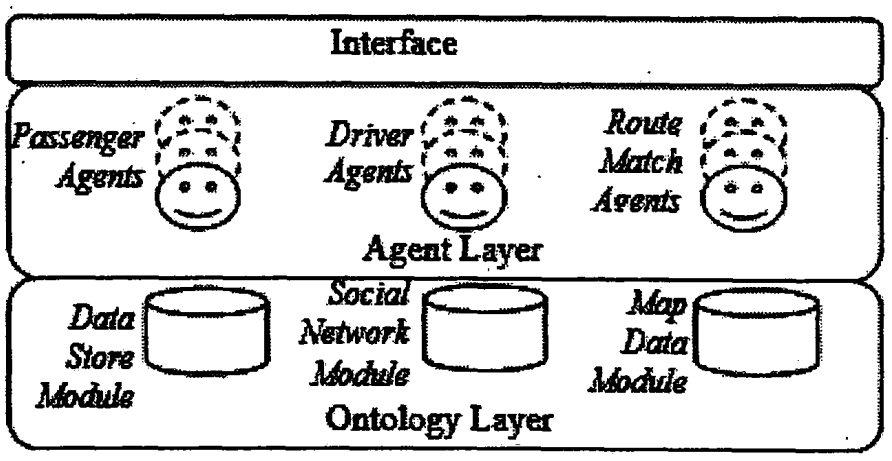
A. Introduction

Carpooling is seen as an environmentally friendly and sustainable way to travel since sharing journeys reduces carbon emissions, traffic congestion on the roads, and the need for parking spaces. An agent based solution is considered to be effective here due to the dynamic nature and complexity of the problem as it is required to match people to carpools or rideshares based on time (temporal) and route (spatial) requirements. In addition other personal preferences such as social and cultural aspects must also be considered.

B. Proposed System

The proposed system has three layers. The interface layer can be a web or mobile based interface where users can register and enter their route details. The interface layer closely interacts with the agent layer. The agent layer consists of passenger, driver and route match agents. Here a passenger agent will send calls for proposals to the driver agents. The driver agents will then contact a route match agent to check if the journey schedule, routes and personal preferences match. If a match is possible the route match agent will inform this to the driver agent. The driver agent will then reply back to the passenger agent proposing or rejecting a ride-share. The final layer is the ontology layer consisting of the map module, data store and social network module. This layer is contacted by the agents in the agent layer to get the information they need to function.

FIGURE 1
SYSTEM ARCHITECTURE



C. Route Match Algorithm

A match is found between a driver (person offering a ride) A and passenger B (person joining the ride) by determining whether their routes overlap. Consider as an example driver A, who is travelling from Moratuwa to Kelaniya and passenger B who needs to travel from Ratmalana to Colombo around the same time. Since B's route intersects with A's route (for travel via the Galle Road), if A was registered on the RideBuddies system, then B could find A on the system and setup the shared ride.

Steps in the route match algorithm:

I. Using Google Maps Directions API, encode the two routes of A and B into lists of geo location data.

II. Find the point P on A's route, which is closest to the start location of B. This will be the point where B can join A's trip. Use the haversine formula to calculate distance between points. If no point can be found within 1km proximity (threshold value), then terminate concluding routes don't overlap).

III. Taking A's route and starting from point P, iterate through B's location points from start point, and calculate the distance between the corresponding location point pairs. Each location point on A that is within 1km (threshold value) proximity to B's location point is marked as a route overlap point .When no overlap is found, it is concluded that the routes do not overlap beyond this point.

IV. Repeat above process for all possible alternative routes for A and B until longest overlap route, if any, is found.

The above algorithm can also be used to setup carpools by finding route overlaps from start to destination within a required threshold value (e.g. routes that overlap from start to destination with 5-10km difference).

D. Conclusion

This paper presented a multi-agent based solution to the carpooling problem and the algorithm used to find journey matches. The carpooling problem however has many challenges which need to be looked into when implementing a solution. The main challenges are flexibility, reliability and security. Flexibility is an issue because riders in a carpool should agree to a fixed timeframe and route and cannot differ from that afterwards. Reliability becomes an issue when there isn't a system with a 'critical mass' of participants that guarantees a match for all registered users. It then becomes near impossible to find a match for a carpool. Finally security becomes an issue when setting up carpools with total strangers. A popular solution to security issues is the setting up of carpools via connections on social media networks.