Free University of Bolzano
Faculty of Computer Science

Thesis

Dycapo: On the creation of an open-source Server and a Protocol for Dynamic Carpooling

Daniel Graziotin
Thesis Advisor: Paolo Massa, Ph.D.

October 2010

Abstract

Carpooling occurs when a driver share his/her private car with one or more passengers. The benefits of carpooling, also called ridesharing, are environmental, economical and social. Dynamic Carpooling is a specific type of Carpooling which allows drivers and passengers to find suitable lifts close to their desired departure time and directly on streets. This dissertation describes Dycapo, an open-source system to provide Dynamic Carpooling services. After a review of the state of the art, the two main "components" are described, namely the protocol and the server architecture. Dycapo Protocol is an open REST protocol for sharing trip information among dynamic transit services, taking inspiration from OpenTrip, a previously proposed protocol. Dycapo Server is a prototype providing a Web Service for Dynamic Carpooling functionalities, implementing Dycapo Protocol. Our aim with the release of an open protocol and open source code is to provide a missing standard and platform that providers of Dynamic Carpooling services can adopt and extend.

Riassunto

Kurzfassung

Contents

List of Figures

List of Tables

Part I
Introduction

Using a private car is a transportation system very common in industrialized countries. Between year 2004 and 2009, the worldwide production of new private vehicles has been of 295 millions of units [1] and, as of 2004, there were 199 millions registered drivers in the U.S.A.[2]. Road transport is responsible for about 16% of man-made CO2 emissions[3].

Private car travelling is an efficient and wasteful way of transportation. Most cars are occupied by just one or two people. Average car occupancy in the U.K. is reported to be 1.59 persons/car, in Germany only 1.05 [Hartwig, S. and Buchmann, M. (2007)]. Private car travelling creates a number of different problems and societal costs worldwide. Environmentally, it is responsible for a wasteful use of a scarse and finite resource, i.e. oil, and causes unnecessary earth pollution. The traffic caused by single occupancy vehicles causes traffic jams and hugely increases the amount of time spent by people in queues on streets. This is a unsavvy use of another scarse resource: time. Moreover, the additional pollution creates health problems to millions of individuals. Lastly, lone drivers in separate cars miss opportunities to meet and talk incurring in a loss of potential social capital.

One possible to all these problems is carpooling, i.e. the act of sharing a trip on a private vehicle between one or more other passengers. Carpooling helps the environment by allowing to use oil wisely and to reduce earth pollution and consequent health problems. It reduces traffic and - consequently - time that people spend in their cars. Carpooling has also the potential of increasing social capital by letting people meet and know each other.

Carpooling is not a widespread practice. It gained government attention in the U.S.A. but unfortunately not in EU. There are already many systems facilitating the match between drivers and passengers, most of them in form of bulletin board-like web-sites. The intention of offering empty seats of a vehicle is usually announced by a driver many days before the start of the trip. The coordination between a driver and the passengers who are candidating for sharing the trip with him/her are usually carried out by e-mails of private messages in the web-site. Therefore, we may see carpooling as a static way of sharing a trip.

The availability of mobile devices connected to the Internet opens up possibilities for the formation of carpools in short notice, directly on streets. This phenomenon is called Dyamic Carpooling (also known as Dynamic Ridesharing, Instant Ridesharing and Agile Ridesharing). Dan Kirshner, researcher in this field and maintainer of http://dynamicridesharing.org website defines it as follows: "A system that facilitates the ability of drivers and passengers to make

[1](Accessed Sept. 9 2010) http://oica.net/

[2](Accessed Sept. 9 2010) http://www.fhwa.dot.gov/ [U.S. Department of Transportation - Federal Highway Administration]

[3](Accessed Sept. 9 2010) http://oica.net/ [Organisation Internationale des Constructeurs d'Automobile]

one-time ride matches close to their departure time, with sufficient convenience and flexibility to be used on a daily basis."[4].

Curently there are no dynamic carpooling systems widely used. In fact, there are many problematic issues related to the implementation and the adoption of dynamic carpooling systems. We analyze them critically in Part II of this dissertation. While we acknowledge all aspects are critical, we claim the basic technological infrastructure is an important required and key building block. In fact we decide to focus on the creation of a solid, open and collaborative base framework for dynamic carpooling. The design and implementation of a open protocol and an open-source server are presented in Part III.

[4] Kirshner, D. (Accessed Sept 5th 2010) - http://dynamicridesharing.org

Part II
State of the art

This part contains a summary of the state of the art regarding dynamic carpooling. It is divided in three parts. In the first section there is a summary of previously published papers, in order of publication. Then we introduce a brief analysis of the deployed systems. In the last section we present the outcomes of the analysis of the whole state of the art and how we decided to move in order to provide a significant contribute in solving the problem of adopting dynamic ridesharing services.

1 Published Papers

During the research phase different papers were analyzed in order to obtain the state of the art. In this section a brief summary of each paper is presented.

1.1 Sociotechnical support for Ride Sharing

[Resnick, P. (2003)]

This paper lists barriers to dynamic carpooling adoption and possible actions to reduce them. It reports about High Occupancy Vehicles (HOV) lane - which are lanes dedicated for people doing carpoling - on streets of San Francisco and Oakland and complains that there should be no fees on bridges for HOVs. The author suggests conventions developed between drivers and passengers (e.g. pickup points near public transportation stops). Regarding security, the paper suggests to give priority to female passengers, to not leave them alone waiting for a ride. The paper reports that there are no stories about rape, kidnapping or murder and the most common reported problem is bad driving.

There are suggestions on research needed:

- Need of location-aware devices, because instant ridesharing is limited to fixed pickups and dropoff locations.
- Simple user interfaces for passengers and drivers.
- Routing matching algorithms: short window of opportunity to match passenger and driver.
- Time-to-pickup algorithms: to help passenger decide whether to use carpooling or Public Transportation System.
- Safety and reputation system design: authenticate passenger and driver before making the match, monitor arrival at destination, feedback system.

The paper discusses about social capital impacts: there is the potential for creating new social connections and also matching drivers and passengers according to their profiles creates bridging across class, race and religious views.

1.2 Pilot Tests of Dynamic Ridesharing

[Kirshner, D. (2006)]

The author presents three pilot tests done in the USA, all of them failed. The reasons of failure are the following:

- Too complicated rules and user interface
- Too weak marketing effort
- Too few users. After 1 month, 1000 flyers distributed to the public and a proposed discount on parking, only 12 users were using the system.

The paper adds the idea of saving money when parking. It also enforces the idea of using social networks to allow car pooling on the fly. The author envsions using a web – and mobile service, also introducing some interesting user stories.

1.3 The smart Jitney: Rapid, Realistic Transport

[Murphy, P. (2007)]

The paper focuses on environmental benefits of dynamic carpooling. It asserts that dynamic car sharing would lower greenhouse gas emissions in a better way than electric/hydrogen/hybrid cars would do. It introduces the idea of Smart Jitney: an unlicensed car driving on a defined route according to a schedule.

The author suggests the installation of Auto Event Recorders on cars, enforcing security. It complaints that challenges are all focused in convincing the population to use the service, proposing a cooperative public development of the system.

1.4 Auction negotiation for mobile Rideshare service

[Abdel-Naby, S. et al. (2007)]

The paper proposes the use of agent-based systems powered auction mechanisms for driver-passenger matching.

1.5 Casual Carpooling enhanced

[Kelley, L.K. (2007)]

The author considers areas without HOV lanes and proposes the use of Radio Frequency IDentificatio (RFID) chips to quickly identify passengers and drivers. Readers should be installed at common pick-up points. The paper complains that it would cost less to pay passengers and drivers for using the service than to build a HOV lane.

1.6 Empty seats travelling

[Hartwig, S. and Buchmann, M. (2007)]

This white paper by Nokia suggests to use the phone as a mean of transportation, creating a value in terms of a transport opportunity. It points out some factors limiting static carpooling, arranged via websites:

- Trip arrangements are not ad hoc
- It is impossible to arrange trips to head home from work or to drive shopping.

The paper notices that people are not widely encouraged to practice carpooling by local governments. It collects obstacles and success factors in terms of human sentences, and their solution. The authors say that the challenge is in the definition of a path leading from existing ride share services to a fully automated system.

1.7 Interactive systems for real time dynamic multi hop carpooling

[Gruebele (2008)]

The author proposes a dynamic multi-hop system, by dividing a passenger route into smaller segments being part of other trips. The author claims that the problems of static carpooling are that matching drivers and passengers based on their destinations limits the number of possible rides, and with high waiting times. Carpooling is static and does not adapt itself well to ad hoc traveling. The paper asks governments to integrate carpooling in laws and to push for its use. The author complaints that the perceived quality of service is increased even driving the passenger away from destination: a driver and a passenger should not be matched only if they share the same or similar destination because perfect matching would require high waiting times.

The paper also addresses social aspects: in a single trip with 3 hops a passenger might meet 3 to 10 people, therefore passengers may be socially matched. It suggests to link the application with some social networks like Facebook, MySpace and use profile information to match drivers and passengers.

As security improvement, the paper suggests: the use of finger-prints, RFID, voice signature, display the location of vehicles on a map, using user pictures, assigning random numbers to be used as passwords.

1.8 Instant Social Ride Sharing

[Gidófalvi, G. et al. (2008)]

The paper proposes matching methodologies based on both a minimization of detours and the maximization of social connections. It assumes the existence of a social network data source in which users are connected by means of groups, interests, etc. In such a network, the number of relatively short paths between a driver and a passenger indicates the strength of their social connection.

It provides algorithms and SQL queries. The authors assume that there is already a large scale of users, and no barriers to adoption are taken into account.

1.9 Combining Ridesharing & Social Networks

[Wessels, R. (2009)]

The author envisions a mobile and web system that interacts with social networks profiles that should improve security and trust by users. Users can register to the system in a traditional way (e.g., by giving email, username, password), then complete their profiles by linking their accounts to multiple existing social networks account, to fill the remaining fields. Otherwise, they have to fill the fields manually and verify their identity in more classical ways. The paper proposes Opensocial[5] as connection interface. An own rating system is also complained, which keeps scores of persons. Amongst the criteria are factors like reliability, safety and friendliness.

It suggests the use of mobile systems, that should make use of GPS and creation of a match on the fly (real-time algorithms). The paper provides some results of surveys: people are willing to loose 23% more time to pickup a friend of their social network rather than a stranger (6%). It also provides a high-level description of the system and implementation details.

The author asks for extra research on psychological factors that increase trust and perceived safety.

1.10 SafeRide: Reducing Single Occupancy Vehicles

[Morris, J. (2009)]

The publication is about a project in the U.S.A. It reports that there is a market-formation problem: to achieve the system that attracts passengers, there will have to be many drivers available. But the drivers will emerge only when it appears profitable or otherwise desiderable, and that depends on there being many passengers, etc. The author complaints that someone must discover a winning formula before anyone will invest.

The paper lists some interesting user stories, as well as algorithms and requirements.

[5]Google, MySpace et al. (Accessed Sept. 5$^{\text{th}}$ 2010) - http://www.opensocial.org/

1.11 Current Trends in Dynamic Ridesharing, identification of Bottleneck Problems and Propositions of Solutions

[Zimmerman, H. and Stempfel, Y. (2009)]

Reviews some papers about Dynamic Carpooling. Also reviews some applications. Identifies barriers and proposes solutions.

2 Deployed Systems

After the theoretical research, also the existing systems were taken into consideration. The following list contains the existing Dynamic Carpooling applications and some static, web-based systems that are either innovative or well-known. All the reported websites were accessed on Sept. 5^{th}2010. Each text enclosed in double quotes is cited verbatim from the website of the application.

2.1 Carriva

https://www.carriva.org/MFC/app

It is a proprietary solution using phone calls as communication system and a fixed price of 0,10€ / km. Currently it has got 1118 active users.

2.2 Avego

http://www.avego.com

It is a proprietary application for Apple iPhone. It uses GPS technologies and presents a simple, intuitive user interface. It handles costs automatically. The passengers are not required to have an iPhone. It will offer information about public transports. The application relies on a proprietary service called Futurefleet, on which no implementation details are given. On October, 10^{th} 2009 the service offered 5310 empty seats.

2.3 Carticipate

http://www.carticipate.com

Carticipate is a proprietary iPhone application that integrates with Facebook, defined as "a location based mobile social network for ride sharing, ride combining, and car pooling". It has a very simple interface looking like Google Maps mobile. According to the website, it is available on 59 countries.

2.4 Piggyback

http://www.piggybackmobile.com/

It is an Android application using a step-by-step approach (maximum one user input at each application screen) and makes wide use of graphical representations instead of text. It offers the possibility to bookmark addresses. The map screen is proprietary. When a driver and passengers are matched their compatibility is showed, represented with stars (0 to 5) and categorized as friendliness, reliability, driving skills and car. The trip cost is also showed. After the ride, the feedback system lets the user set the points for the aspects listed above. The application lets also plan rides using a static carpooling approach.

2.5 Aktalita

http://www.aktalita.com/

It is an under development application, supposed to be proprietary.

"Aktalita combines the Web, a geospatially enabled database, and a Java enabled cellphone to provide real-time dynamic carpooling between drivers and passengers"

2.6 RideGrid

http://www.highregardsoftware.com/ridegrid-dynamic-ridesharing.html

Ridegrid is another proprietary, not yet in production system. "RideGrid is a service that uses mobile internet and location technology to enable individuals to obtain rides to and from any location, spontaneously. [..] RideGrid works by dynamically combining routes. We evaluate the change required in a driver's route such that it passes through the desired source and destination of a compatible rider, and broker the agreement. We have proprietary means to calculate the routes, monetize the transactions, and introduce people to others they trust. "

It uses an internal credit system. The client has an outdated classical Java Micro Edition interface.

2.7 Project Carpool

https://launchpad.net/carpool

Carpool was the only open-source project, using PHP and Javascript. The development was stuck at the research time. The project is now closed.

2.8 GoLoco

http://goloco.org/

GoLoco is proprietary web application that also relies on Facebook. It uses a private payment system.

2.9 Ecolane DRT

http://www.ecolane.com/

Ir is a proprietary solution, web-based, focused on security. It provides a customized Nokia touchscreen device. Among the features, they declare that the device is capable of real-time data communication, reports of arrivals and departures with time information, device locking mechanisms, GPS location and direction, mileage tracking, detailed trip information.

2.10 Divide The Ride

http://www.dividetheride.com/

The project is a static, web-based solution organized around children activities. Families invite other trusted families to join their group. Groups get notifications when a ride is needed.

2.11 iCarpool

http://www.icarpool.com

This application is a static, web-based system that does not require payments. They declare to use advanced proprietary algorithms for ride matching. "High precision trip matching. helps you find the best carpool match. Find co-workers, neighbors and friends for carpool. Use for daily commute, recurring trips, long distance trips and events Plan ahead or use on-demand". Matching criteria includes social relationships, but no details are given.

2.12 Hover

http://www.hoverport.org/

It is a casual carpooling system using RFID technologies and an own credit system. The members are approved after human verfication tests. Participants must meet at a location called "Hover Park" and are identified by the RFID system. On exiting the Hover Park, the system recognizes driver and passengers and distributes credit points. There are several destination points available, that register the arrival in the same way. It also offer a guaranteed back-to-home system, by using taxis.

2.13 Flinc

http://www.flinc.mobi/

Flinc is a dynamic carpooling system using smartphones (Android and Apple iPhone). "Flinc connects navigation systems and mobile phones and arranges available seats within a few seconds - directly in the car and on the pavement. Flinc combines GPS and location-based capabilities with social networking to offer a dynamic and automated method of getting from one place to another. The service can be used on smartphones or on the PC or Mac, helping users create rides within a few seconds via an entirely automated process.". The system is currently under active development.

3 Comparative Analysis of Dynamic Carpooling Issues

The analysis of the state of the art brought some issues related to adopting Dynamic Carpooling systems, many of them recognized also by [Zimmerman, H. and Stempfel, Y. (2009)]. We categorized the issues gathered from the state of the art and their proposals in the following categories:

- Interface Design - all issues related to graphical implementation of clients and ease of use
- Algorithms - the instructions regarding driver/passengers matching problems
- Coordination - the aspects related on how to let people meet, authenticate and coordinate.
- Trustiness - the problems related on raising user confidence on dynamic carpooling systems
- Safety - the issues regarding ensuring protection of users
- Social Aspects - all the issues related to create social connections and raising social capital in dynamic carpooling systems
- Reaching Critical Mass - the problems on reaching a sufficient amount of persons using the system that would attract more other people
- Incentives - all the political, motivational and economical issues related to dynamic ridesharing systems
- System Suggestions - everything else that we consider relevant for building dynamic carpooling systems

In Table 1,2 and 3, for each category (columns) we list the suggestions and interesting points made in the different research papers (rows). Table 1,2,3 is our contribution rationalizing the many problematic issues involved in the creation and deployment of dynamic carpooling systems and in summarizing best practices and suggestions in how to deal with them.

Paper	Interface Design	Algorithms	Coordination
[Resnick, P. (2003)]	Give start and ending points and clear indications. Select what information to reveal		
[Kirshner, D. (2006)]	Lots of flexible settings		Entries made hours before the ride, more static than dynamic
[Murphy, P. (2007)]	Levels of services: - simple: just destination and pickup - groups preferences (only women etc.) - scheduling of rides		
[Abdel-Naby, S. et al. (2007)]			
[Kelley, L.K. (2007)]			One-time registation process RFID devices for drivers and riders
[Hartwig, S. and Buchmann, M. (2007)]			
[Gruebele (2008)]	Some suggestions about easy of use		Built around PQOS (ie: driving away from destination but to a location that increases PQOS, like a subway station)
[Gidófalvi, G. et al. (2008)]		Given, built around social connections. Social network needed.	Built around social connection between users
[Wessels, R. (2009)]	Simple registration system. In a second phase there is the link of social networks profiles, or manual fill Asks for a very simple UI		
[Morris, J. (2009)]		Both data structures and Algorithms for matching are given	
[Zimmerman, H. and Stempfel, Y. (2009)]	Must be similar simple and intuitive like Twitter. Parameters like “where are you going” are proposed to be as last time but modified easily. Car position is essential: drivers should get a message and just confirm or refuse a ride		Just suggestions about the legal pick up points

Table 2: Paper Grid Outcomes: Interface Design, Algorithms, Coordination, Trustiness

Paper	Trustiness	Safety	Social Aspects
[Resnick, P. (2003)]		Basic ideas authentication before the match: password / PIN monitor arrival at destination feedback system a la Ebay	Illusion of private space is ok, but is also ok to break it. DCP has potential for creating new social connections Announcing matching items in profiles before the ride is a good idea Asks for research in social capital aspects
[Kirshner, D. (2006)]		PIN created at registration on web page	Idea of using social networks to allow car pooling on the fly
[Murphy, P. (2007)]	Branding the idea: apply stickers on every car that participates. Limitation of drivers: age limits, extra driving tests, check on criminal records etc.	Auto Event Recorders on cars Emergency button on mobile phone, GPS Feedback system a la Ebay	
[Abdel-Naby, S. et al. (2007)]			
[Kelley, L.K. (2007)]	RFID, as carpooling activity is recorded by the system	Everything built around RFID	
[Hartwig, S. and Buchmann, M. (2007)]	Communities and governments should be involved in planning and implementation phases	Registered users only; Feedback system	Possibility to create social connections
[Gruebele (2008)]		Proposes RFID, GPS Rating system based on more than one PQOS metrics. View vehicle and driver information before entering a vehicle. Displaying participants pictures. Assignment of random numbers for passenger pickups to confirm the ride Advanced voice and video proposals.	Multi-hop system is better than single hop: in single trip with 3 hops a passenger might meet 3 to 10 persons Passengers may be socially matched Suggests to link the application to social networks and use profile information to match
[Gidófalvi, G. et al. (2008)]	The use of social networks should improve it		
[Wessels, R. (2009)]	People are ready to spend 17% more time to pickup a friend of the social network rather than a stranger	Rating system GPS Asks for extra research	
[Morris, J. (2009)]	Suggests the social	GPS Help button	

Paper	Critical Mass	Incentives	Suggestions
[Resnick, P. (2003)]			Location awareness system Use of mobile phones
[Kirshner, D. (2006)]	Mass marketing Start-up incentives	An institutional sponsor is needed. Parking spaces provided to participants	Web and mobile Ride requests submitted hours before actually needing them. A many-to-one system: all at a single destination
[Murphy, P. (2007)]		Cooperative public development of the system	Web and mobile (phone call)
[Abdel-Naby, S. et al. (2007)]			
[Kelley, L.K. (2007)]		Employers should incentive employers Regional Transportation Boards	
[Hartwig, S. and Buchmann, M. (2007)]	Incremental service, starting from a thread of backwards compatible services (bus, taxi) Service should not introduce new devices for its use	Find a way to make the service a business case Public incentives	Mobile only GPS non-obtrusive system for authentication QOS measure needed
[Gruebele (2008)]	Multi-hop system brings higher PQOS. Set of potential riders and drivers increases	Governments should be convinced and change laws to enforce carpooling	Dynamic, multi-hop, real- time mobile system Minimize waiting times, one hop at a time
[Gidófalvi, G. et al. (2008)]			Use of mobile phones and sms. Use of GPS. High-level description
[Wessels, R. (2009)]	Users should be involved in some parts of development process Asks for extra research		Mobile + web system that interacts with social networks profiles Use of Opensocial + other social network APIs High level description of the whole system
[Morris, J. (2009)]	Market-formation problem: someone must discover a new, winning formula Start with an existing service, like taxis Find large employers Serve events (ie. concerts)	Money? Incentives from governments	Very high-level Lots of Use Cases Detailed functional requirements Detailed non-functional requirements
[Zimmerman, H. and Stempfel, Y. (2009)]	Main problem is to convince the user to use the system for the first time: marketing money is probably	Provide a free taxi of public transport ride in case of no return possibility (only for the first months)	Mobile, GPS Multi-hop system

Our rationalization of dynamic carpooling issues and possible solutions, summarized in Table 1,2,3 shows how dynamic carpooling systems still have many important open issues to be addressed and solved. This fact explains the current absence of any dynamic carpooling system deployed and used for real. We decided to address the overall challenge from a very core point of view and to focus on technical aspects. Among them, we observed that the source code of the projects and the prototypes produced was not freely accessible by the general public. There are no information regarding the servers, that are all proprietary and obscured. Another issue seems related to a missing standardization of the protocols used. While, in order to overcome the "reaching critical mass" issue, we believe that it is important that providers of dynamic carpooling services can exchange their data easily so that cross provider matchings are possible. Therefore, every project started from zero, "reinventing the wheel".

Based on this analysis, we decided to create to basic technological building blocks:

- An open, discussable protocol for communication between dynamic (and static) carpooling systems

- An open-source server prototype implementing the protocol

We decided to release the protocol under Creative Commons License and to release the source code as open-source because our goal is to fill a void and providing a basic infrastructure that future providers of dynamic carpooling services can adopt, extend and in general build on.

Therefore, the Dycapo project was born.

Part III

Dynamic Carpooling Project: Dycapo

Dycapo is the name given to a project aiming to share knowledge outcomes and technological products on dynamic carpooling. The information are created using a fully open and collaborative system. The website, http://dycapo.org, is the start point and the container of each project component. It is a Wiki[6], freely accessible and discussable. The Wiki content and the source code are available under permissive and open licenses. We used a blog and social networks to update in real-time about the status of the researches and the development. The website was opened on Oct. 11 2009. As of Sept. 11 2010 we had 1617 visits and many e-mail contacts, about proposals of collaboration and general interest.

The following are the components of Dycapo:

- Research - all the theoretical knowledge summarized in Part II
- Protocol - the last draft of the protocol aiming to become a world standard
- Server - the prototype implementing dynamic carpooling functionalities and the protocol
- Client - will act as a start point for the clients implementing the protocol

Our contributions are related to the first three issues: the research, the protocol and the server. A client is under development by a university colleague and will be object of another Bachelor thesis.

For better introducing the protocol and the server prototype, we now present come user stories on system functionalities. After we illustrate a high level description of the architecture of Dycapo system.

3.1 User Stories

A simple trip Paul is in Via Roma, Bolzano, Italy. He wants to move to the cemetary. He takes out his smart-phone with GPS activated, opens his Dycapo client and gives the desired destination, therefore searching a trip. Angela is travelling with her car along Corso Libertà, Bolzano. Her destination is Laives, Italy, southern to the cimitery. She receives a notification from her Dycapo client running on her smartphone, located in a car docking station. The notification contains Paul's position and she accepts his participation request. The client displays the directions for reaching Paul, using the GPS-chip . Paul and Angela meet and share the trip, dividing the costs.

[6]MediaWiki - http://www.mediawiki.org/wiki/MediaWiki

A planned concert Anna lives in Trento, Italy. In two days she will attend a concert in Milano, Italy. She takes out her smart-phone running a Dycapo client and inputs her travel intentions. Two days after she starts her trip. Just after her start, she discovers that Mary is in Rovereto, Italy and would like to have a ride to Verona, Italy. The clients make them meet and share part of Anna trip. While driving Mary to Verona, Anna receives another ride request from John, who is in Cremona, Italy and wants to travel to Milano, too. Anna realizes that the deviation from her planned route is too much and simply refuses the request. Anna arrives to Milano and enjoys the concert.

3.2 Architecture

Dycapo system follows the client-server model, in which a server receives messages from clients (mobile phones) using a protocol, processes them and returns them using the protocol again.

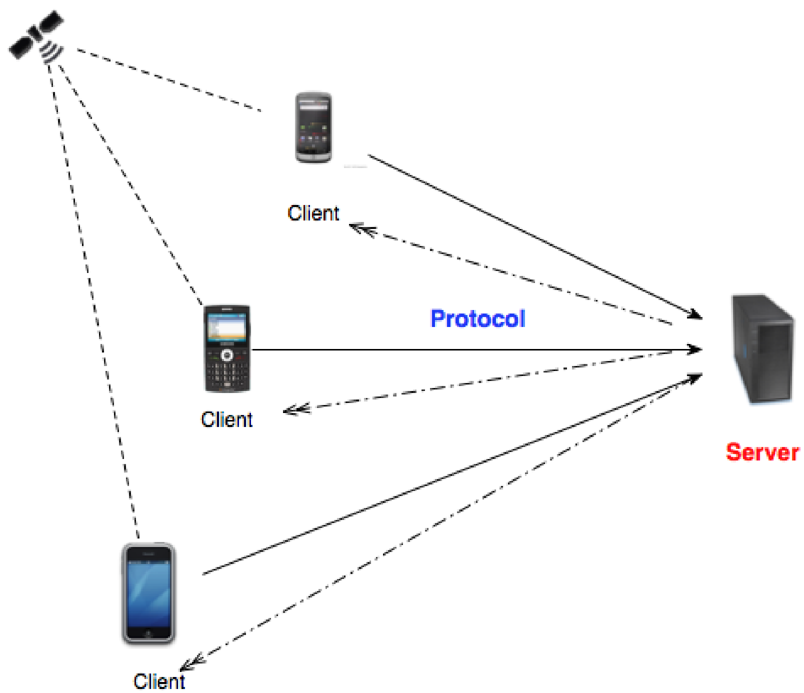

Figure 1: High level view of Dycapo components architecture

4 Protocol

Dycapo Protocol (also known as DycapoP) is an open application-level protocol for enabling communication between dynamic (and static) carpooling servers and clients, using HTTP for operations and JSON (JavaScript Object Notation) as data exchange format. The protocol models are inspired by OpenTrip Core[7], a former proposal of data exchange format for Carpooling and Dynamic Carpooling presented during the MIT "Real-Time" Rideshare Research workshop[8]. DycapoP is released under Creative Commons Attribution-ShareAlike 3.0 Unported license[9]

4.1 Entities

The entities defined by OpenTrip core (using Atom Syndication[10] format) have been extracted and represented using UML 2.1 Class diagrams. While proceeding and discussing the development of the server, those entities were extended and adapted. The entities are represented in the Protocol using descriptive tables and real world code snippets. The Dycapo Protocol draft version of September 15, 2010is included in the Appendix A.

The following are the Models of Dycapo Protocol:

Location A Location represents:

- A geographical position, using human understandable attributes (like street, town, postcode, region, subregion and country) or geolocation values such as georss_point[11]
- An interesting point from a Person's perspective, by setting attribute label to values like 'home', 'work' etc.
- A hop point in a Trip, by setting point and leaves values

[7] http://opentrip.info/wiki/OpenTrip_C ore
[8] http://ridesharechoices.scripts.mit.edu/home/workshop/
[9] http://creativecommons.org/licenses/by-sa/3.0/legalcode
[10] http://www.ietf.org/rfc/rfc4287.txt
[11] http://www.georss.org/simple#Point

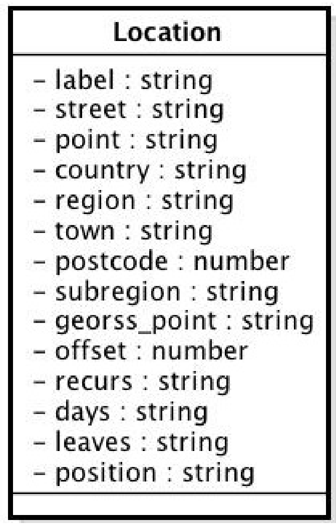

Figure 2: Location Model

Person A Person represents a user of the system and contains useful information for building social connections or search preferences.

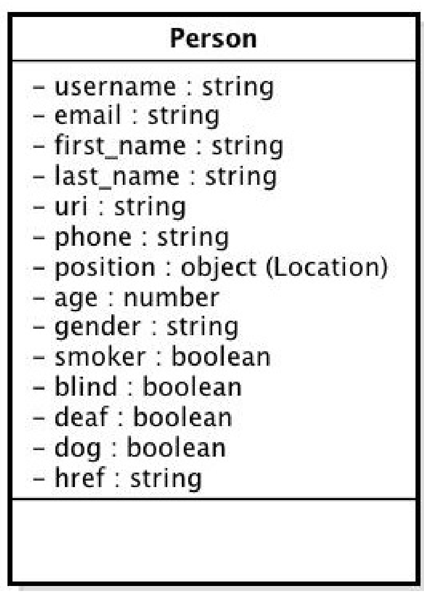

Figure 3: Person Model

Trip A Trip represents a single course of travel taken as part of one's duty, work, etc, offered by a Person. It is the most complex object of the Dycapo Protocol. It contains all the information needed for doing operations with Trips.

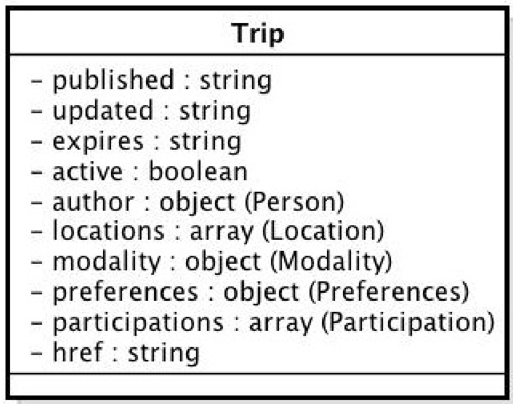

Figure 4: Trip Model

Modality A Modality object is:

- A description of the mode of transportation being used by the Person offering a Trip.
- A representation for the cost of the Trip

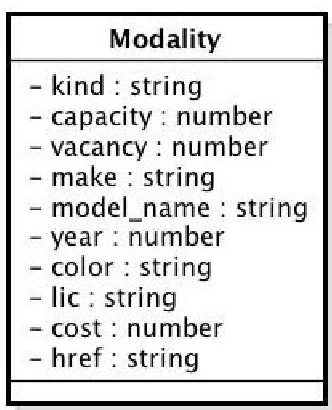

Figure 5: Modality Model

Preferences A Preferences object represents the preferences of Driver when performing a Trip, such as the age range of the Persons he will accept as passengers, whether they can smoke during the Trip, etc.

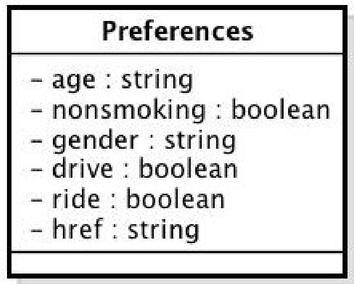

Figure 6: Preferences Model

Participation A Participation object represents the fact of taking part in a Trip. The attribute status represents the status of a Participation, being it a request, a refuse, a start etc.

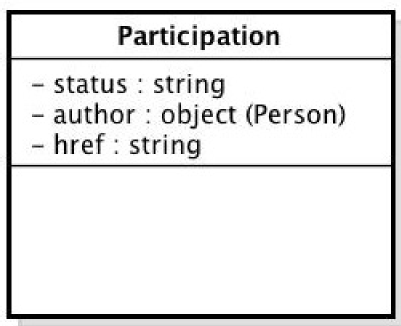

Figure 7: Participation Model

In Figure 8 we represent the possible states of a Participation, using a UML State Diagram. For each transition, we report the value of attribute status and the HTTP operation performed by the operation. We also report those transitions that only the author of the Trip (i.e. the driver) can cause.

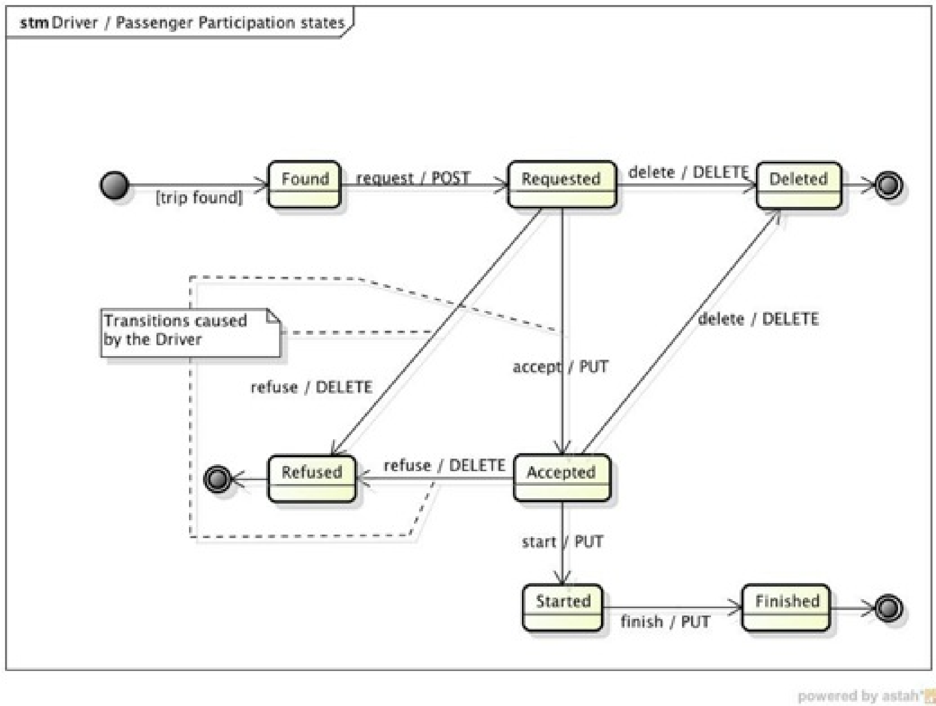

Figure 8: Participation Status

Search A Search is a helper object to construct the complex queries related to the search of a Trip. A client first posts a search object with the desired origin and destination. The Server stores the object in a unique URL that the client can access for retrieving the results of the search.

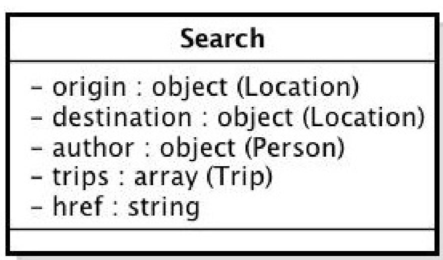

Figure 9: Search Model

4.2 Operations

DycapoP is implemented using a resource oriented architecture, i.e. the definition of a full REST [Fielding, R.T. (2000)] application-level protocol.

All the operations are performed using HTTP methods GET, POST, PUT, DELETE [Fielding, R.T. et al. (1999)]. For example, to retrieve the list of all the stored trips, a client must perform the following HTTP request:

```
GET /api/trips/ HTTP/1.1
Authorization: Basic cmlkZXIxOnBhc3N3b3Jk
Host: 127.0.0.1
```

It is self-descriptive: a GET request of the collection of trips using HTTP version 1.1. The authorization line is present because in our implementation of the protocol the authorization is done using HTTP Basic Authentication[12]. The Host part is required by this HTTP version.

The server will respond in the following way, using a HTTP response.

```
HTTP/1.1 200 OK
Date: Mon, 13 Sep 2010 14:35:26 GMT
Server: Apache/2.2.14 (Unix) mod_ssl/2.2.14 [..]
Vary: Authorization,Cookie
Transfer-Encoding: chunked
Content-Type: application/json; charset=utf-8
X-Pad: avoid browser bug
922
[ {
"updated": "2010-09-02 16:03:24",
"participations": { "href": "http://127.0.0.1/api/trips/3/participations/" },
"preferences": {
    "nonsmoking": false,
    "gender": "",
    "ride": false,
    "drive": false,
    "href": "http://127.0.0.1/api/trips/4/preferences/",
    "age": "18-30"
},
"author": {
    "username": "driver1",
    "gender": "M",
    "href": "http://127.0.0.1/api/persons/driver1/"
},
"expires": "2010-09-05 16:03:05",
"locations": [ {
    "town": "Bolzano",
    "point": "orig",
```

[12] http://www.ietf.org/rfc/rfc2617.txt

```
    "href": "http://127.0.0.1/api/trips/3/locations/",
    "country": "",
    "region": "",
    "subregion": "",
    "days": "",
    "label": "Work",
    "street": "Rom Strasse",
    "postcode": 39100,
    "offset": 150,
    "leaves": "2010-09-02 16:02:22",
    "recurs": "",
    "georss_point": "46.490200 11.342294"
}, {
    "town": "Bolzano",
    "point": "dest",
    "href": "http://127.0.0.1/api/trips/3/locations/",
    "country": "",
    "region": "",
    "subregion": "",
    "days": "",
    "label": "Work",
    "street": "Piazza della Vittoria, 1",
    "postcode": 39100,
    "offset": 150,
    "leaves": "2010-09-02 16:02:22",
    "recurs": "",
    "georss_point": "46.500740 11.345073"
} ],
"href": "http://127.0.0.1/api/trips/3/",
"published": "2010-09-02 16:03:05",
"modality": {
    "kind": "auto",
    "capacity": 4,
    "lic": "",
    "color": "",
    "make": "Ford",
    "vacancy": 1,
    "cost": 0.0,
    "href": "http://127.0.0.1/api/trips/3/modality/",
    "year": 0,
    "model_name": "Fiesta"
    }
} ]
```

The first lines are common HTTP response parts. The status line is a HTTP 200 ok, specifying that the request has been successful. The Content-Type header informs that the server is returning in JSON format that we now briefly analyze. What is returned in this example is an array of Trip objects, enclosed in square brackets. It currently contains just one Trip object, enclosed in curly brackets. Each Trip attribute is in the form of: *"attribute name" : attribute-value*. All attributes are separated by a comma. Some members have simple value types like boolean, number and string, e.g. published, href. Other attributes have complex objects as values, e.g. modality, preferences.

The protocol operations are available in Appendix A. This dissertation focuses on the implementation of the functionalities in Part IV.

The UML 2.1 sequence diagram in Figure 10 summarizes a full user story - the creation of a Trip, the participation of a passenger and its end.

5 Server

Dycapo Server (also known as DycapoS) is a software built using first throwaway then evolutionary prototyping - as defined by [Davis, A.M. (1992)] to demonstrate and support the definition of a standard protocol for dynamic carpooling. In this part we describe the software components, the enabling technologies, the models and the functionalities.

DycapoS source-code is available in an open GIT repository [13], under the Apache License, version 2.0[14].

5.1 Components

Dycapo Server is written using Python programming language, which allows a partitioning of a software into modules and packages. A module is a component providing execution statements and definition of functions, variables, classes, etc. Each module correspond to a single Python file. A module has its own private symbol table. A package is a set of modules or other sub-packages, implemented as a directory. Dycapo Server is organized using separation of concerns, each concern held in a separate package.

We represent in Figure 11 a high-level overview of Dycapo Server components, here described:

- rest - this module holds all the wrappers of the resources exposed to the public using REST architecture.
- piston - this module holds an open-source framework that lets us create application programming interfaces implemented with RESTful principles

[13]http://github.com/BodomLx/dycapo
[14]http://www.apache.org/licenses/LICENSE-2.0.html

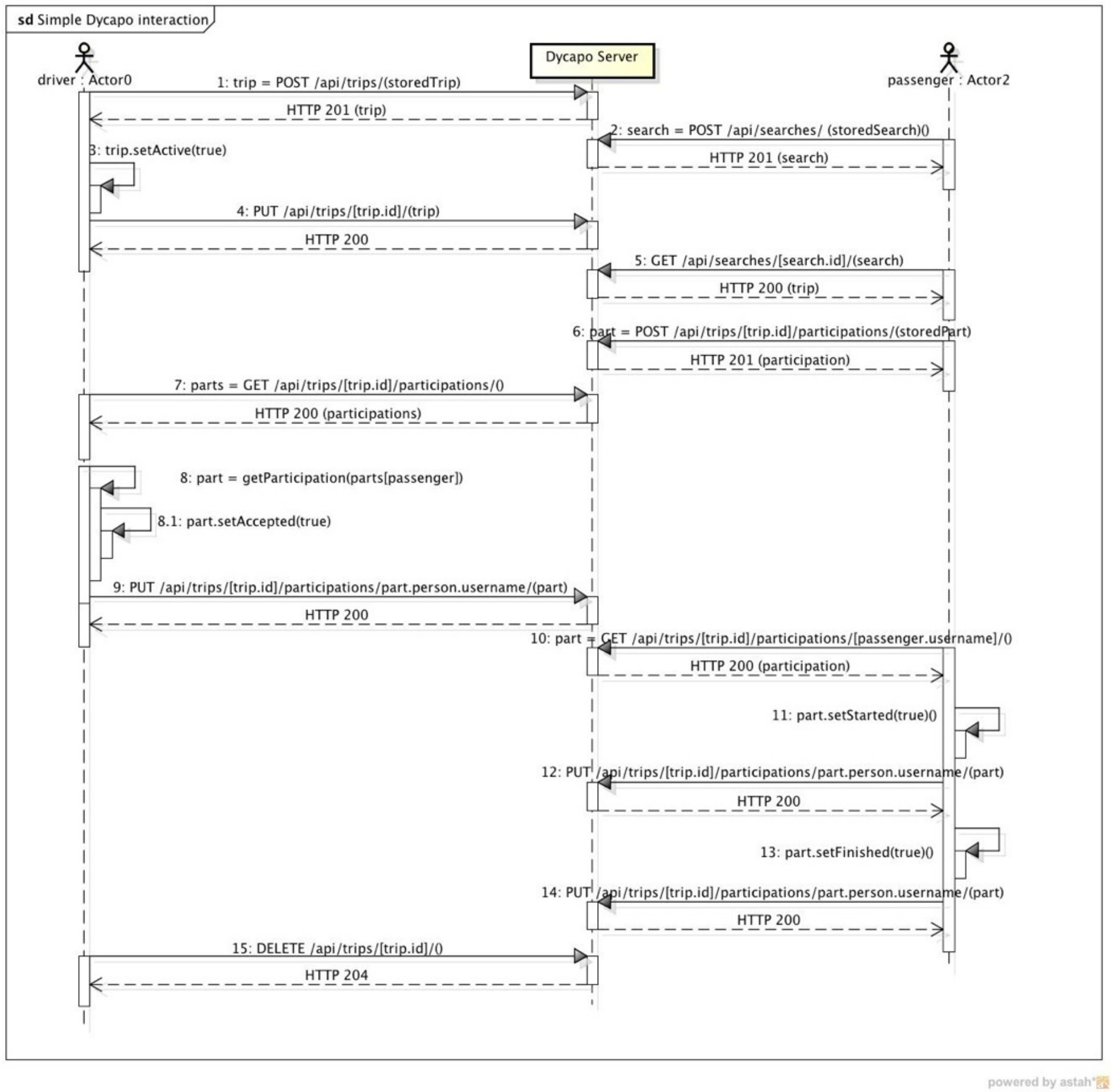

Figure 10: Interaction of two clients and Dycapo Server using Dycapo Protocol

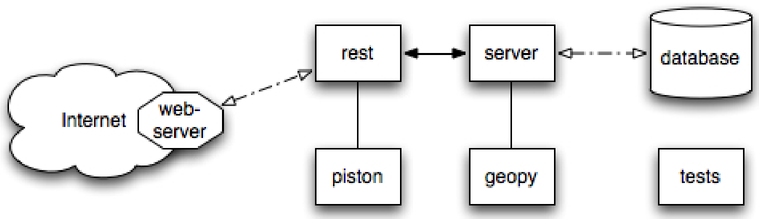

Figure 11: Dycapo Server components

- server -
- geopy - in this module we have a geocoding library for Python for our matching algorithm and for completing Location objects when received from clients
- tests -

5.2 Enabling Technologies

5.3 Models

5.4 Functionalities

Part IV
Conclusions

A Appendix: Dycapo Protocol

References

[Abdel-Naby, S. et al. (2007)] Auctions Negotiation for Mobile Rideshare Service

[Davis, A.M. (1992)] Operational prototyping: a new development approach

[Fielding, R.T. (2000)] Architectural Styles and the Design of Network-based Software Architectures

[Fielding, R.T. et al. (1999)] Hypertext Transfer Protocol – HTTP/1.1

[Gidófalvi, G. et al. (2008)] Instant Social Ride Sharing

[Gruebele (2008)] Interactive System for Real Time Dynamic Multi-hop Carpooling

[Hartwig, S. and Buchmann, M. (2007)] Empty Seats Traveling

[Kelley, L.K. (2007)] Casual carpooling—enhanced

[Kirshner, D. (2006)] Pilot Tests of Dynamic Ridesharing

[Morris, J. (2009)] SafeRide: Reducing Single Occupancy Vehicles

[Murphy, P. (2007)] The Smart Jitney: Rapid, Realistic Transport

[Resnick, P. (2003)] SocioTechnical Support for Ride Sharing

[Wessels, R. (2009)] Combining Ridesharing and Social Networks

[Zimmerman, H. and Stempfel, Y. (2009)] Current Trends in Dynamic Ridesharing, identification of Bottleneck Problems and Propositions of Solutions