A Mobile Phone Solution to Improve Geographic Mobility

Sacha Miteche, Alfredo Terzoli and Hannah Thinyane
Department of Computer Science
Rhodes University,
Grahamstown, South Africa.
email: smiteche@yahoo.co.uk,{a.terzoli, H.Thinyane}@ru.ac.za

Abstract-Motor vehicle ride sharing is another way of collective travel other than the public transport systems. Dynamic Ride Sharing (DRS) systems targeting mobile phone platforms have been developed to enable spontaneous ride share agreements between drivers and unknown commuters with similar travel destinations. Such spontaneous ride sharing agreements are also popular in South Africa through hitch hiking. The DRS systems that are available have been designed for use within a developed world context, and are therefore not relevant to the majority of the population of developing nations like South Africa. This paper presents the work that was done in developing an appropriate DRS system for people who use hitch hiking travels in South Africa.

Index Terms—Hitch hiking, Dynamic Ride Sharing.

I. INTRODUCTION

Vehicle ride sharing between drivers and commuters with common travel destination has existed over the years as an alternative mode of transport to public transport. In some cases, ride share travel arrangements are a result of campaigns to address issues such as high carbon emissions and traffic congestions [1]. This is the case in many developed nations. While in other cases, ride sharing travels happen without such formal influence. This is the case in many developing nations where ride sharing offers an alternative to the limited or poor public transport services.

Online systems to organize vehicle rideshares have existed for a number of years and have gradually changed in design responding to the advancement of technology [2]. Now the focus is on Dynamic Ride Sharing (DRS) methods which are flexible enough to overcome the barrier of fixed travel times whereby participants must schedule a round-trip ride in advance [3]. As such, DRS systems have targeted mobile phones which as described by Massaro [4] are the most pervasive modern-day "nomadic device" for our less predictable lifestyles. This has brought in the culture of arranging instant ride shares whilst on the move.

In South Africa, hitch hiking travels involving private car owners (drivers) and commuters are popular as observed in the Eastern Cape [5]. Such travel arrangements can be organized and managed using the capabilities of current state DRS systems. But, the available DRS systems are designed with a particular context of use, and user characteristics in mind as is the case with all software [6]. The design considerations for hitch hiking travels and the target mobile phone platforms (e.g. smartphones only) are not appropriate for South Africa's context: Majority of people who use hitch hiking travel belong in the lower LSM1-10 (Living Standard Measure) bracket [5]. Therefore, an appropriate DRS system that considers users with different backgrounds (e.g. ICT literacy levels) through simplified operations and accommodates a range of mobile phones (used in South Africa) is necessary for hitch hiking travelers to benefit from the use of DRS services.

This paper presents work that was undertaken to design and implement a DRS system for spontaneous ride share arrangements (using mobile phones) based on hitch hiking travel in South Africa.

II. RELATED WORK

A) Dynamic ride sharing systems.

Dynamic ride sharing, also known as real-time ride sharing, is a form of ride sharing that is used for single, one way trips rather than for trips made on a regular basis at scheduled times [7]. Therefore, according to Levofsky [7], designs of DRS systems must have at least the following considerations:

1. Consider each trip individually.
2. Accommodate trips to random points at random times by matching user trips without regard to trip purpose.
3. Provide match information close to the time when users need travel.

Previous studies on ride sharing systems [2] have shown that there are several important areas of consideration for any DRS system to be successful. These include incentives, trust, convenience and usability.

Incentives can be financial (reduced cost), environmental (reduce carbon emissions) or time saving (driving in high occupancy vehicle lanes) [2 ,7]. Incentives encourage people to participate making the ride share operations complete e.g. high availability of ride offers by drivers.

The matching of unknown ride partners brings out trust issues that lead to security concerns. As such, most modern DRS systems use Global Positioning Satellite (GPS) to track a ride share trip progress e.g. Avego [8]. Other designs are now incorporating social networks for verifications of proposed ride partners to improve trust [1, 9].

The convenience of arranging random meeting points for drivers and commuters is another important aspect in DRS. Modern DRS systems are using GPS technology and real-time map information to suggest the best meeting points [8, 10]. This has improved the usability of DRS systems.

B) Open source software solutions.

The use of appropriate ICT solutions (e.g. hardware and software relevant to a particular context of operation) has been identified as a crucial factor in sustaining ICT for Development (ICT4D) projects [6]. Therefore, the sustainability of our developed system depends on the choice of software tools made in the development process and the deployment environment.

We now discuss selected open source software tools that are being used in the implementation of ICT projects and were considered in our DRS system.

1. Modular application development tools.

Modular application development is considered a best approach to develop complex systems that are robust and maintainable [4]. We now focus on Spring, OSGi and Spring Dynamic Modules (SpringDM) technologies as enabling tools for the development of modular applications and the benefits they provide.

Spring is an open source lightweight framework that promotes loose coupling of classes leading to modular programming [11]. The loose coupling is achieved through a technique known as Dependency Injection (DI) whereby objects are passively given their dependencies instead of creating or looking for dependent objects for themselves. This improves the maintainability and testability of software projects [12].

OSGi is a technology that addresses Java's limitation regarding modularity. The OSGi specifications require the following [11]:

- Java classes defined in deployments as modules.
- Finely set visibility rules between modules.
- Define a lifecycle of modules
- Modules interact via the service registry.

This promotes the good practice of developing modular applications composed of a complete set of logical pieces (modules) such that the internal implementations are only visible to code that is part of the module. For the rest of the application code, the only details that are visible from a module are explicitly exposed through a public API [12]. Independent lifecycle for each module enables simplified management of services. Services can be stopped, removed or updated without restarting the container or server. Modules in OSGi are called bundles and they are standard JAR files with additional metadata in a manifest file (e.g. Import Packages) [11].

Spring and OSGi technologies have their limitations. Spring has no built in support to update bean dependencies at run time while OSGi lacks the tools to design and implement bundles [11]. Spring Dynamic Modules (SpringDM) can be described as a bridge between Spring applications and OSGi which takes care of the limitations that exist in both technologies. SpringDM bundles are deployed in an OSGi container to watch other bundles and to create Spring application contexts for them [11]. In this way, Spring framework enhances the OSGi platform with its features to design and implement bundles.

Figure 1 shows OSGi bundles with Spring contexts created by SpringDM, in an OSGi container. Equinox is an example of a Spring container that supports the OSGi framework.

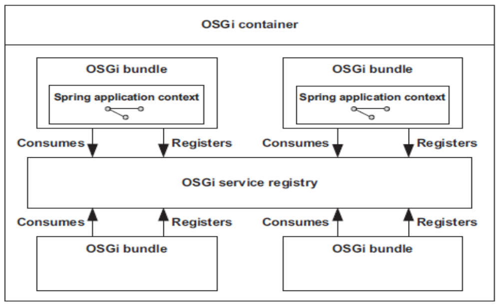

Figure 1: SpringDM and OSGi, taken from [11].

We have discussed how Spring framework and OSGi technologies are integrated using SpringDM for the development of modular applications that use features of both technologies. Next, we will discuss a platform that hosts such modular applications as services that can be referenced by other modules or external systems.

2. Service deployment platform.

Teleweaver is a service platform for ICT4D projects that integrates applications for marginalized communities under the Reed House Systems [13]. The middleware uses OSGi technology in a customized Equinox container suitable to the environment of limited resources, in terms of hardware and software [13].

Teleweaver provides a foundation for developing component and service based solutions. Therefore the platform follows all principles of Service Oriented Architecture (SOA) in a particularly lightweight way [11]. Currently Teleweaver hosts web and mobile applications for e-Government and e-commerce [13].

Developing modular applications which are manageable and deploying them in a platform designed for their environment of operation should ensure the sustainability of the developed solution.

3. Open source WAP and SMS solutions.

Kannel [14] is an open source WAP (Wireless Application Protocol) and SMS (Short Message Service) gateway for GSM networks. WAP defines a simpler markup language than HTTP which is ideal for low capacity mobile phones with at least GPRS (General Packet Radio Service) capability. Kannel, as a WAP gateway, provides interface between the WAP protocol stack in GSM networks and the HTTP protocol for content providers on the Internet [14]. As an SMS gateway, it enables relaying of SMS contents to content providers through the HTTP protocol. Kannel can be configured to run an SMS Service that triggers an action whenever a specified keyword is included in an SMS. Using Kannel’s HTTP client, the triggered action can be a web service call to a content server on the internet that responds with some content [14].

We have looked at the open source tools used for modular application development followed by the deployment platform that supports SOA solutions. Then, an open source solution to interface mobile telecommunication protocols (e.g. SMS) with Internet Protocols (e.g. HTTP) has been explained to show how services hosted in IP networks can be accessed by mobile phone clients.

III. METHODOLOGY

This section describes the steps that were undertaken to understand hitch hiking travels in South Africa for the development of an appropriate DRS system.

1. Defining the research area.

We conducted our research in the urban and rural areas of the Eastern Cape, South Africa. In this region, hitch hiking travels are popular as reported by Sicwetsha [5]. We covered the urban areas that included Grahamstown, Port Elizabeth and King Williams Town. For rural areas, Dwesa and its surrounding communities in the Mbhashe municipality were covered in the study.

2. Survey literature.

A survey of literature was performed to determine the current state of hitch hiking within South Africa. We were looking for information on the reasons behind hitch hiking travels and the people who actively participate.

3. Survey on hitch hiking travels.

We conducted a survey on hitch hiking travels in both urban and rural perspectives. Questionnaires and interviews were used to obtain information from people who use hitch hiking travel. This was done using the snowball sampling method.

Our survey goals were to understand the following:

- The incentives to drivers and commuters.
- The hitch hiking process.
- Risks associated with hitch hiking travels.
- People's thoughts on how to improve hitch hiking travels.

4. Participation in hitch hiking travels.

To further enhance our understanding on hitch hiking, we took part in several trips in a hitch hiker's role. The trips were arranged between Grahamstown and other locations in the Eastern Cape (Port Elizabeth, King Williams Town and East London). Informal interviews with hitch hikers and drivers were performed in the process.

5. Hitch hiking models.

After gathering the survey results and analyzing them, we developed two hitch hiking models, for urban and rural scenarios. These models highlight the different characteristics that we found in both hitch hiking contexts.

6. System design.

The design process was guided by the hitch hiking models and other survey findings such that the use cases and system specifications were designed to be appropriate to the hitch hiking culture in our research area.

7. Implementation of DRS Service.

Using the open source software tools discussed in Section II, a DRS Service was developed with OSGi specifications and deployed in Teleweaver.

8. Setting up the test environment.

A test environment was setup for mobile phone clients of different capabilities (low and high end phones) to access the DRS Service in Teleweaver. Kannel, running on a separate machine, was included to provide interface between mobile phone clients on GSM networks and Teleweaver.

IV. SURVEY RESULTS AND ANALYSIS

We now present the results and analysis of our study on hitch hiking travels.

1) Survey literature results.

In our survey for similar studies on hitch hiking in South Africa, we found a report on the reasons behind hitch hiking in the Eastern Cape [5]. This study involved a sample population of motorists who pick hitch hikers, commuters and representatives of the Minibus Taxi industry. Questionnaires were randomly given to people in urban and rural areas. Table 1 shows the results of the study showing the main reasons why people support hitch hiking travels.

	Total	Support hitch hiking	Main reason
Commuters	66	41	Dissatisfied with the Taxi Industry
Motorists	75	59	Supplement fuel cost

Table 1: Questionnaire results on reasons for hitch hiking

Inefficiencies within the Taxi Industry have made people to prefer hitch hiking travels. These include bad customer care, bad relationships with customers, delays at the rank, etc. [5]. This has made drivers with long distance trips take advantage of the situation by offering ride shares in order to minimize their travel costs (fuel cost).

2) Survey results on our hitch hiking study.

In our questionnaire survey, we targeted a sample size of 30 people who use hitch hiking travel within the Grahamstown area. A total of 22 people gave their feedbacks. Table 2 shows the general responses that were given.

Question	General Responses
As a commuter, how do you find an appropriate hitch hiking spot?	• There are known places. • Strategic points along a street that leads to the intended destination.
What do you think are the benefits of hitch hiking travel?	**Hitch hiker role:** • Quick travel time. • Cheaper than Minibus Taxis.
	Driver role: • Save travel costs (fuel).
What do you think are the risks associated with hitch hiking travel?	**Hitch hiker role:** • Theft cases. • Rape cases for women. • Travel using vehicle with unknown details of owner and its condition for long distance travel. • Traveling with criminals.
	Driver role: • Car hijacks cases. • Transporting criminals • Travelling with strangers.
What recommendations would you suggest to improve hitch hiking travel?	• Do not travel alone. • Travel details like car and the driver must be made available to relatives or friends.

What information would you prefer to have, before participating in a hitch hiking travel?	• Details of a hitch hiker or driver • The car details.
As a commuter, what estimated waiting-time would you allow before the start of an organized ride share trip?	Most indicated the waiting time within minutes to an hour (15 min, 20 min, and 30 min to 1 hr.).

Table 2: Questionnaire responses

In our interviews, we used the same questions from the questionnaire. A total of 17 interviews were done in the urban areas: 10 were formal and 7 informal interviews. The informal interviews were done during the participated hitch hiking travels. The responses were similar to the questionnaire results listed in Table 2.

In the rural areas of Dwesa, 4 interviews were conducted with the locals of whom 3 were teachers. The interviews were done at different location namely Ngwane, Mpume, Badi and Nqabara. We found that the main reason for hitch hiking travel was due to limited public transport services. In this area, public bus service operates twice a day, morning and evening. Figure 3 has other characteristics that we found in rural hitch hiking travels.

3) Hitch hiking process.

The results of our survey showed that commuters know exact places for hitch hiking which drivers target for ride share partners. At the hitch hiking spot, a driver and a hitch hiker agree a ride share using the destination name only e.g. Port Elizabeth. It is a common practice for hitch hikers to hold a post with the name of a destination if the hitch hiking spot serves many possible travel destinations. This assists drivers to easily identify possible ride partners.

Agreements on travel cost and the final drop point for passengers are done whilst the trip is in progress. It was observed that people are well aware of how much it costs moving from one point to another. In urban areas, the hitch hiking travel cost was found to be relatively lower than the available public transport cost e.g. Minibus Taxis. For example, we paid a ride share cost of R50 to travel between Grahamstown and Port Elizabeth while on Minibus Taxi it was R80. However we also found that in rural areas, the hitch hiking cost is higher than the alternative public transport cost.

The hitch hiker's final destination point depends on the driver's decision on where to drop the passengers. We observed this in the trips we took part. In our three trips to Port Elizabeth we had different drop off points on each occasion. This makes the hitch hiker's final destination point random in the destination location.

4) *Risks associated with hitch hiking.*

For drivers, the major risks identified were car hijackings and giving transport to criminals. Commuters are concerned with kidnappings, rape cases, travelling in unknown vehicle details and condition, etc. According to Sicwetsha [5], most people are well aware of the major risks. We also found that people are aware of the risks from the survey we conducted.

To reduce the risks, drivers and hitch hikers suggested that verification of ride partners for their criminal records would be important to trust them. Others also suggested the availability of ride share trip details (e.g. vehicle used, driver name) to friends or relatives.

5) Hitch hiking models

Hitch hiking models were designed to highlight the characteristics of hitch hiking in urban and rural areas respectively as shown in Figures 2 and 3.

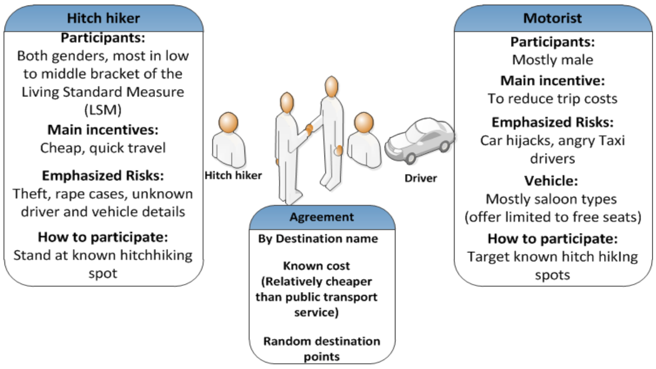

Figure 2: Urban hitch hiking model

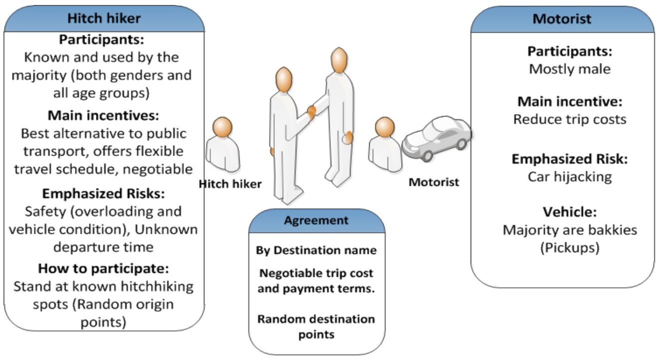

Figure 3: Rural hitch hiking model

V. DESIGN

Based on our understanding of hitch hiking travels in our context and features of existing DRS systems, we designed our DRS system with use cases as shown in Figure 4.

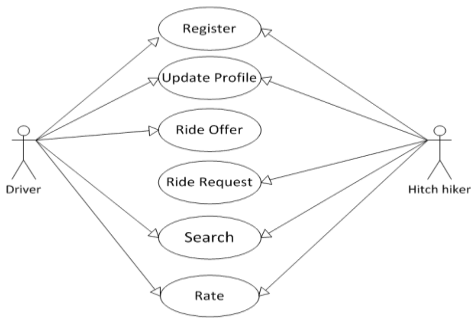

Figure 4: Use Case diagram

The design specifications were set as follows:

1. Registration of users of the system..

The registration process takes in the full name, sex, phone number, home location, username and password.

2. *Registration of vehicle used for hitch hiking travel*.

A driver must provide some details of the vehicle to be used for a ride sharing trip. The details are vehicle registration number, make, body type, colour and the passenger seats.

3. *Creating a ride share trip*.

A driver creates and owns a trip. The trip has an origination point which is a well-known hitch hiking spot. Therefore details of the originating point must include the location name with associated street and nearest landmark (if available). The trip has a final destination name and optional transit route points (maximum of 2) if the driver is willing to pick transit commuters. The process generates a secret trip code that is made available to hitch hikers who are accepted in the trip.

.

4. *Ride offer (by driver).*

A driver makes a ride offer by specifying the created trip and additional conditions (e.g. cost, luggage size). A trip offer time window must be specified to indicate the minimum and maximum time for pickup. The number of seats offered must be specified or else the vehicle seat capacity will be used. Optional conditions such as luggage limit or expected ride share cost can also be specified. A trip's ride offer ceases to be available outside its specified offer time window.

5. *Ride request (by Hitch hiker).*

A person looking for a spontaneous ride offer will take the role of a hitch hiker. Details of a hitch hiker must include the current standing point (hitch hiking spot), the travel destination name and optional conditions such as vehicle space requirements.

6. *Matching algorithm*.

The algorithm has the following specifications:

- Use of origin and destination parameters.
- Matching results limited to the offered seats.
- If match results do not meet the offered seats then transit commuters are searched using the driver specified transit route points.
- Sort hitch hiker ride requests based on their timestamp to match with ride offers.
- Pend hitch hiker ride requests until a ride offer is available.

7. *Rate ride partners.*

After a trip, ride partners can rate each other and add recommendations.

The design specifications took into account some features of existing DRS systems as well as feedback from questionnaire and interviews described in the previous section.

The generation of a secret code for a trip is used as a security measure in Avego [8]. In our DRS, this serves the same purpose and the secret code has the format of a trip's final destination plus a 6 digit number e.g. PE014563 where P.E is Port Elizabeth as the trip destination.

The use of a ride offer window clears the uncertainty in specifying the exact meeting time for driver's arrival and departure at a hitch hiking spot [4]. The driver has to make decisions on accepting hitch hikers for the trip within the offer window. We set the offer time window range from 15 min to 1 hour using the feedback from the survey (Table 2).

The matching algorithm used is based on origin and destination names. Other algorithms in DRS systems are based on the commonality of the travel route of a driver and commuter [15]. In our context of hitch hiking, origin and destination based algorithm fits well.

VI. IMPLEMENTATION AND TEST ENVIRONMENT

The implementation process followed guidelines for developing applications for Teleweaver [13]. Java was the choice of programming language and Eclipse IDE provided the development environment with OSGi framework.

1. Project setup in Teleweaver.

The project structure has two bundles, API and the implementations. The implementation bundle which is subject to frequent updates benefits from the OSGi dynamic features such as on-the-fly service updating [11]. Figure 5 shows the bundles that make up the DRS Service inside Teleweaver.

The API bundle has two Java interface classes. A service interface class defines the functions for ride sharing service e.g. Offer Ride, Ride Request. The second interface class has the definition of Data Access Object (DAO) functions.

The implementation bundle has the implementation classes for the API interface, service and DAO implementation. In addition, the domain model with all entity classes is defined. Hibernate provides the Object Relational Mapping (ORM) to the MySQL Database. This bundle has imports of Java packages for dependencies explicitly set in its manifest file. The dependencies include packages for Hibernate, MySQL, SpringDM, etc.

To enable HTTP communication between Teleweaver and Kannel, a web service bundle that uses Representational State Transfer (REST) architecture was developed using the Apache CXF framework [17].

Then, a SpringDM Model View Controller (MVC) bundle [11] was added to generate Java Server Pages (JSP) with Wireless Markup Language (WML) tags for WAP browsers.

Finally, a Google Web Toolkit [16] bundle was developed to provide AJAX web pages for JavaScript enabled web browser clients.

.

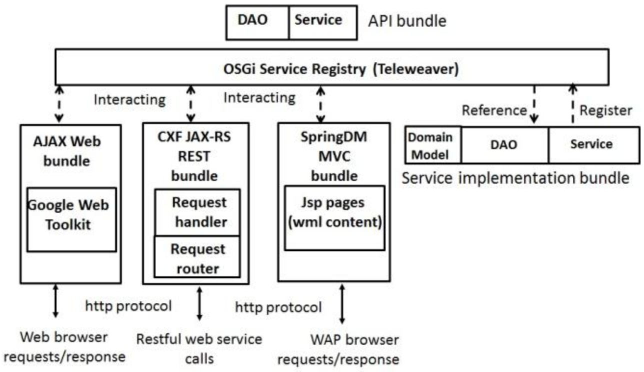

Figure 5: The DRS system bundles in Teleweaver

In Teleweaver, the DRS Service is registered in the OSGi registry for reference by other OSGi bundles (e.g. the SpringDM MVC bundle) as shown in Figure 5.

2. Test environment setup.

As shown in Figure 6, Teleweaver serves the DRS Service using three different mobile phone access methods that are available in South Africa. The access methods are:

1. SMS application. Targeting low capacity GSM mobile phones using the more familiar SMS style of communication.
2. WAP service. To provide faster web browsing for low capacity phones with at least GPRS connection.
3. HTTP service. For high end mobile phones with web browsers capable of running JavaScript (targeting mobile phones with at least 3G network connection).

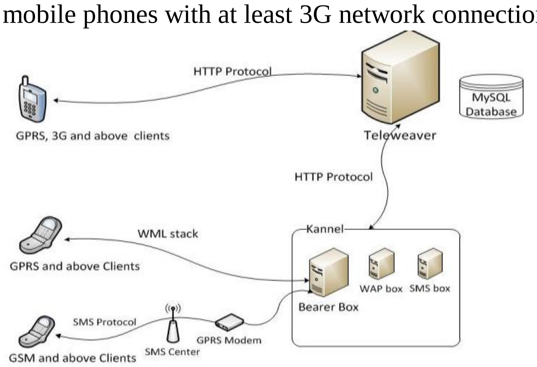

Figure 6: Overview of the test environment

In an SMS application scenario, an SMS with a keyword and the required parameters (e.g. for Ride request) is sent to Kannel (connected to a modem with SIM card) which forwards the request to Teleweaver via the HTTP protocol. In Teleweaver, the DRS Service is referenced by the REST web service bundle for a response that is sent back to Kannel. Kannel structures the message back into SMS format and sends it to the requesting mobile phone client.

VII. CONCLUSION

Vehicle ride sharing is practiced in both developed and developing nations. Different incentives are behind the participation of drivers and commuters in such travel arrangements. By studying the way people arrange their dynamic rideshare through hitch hiking in our context, we have designed a relevant DRS system that mirrors the informal process currently undertaken by hitch hikers. We expect that the addition of some features of existing DRS system such as logging of all trip details will improve hitch hiking travels. It is anticipated that once functional tests followed by usability tests prove successful then the DRS system will be relevant to hitch hikers and drivers in South Africa.

VIII. REFERENCES

[1] V. Chaube. Understanding and Designing for Perceptions of Trust in Rideshare Programs. Masters Thesis, Virginia Polytechnic Institute and State University, July 2010.

[2] S. Heinrich. Implementing Real-time Ridesharing in the San Francisco Bay Area. Master's thesis, Mineta Transportation Institute San Jose State University, 2010.

[3] E. Deakin, K. Frick, and K. Shively. Markets for Dynamic Ridesharing? Case of Berkeley, California. University of California Transportation Center, Working papers, University of California Transportation Center, 2011.

[4] D. Massaro, B. Chaney, S. Bigler, J. Lancaster, S. Iyer, M. Gawade, M. Eccleston, E. Gurrola, and A. Lopez. Carpoolnow - Just-in-time Carpooling without Elaborate Preplanning. In Joaquim Filipe and José Cordeiro, editors, *WEBIST 2009 - Proceedings of the Fifth International Conference on Web Information Systems and Technologies, Lisbon, Portugal, March 23-26, 2009*, pages 219–224. INSTICC Press, 2009.

[5] M. Sicwetsha. Hitch-hiking Research Report, Reasons Behind Hitch-hiking in the Eastern Cape. http://www.ectransport.gov.za/uploads/Reports/reasons-behind-hitch-hiking-in-the-eastern-cape1.pdf, Online [Last accessed on 12/04/12].

[6] V. van Reijswoud. Appropriate ICT as a Tool to Increase Effectiveness in ICT4D: Theoretical Considerations and Illustrating Cases. *The Electronic Journal on Information Systems in Developing Countries*, 38:1–8, 2009.

[7] A. Levofsky and A. Greenberg. Organized Dynamic Ride Sharing: The Potential Environmental Benefits and the Opportunity for Advancing the Concept. Report, Transportation Research Board, January 2001.

[8] Avego. Avego Real-time Ridesharing. http://www.avego.com/products/real-time-ridesharing/, Online [Last accessed on 30/05/12].

[9] R. Wessels, R. Pueboobpaphan, J. Bie, and B. van Arem. Integrating Online Social Networks with Ridesharing Systems: Effects of Detour and Level of Friend. http://www.pooll.nl/wp-content/uploads/2010/02/Paper-v5.pdf, Online [Last accessed on 3/03/12].

[10] OpenRide. OpenRide Ridesharing. http://www.open-ride.com/, Online [Last accessed on 25/04/12].

[11] A. Cogoluegnes, T. Templier, and A. Piper. *Spring Dynamic Modules in Action*. Manning Pubs Co Series. Manning Publications, 2010.

[12] C. Walls and R. Breidenbach. *Spring in Action*. Manning Publications, 2008.

[13] Reedhouse Systems. http://reedhousesystems.com/, Online [Last accessed on 2/04/12].

[14] Kannel 1.5.0 User's Guide, WAP and SMS Gateway. http://www.kannel.org/download/1.5.0/userguide-1.5.0/userguide.pdf online [Last accessed on 9/05/12].

[15] K. Ghoseiri, A. Haghani, and M. Hamedi. Realtime Ridesharing Matching Problem. Technical report, University of Maryland Department of Civil and Environmental Engineering, 2011.

[16] Google Web Toolkit. http://developers.google.com/web-toolkit/, Online [Last accessed on 4/05/12].

[17] Apache CXF, Restful Services. http://cxf.apache.org/, online [Last accessed on 23/07/12]

Sacha Miteche received his Honours in Computer Science in 2010 from Rhodes University and is presently studying towards his Master of Science degree at the same institution. His research interests include computer networking, mobile and database technologies.

This work was undertaken in the Distributed Multimedia CoE at Rhodes University, with financial support from Telkom SA, Tellabs, Genband, Easttel, Bright Ideas 39 and THRIP.