Dynamic Routing to Meet Up

Nara Kim
Computer Science and Engineering
University of Washington
Seattle, WA 98195 USA
narakim@cs.washington.edu

Matt Moyers
Computer Science and Engineering
University of Washington
Seattle, WA 98195 USA
mmoyers@cs.washington.edu

ABSTRACT
We explore using dynamic routing on a smart phone to guide one user to the other(s) to meet up. We built a prototype and conducted various tests including user study. Users found the application very useful and yet further work must be done to improve and overcome the limitations and the technical difficulties such as GPS accuracy and indoor route operation. Interestingly, it was initially intended for blind or visually impaired users, yet we found out that the same problem arises among non-blind users as well.

General Terms
Human Factors, Design.

Keywords
Blind, accessibility.

1. INTRODUCTION
People with visual impairments experience challenges in finding each other physically at a meeting location. Without extra help, knowing details about the location is inevitable before they leave. However when it comes to large areas such as a park, a shopping mall, or a convention center, difficulty of the task is even greater. The same problem arises among non-blind people when they have never met each other or when they have never been to the meeting place.

Sharing user's location with friends is a popular feature called "check-in" in commercial systems, yet the lack of details on friend's location does not alleviate the problem.

We explore a way to give routing information to users dynamically as they move closer to each other. We built a prototype in Android operating system and tested with a blind user.

2. RELATED WORK
Many applications exist that deal with giving location and routing information to the user.

Many popular GPS apps exist telling users when to turn how to navigate the streets, some even offer walking routes. Others also provide assibile interfaces targets and blind or low vision users. This provides an easy way for one user to get to a fixed location.

One emerging commercial feature called "check-in" from Facebook, Foursquare or Google Latitude provides a great tool to share a user's location with friends. It also supports a routing feature that gives one user a route to where someone last checked in. Once again this routing information is based on a fixed destination location.

Figure 1. Check-In Feature provided by Google Latitude on the iPhone.

3. Prototype & Solution
The MeetUp Application provides a different approach to the problem of finding people. Unlike the options currently avaible, Meet Up does not rely on fix location routing. Instead both the orgin and desination can move. Since each point is moving the application must handle detecting and updating a change in the route. This can be caused by one of the points not following the orginal route provided.

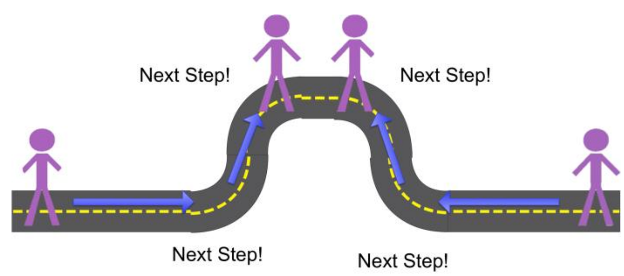

Figure 2. Typical case of routing two people to each other.

Our solution has two main components: A Webserver and a MeetUp Client Application.

3.1 MeetUp Client Application
This is an android application that handles interaction with the user. The GUI consists of 4 screens: Contact, MeetUpAct, Direction, and Travel. Each screen has a certain function within the application.

Contact: Select user from contact list.

MeetUpAct: Call, Text or MeetUp on the previously selected contact.

Direction: Gives direction information to the user about the current route.

Travel: Changes the travel mode to walking, driving, or bus.

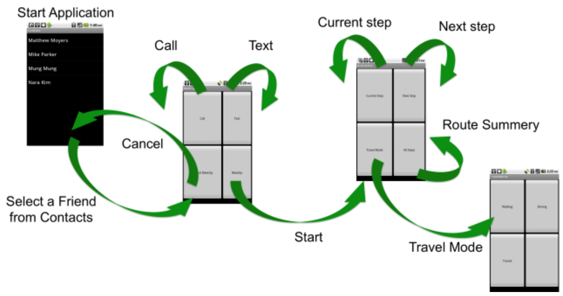

Figure 3. Overview of GUI interface, showing the four screens and how to each one corresponds to each other.

This application has three states it can be in: Start, Waiting and Active. Moving between each state and what state the application is in depends on two factors, your MeetUp State and your friends MeetUp State.

When the application first opens, it is in the **Starting** state.

When the user starts MeetUp on a friend, the application goes into the **Waiting** state. Our system works on a two-way handshake protocol to establish when location data should be shared. Therefore both users have to start MeetUp on each other before any data is share.

When your friend has started MeetUp, the application goes into the **Active** state. In this state location data and route information is refreshed. All routing information is obtained from Google Map API System.

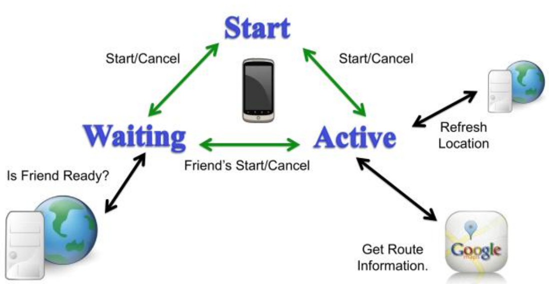

Figure 4. Architecture of the MeetUp.

3.2 Webserver

The Webserver handles relaying location data for all the users currently using the MeetUp Client Application. Mainly written in PHP, the webserver uses MySQL to store all data that is needed and is completely stateless, following the RESTful server model.

The API to the Webserver allows three operations to be performed: Start, Cancel and Update. The Start operation begins the first part of the two-way handshake needed to activate MeetUp. Also this operation sets up the database for this user. The Cancel operation clears all data about that user from the database, preventing anyone from accessing it later. The Update operation refreshes the data in the database and gets the last known location of their friend.

3.3 Compass

As part of the MeetUp Application we wanted to give the user relative feedback based on the current direction they were facing. For this we build an entire system dealing with the compass and route direction to give this feedback to the user. In this system we investigated different ways to inform the user of the direction they needed to go. Three systems were considered, PhoneWand, ScreenEdge and Pattern. Based on research done by Shiri Azenkot [1], Pattern was selected. Pattern uses vibration feedback to tell the user straight, left, right or back.

Upon extensive testing, the compass was deemed too inaccurate and error prone to provide useful information to the user. It was decided to omit from the application until a better solution was

4. LIMITATION

The MeetUp relies on many third party systems such as Google Maps and GPS. GPS accuracy was an issue with our application. This accuracy varies by phones and the users location. We found that tall buildings, weather and general location, i.e. some areas have more GPS satellite coverage than others, affect the accuracy of the GPS. This accuracy ranges from 30 to 80 meters at times.

For indoor operations of our application, the accuracy decreases drastically as well. That paired with limited indoor mapping data makes our application exclusively used for outdoor operations.

Alternatively, network provider can be used, yet the accuracy was worse than the GPS accuracy. We are still looking for a solution to this problem by combining multiple sensor data to get a better position fix. It is our hope that as this technology improves our application can be used in more locations across the globe.

5. FUTURE WORK

5.1 Group MeetUp

Currently the Meetup is restricted to two users meeting each other. We can add a group MeetUp feature, which supports the same operations as for two users but for a group of people trying to meet up. One solution is to pin point, which has minimum distance from every one in the group and routes everyone there.

5.2 Network Optimization

When user is at active state, the server sends updated location of user and friend to Google Map and receives updated routes. However, currently it requests an update every 10 seconds, which can be optimized greatly. Route should be updated only when the user is off the track and at the same time, it should minimize the error in user feedback as opposed to constant updates. This can be obtained by moving to a push notification based system.

5.3 Bus Routing

All routing information is obtained from Google Maps API. However that system doesn't provide bus routes, which from user feedback would be greatly used. This data might come from local METRO systems but further investigation is required.

6. ACKNOWLEDGMENTS

The authors thank Peggy Martinez from Seattle Light House for Blind, Professor Alan Borning at Computer Science and Engineering, University of Washington, and PhD student Shiri Azenkot at Computer Science and Engineering, University of Washington.

7. REFERENCES

[1] Azenkot, S., Ladner, E. R., and Wobbrock, O. J. 2011. Smartphone Haptic Feedback for Nonvisual Wayfinding. DOI=http://www.cs.washington.edu/homes/shiri/papers/azenkot_assets2011.pdf .