Design Decisions in the RideNow Project

Rick Wash
School of Information
University of Michigan
rwash@umich.edu

Libby Hemphill
School of Information
University of Michigan
libbyh@umich.edu

Paul Resnick
School of Information
University of Michigan
presnick@umich.edu

ABSTRACT

The RideNow Project is designed to help individuals within a group or organization coordinate ad hoc shared rides. This paper describes three design decisions the RideNow team made in order to allow incremental adoption and evolution and to capitalize on local conditions. (1) The system allows users to interact with the system through email or Web, because we anticipate that email will be most convenient when there are few users but the Web interface will be more useful as the number of users increase. (2) The system does not force structure on user-entered data such as dates, times, and locations, instead allowing conventions to emerge. (3) We use the group's shared physical spaces to provide additional information about ride sharing activity.

Categories and Subject Descriptors

D.2.2 [**Software Engineering**]: Design Tools and Techniques

General Terms

Design

Keywords

Social Computing, Design Methods

1. INTRODUCTION AND MOTIVATION

In the United States, there is tremendous unused transportation capacity in the form of unoccupied seats in private vehicles. Most vehicles can comfortably seat four or five, but rarely carry more than one[7]. Filling some of those empty seats could create tremendous benefits both to individuals and to society as a whole. Many riders would get to destinations more conveniently. Drivers and passengers sharing a ride might enjoy the pleasure of conversation. If riders would otherwise have taken their own vehicles, the total number of miles driven would decline. That would reduce congestion for everyone using the roads and reduce smog and other negative externalities resulting from fuel consumption.

There are, however, several major barriers to ride sharing. First, it is difficult to coordinate routes and schedules, especially if people require flexible schedules. Uncertainties about coordination compound the problem, especially for riders: a one-way ride with the possibility of being stranded is of limited utility. Second, there are risks from riding with strangers, ranging from social discomfort with sharing what are currently thought of as private spaces to differing expectations about cleanliness or driving habits to miscommunication about routes and schedules to verbal or physical assault. Third, much as users have refused to adopt some office groupware systems because those paying the costs were not the primary beneficiaries[3], ride sharing may not occur if the individual participants do not benefit. Riders may benefit while drivers incur costs. Both may experience inconvenience while others reap the benefits of reduced congestion and smog.

To overcome these barriers, in the United States local governments and employers have instituted services and policies to reduce the coordination costs associated with ride sharing and to increase the benefits for those who participate. Matching services help people form car pools. Parking lots are available near major highways so that members of the car pool can drive to a meeting point, thus reducing daily coordination costs. Many large employers provide subsidies and preferential parking to employees who form car or van pools. In some urban areas, congested highways have High Occupancy Vehicle (HOV) lanes reserved for cars with multiple passengers. Regular car pools with the same passengers also overcome the risks of sharing a ride with strangers, since after the first few rides the participants are no longer strangers.

Efforts to encourage commuters to form car and van pools have been somewhat successful, though only at the margins. According to one survey, just 5% of people used an organized carpool at least once per month while 14% used some form of public transportation during the same period[8].

In a few cities, such as Washington, D.C.[1] and San Francisco[2], "instant" ride sharing has emerged among strangers. To take advantage of HOV lanes during peak commuting hours, drivers want to take passengers. Passengers line up at designated places; drivers pick them up and drop them at a fixed destination. Instant ride sharing reduces the coordi-

[1] http://www.slug-lines.com
[2] http://www.ridenow.org

nation costs of fixed schedules inherent in regular car pools. On the other hand, instant ride sharing is less convenient because it does not provide the last mile of transportation at either end of the commute.

Several computer-mediated services try to coordinate ad hoc ride sharing among strangers. For example, in the San Francisco Bay Area, Dan Kirshner's Ride Now service is trying to match people dynamically for the last mile from their homes to and from the subway station, where parking is limited. In many cities, the popular site craigslist.com includes a ride sharing message board.

We have designed a service, also called RideNow[3], that takes a somewhat different approach. Rather than trying to serve a region's whole population, who may have nothing in common other than traveling in the same direction, our service is intended to coordinate ride sharing within groups or small organizations of a few hundred to a few thousand people. In doing so, we forego some of the potential benefits of scale; in particular, there may be no one driving when and where someone wants to go. On the other hand, the risks of dealing with strangers are reduced, and the potential benefits of conversation are increased. And given the well-known in-group biases[5], we expect that drivers will be more motivated to help members of their own organization than random others.

2. DESIGN APPROACH

Our design approach tries to capitalize on two opportunities that arise given our focus on ride share coordination within groups or organizations. First, development and usage of the system can occur incrementally, as there will be less need for an initial critical mass of usage. Second, the system can draw on features of the local situation, including technical infrastructure, social norms, naming conventions, and shared physical spaces. These two opportunities, incrementalism and tailoring to local needs and resources, were the basis of our design approach.

2.1 Incrementalism

As evidenced by other similar ride sharing projects, many conventions, both social and technological, will likely develop as more users begin to use the system for a variety of purposes. We designed the system to allow incremental changes to occur. Incremental development should build user momentum for the system and avoid the costs of over-engineered solutions[2].

By choosing an incremental approach, we allow users to indirectly determine the features that get developed and added on to future implementations of the system. For example, users might want to specify routes rather than just destinations, and they might want to specify recurring ride requests, such as "any weekend." Rather than trying to anticipate and prioritize all these needs, we developed a bare bones system initially and plan to watch the patterns of usage and feature requests that users make. Our goals were to avoid technological constraints on future expansion and to provide an interface that encourages new uses and doesn't unnecessarily constrain the users.

[3]The RideNow name is a registered service mark of Ride Now, Inc. (`www.ridenow.org`). Thanks to Dan Kirshner for inventing such a great name and for giving us permission to use it. Dan is conducting other tests of ride coordination services. We share a name and the goal of promoting ride sharing, but our projects are independent.

2.2 Localization

Scalable, general systems do not allow nuance, are not socially flexibile, and do not allow for ambiguity[1]. Shirky describes very localized systems developed in the context of NYU's Interactive Telecommunications Program. Those systems capitalized on shared physical spaces and well established communities to create personalized, situated software that had direct ties to the community[6]. Our goal with RideNow was to design something similar for our own initial test community but still remain flexible enough to expand our system to include other communities in the future.

Our design allows communication in the system to be nuanced and ambiguous; the offline community will supply the contextual cues necessary for successful interpretation of ambiguous information in the system. For example, our target community has shared knowledge about community-wide events and locations. We allow users to describe ride requests and offers using vocabulary that may be understandable only within that community.

Our target test community also has shared physical spaces. By placing a monitor in a hallway near one of the busiest entrances to office spaces, we situate the application in its social context. When entering and leaving the building, potential users are exposed to RideNow's functionality and are able to quickly see how much traffic the system is receiving. This shared display allows people to scan for rides requested or offered without checking email or logging onto the website and may even encourage immediate ride sharing among people who run into one another in this shared hallway.

3. DESIGN DECISIONS

The architecture of the RideNow system consists of one backend database and two frontends: an email list and a website. The backend database is a MySQL database that stores data about rides, users, locations, and all of the messages that have been sent to the system. The email list is a standard Mailman mailing list with a bot subscribed that can parse semi-formatted messages and insert them into the database. Users can use the raw email messages to get the information, and the format is simple for users to compose. The web interface presents ride information in a sorted format, and provides forms users can complete to submit new ride requests and offers.

3.1 Email and Web Interfaces

Email and Web have different interaction capabilities, and therefore allow different types of interactions with the data. One advantage of the email interface is that ride sharing messages can be harvested from email lists that are multi-purpose, which may help achieve critical mass. Initially, a group that already communicates through an email list would simply begin to send ride requests and offers, in addition to their other messages.

We anticipate that, as use increases, managing RideNow email will become cumbersome for users, since most messages will interest only a few people. The Web interface offers additional functionality for interacting with the system. Figure 1 shows how rides offered are displayed in the Web interface. The table is sorted on Date and Time, rather than by the time when the information was entered into the

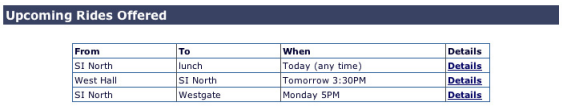

Figure 1: Rides Offered list

system, as an email archive would. Expired offers and requests are not displayed.

The current email and Web interfaces allow users to (1) request a ride, (2) offer a ride, and (3) thank a driver or passenger; however, we can imagine features such as ride matching and mapping that could be useful. Our incremental approach allows us to release the RideNow software before spending development resources on such complex features that may never see actual use. Releasing the software without these advanced features limits our initial investment and reduces the risk of over-engineering. Adding features to later implementations allows us to sustain momentum for participation and to recruit new users.

In order to allow users to interact with the data in their preferred interface, it is important that the basic functionality of all interfaces be the same and that the data backing these interfaces is the same. RideNow ensures this consistency by relying on the email list for distribution of information. For all ride requests, ride offers, and thank you messages created through the web interface, the system creates an email and sends it to the list. A user could be an active member without ever visiting the RideNow website; he could choose to use RideNow only as an email list and simply read, post, and respond to ride offers and requests without leaving his email program.

By having an email list as the backbone of the system, we make it possible for multiple lists to develop. People will be able to subscribe to multiple lists; for example, they may join one list with their coworkers and another with their volunteer organization. Because the underlying architectures are identical, such lists could be federated if their users desire. This setup allows organizations to establish their own implementations of the RideNow system; each implementation is localized for the group that uses it. Using an email list also provides a security advantage; the database is add-only and accepts input only from the list. This setup makes it easier to set privileges and reduces the risks associated with providing read/write database privileges to individuals.

3.2 Semi-Structured Data Fields

Long ago, Malone et al[4] explained the flexibility and power that semi-structured data fields provide: input that a computer can understand can be used for sorting or filtering, but unparsed text may still be understood by human readers. We treat times and locations in this way. For example, in the "Be a Passenger" form (Figure 2), users are able to enter when they want a ride in a free-form text box. The system attempts to parse what it can out of this 'unstructured' data field into a 'structured' date/time range. This parser can understand simple phrases like 'tomorrow afternoon' to mean anytime tomorrow between noon and 5pm, or 'next week' to be anytime after sunday at midnight before next sunday at midnight. Once the system has this range (a start date/time and an end date/time), it displays it for the user to validate. The user can either accept the interpretation,

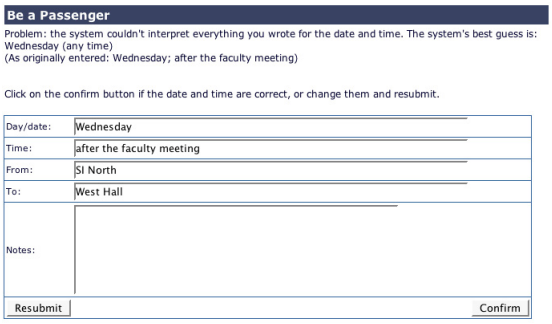

Figure 2: Be A Passenger form

or modify the free text entry and try again. We do this to verify that the system has interpreted the user's intent correctly and to help users become familiar with the capabilities of the parser. Unparsed or partially parsed inputs are still acceptable, though. (An example of partially parsed input would be 'Wednesday after the faculty meeting,' from which the system can only understand 'Wednesday,' as illustrated in Figure 2)

The structured time data has three main uses. First, we use the beginning of the structured range as the sorting criteria. Rides whose start time is earlier are placed higher on the list since they are available sooner. Secondly, we use the end time of the structured range for filtering the list. Any ride whose end time is in the past does not appear on the list since it is no longer relevant. Finally, the structured data is converted on the fly into a more natural language for display to the user. This feature is really useful when, for example, the user enters a relative date like 'tomorrow.' If this ride request is viewed 24 hours later, the system will display the date as 'today' rather than the now-inaccurate 'tomorrow' that was entered.

As long as the unstructured input was parsed correctly, the above will work fine. However, since we cannot parse all useful inputs, we have some rules in place for dealing with unparsed or partially parsed inputs. The sorting and filtering remain the same, with default start/end times used for those if absolutely nothing could be parsed out of the input. When attempting to display an unparsed or partially parsed input for the user, we display the original unstructured input and append in parentheses the date that the input was submitted. This allows other users to see what the original submitter wrote and hopefully allows them to figure out its meaning. Also, if the original submitter used a relative date and time, then it is still possible to figure out what they meant since users will have the date/time at which the original entry was submitted.

Currently, the 'When:' field has a default of 'today anytime.' We hope to generalize this default by adding a dropdown list of common entries. This list will function both as suggestions to new users, and an easy entry interface for common entries for repeat users.

Semi-structured data fields not only offer flexibility, they also enable incremental improvement over time as usage patterns and the system co-evolve. Users may learn what the system is able to parse through trial and error and by seeing the formatted displays that the system generates (e.g., "this afternoon"); developers can update the parser to handle frequently used phrases.

3.3 Displays in Shared Physical Spaces

Most members of RideNow's target community share physical spaces such as hallways, copier rooms, etc. The community is spread out over a number of campus buildings and departments. In order to capitalize on these shared spaces, RideNow uses a TV monitor in a heavily trafficked hallway to display ride sharing information. The monitor displays a rotating webpage that lists rides offered, rides requested, and highlights from rider and driver Thank You notes.

After their ride offer or request has expired, users are asked to provide feedback to their driver/passenger on the experience. This feedback is sent to the email list as a Thank You, and messages stored in the database are randomly selected to be shown on the shared display. RideNow's Web interface allows users to send these Thank You notes through the web as well.

4. FUTURE RESEARCH

This paper has described three critical design decisions the RideNow team made in designing the system. Future research on RideNow will examine four aspects of this system: modular development, conventions of use, motivation, and social capital implications.

The system has a modular structure to allow incremental changes to occur as users find uses for them. In particular, we are interested in providing functionality for locating destinations, mapping routes, and matching interested parties based on proximity to suggested routes. Additionally, we are looking at additional modular interfaces to the system, such as cellphone or PDA based interfaces to search for and submit rides.

We are also interested in the conventions of use that emerge. As people observe how others fill in the free-text fields, conventions will develop as to abbreviations and level of specificity. We are interested in how these conventions are established and how this development can be influenced, encouraged, or discouraged.

We are interested to determine what motivates users, especially drivers, to participate in ride sharing programs. Without the HOV lanes as motivation, drivers may be reluctant to participate. For what kinds of groups are "Thank You" messages a sufficient motivator? We are also interested in exploring the development of trust in this community. What allows people to trust the system and community enough to get in the car with a stranger?

Finally, we are interested in how this system can be used to develop social capital. Can we match people for rides by interest areas such that they can then have interesting discussions during the rides? Would people be willing to use the system for that purpose?

5. ACKNOWLEDGEMENTS

This material is based in part upon work supported by the National Science Foundation under Grant No. 0325837. Additional financial support was provided by Microsoft Research. Discussions with Marc Smith have significantly influenced the design approach and the technical architecture described in this paper.

6. REFERENCES

[1] M. S. Ackerman. The intellectual challenge of cscw: The gap between social requirements and technical feasibility. *Human-Computer Interaction*, 15(2/3):179–203, 2000.

[2] R. G. Fichman and S. A. Moses. An incremental process for software implementation. *Sloan Management Review*, 40(2):39–52, 1999.

[3] J. Grudin. Groupware and social dynamics: Eight challenges for developers. 37(1):92–105, 1994.

[4] T. Malone, K. R. Grant, K.-Y. Lai, R. Rao, and D. Rosenblitt. Semi-structured messages are surprisingly useful for computer-supported coordination. *ACM Transactions on Office Information Systems*, 5(2):115–131, 1987.

[5] R. L. Moreland and J. M. Levine. Socialization in organizations and work groups. In M. E. Turner, editor, *Groups at work: Theory and research*, pages 69–112. 2001.

[6] C. Shirky. Situated software, Mar. 2004. available at `http://www.shirky.com/writings/situated_software.html`.

[7] U.S. Department of Transportation (USDOT), Bureau of Transportation Statistics (BTS). Highlights of the 2001 national household travel survey, 2001. available at: `http://www.bts.gov/`.

[8] U.S. Department of Transportation (USDOT), Bureau of Transportation Statistics (BTS). Omnibus survey household survey results, Oct. 2004. Available at `http://www.bts.gov`.