OneBusAway: A Transit Traveler Information System

Brian Ferris[1], Kari Watkins[2], and Alan Borning[1]

[1] Computer Science & Engineering, University of Washington, Seattle, WA
[2] Civil & Environmental Engineering, University of Washington, Seattle, WA

Abstract. Public transit is an important tool for those looking to ease their commutes, reduce their car dependence, or perhaps minimize their environmental impact. Unfortunately, the usability of transit systems often leaves much to be desired, to the point of deterring new riders. Tools on the web and mobile devices are increasingly being used to help tame confusing transit systems. OneBusAway is one such set of tools, providing access to real-time transit information for Seattle bus riders through a variety of interfaces, including web (http://onebusaway.org), phone, SMS, and mobile devices. We describe the current system, and then discuss current and planned research that builds on it to use increasingly-powerful smart mobile devices to provide location and context-aware tools for navigating transit systems.

1 Introduction

Public transit systems play an increasingly important role in the way people move around their communities. By helping travelers move from single-occupancy vehicles to transit systems, communities can reduce traffic congestion and the environmental impact of transportation. While there are significant benefits to using transit, many choice riders (that is, riders for whom transit is not the sole option) are ultimately reluctant to make the switch. Riders are often confused or intimidated by the complexity of large transit systems. Transit agencies often do themselves no favors by failing to provide information about the systems they maintain in simple, understandable ways.

Increasingly, smart mobile devices are being used to help manage the complexity of using transit. Whether it be a simple phone or SMS interface, or a more complex native mobile application, these systems can provide schedules, routes, real-time arrival information, and service alerts to users where they need it most: out and about, using their transit systems. We refer to these tools that help riders understand their transit systems as "transit traveler information systems." We already have some experience working on these systems, as we run a system called OneBusAway (http://onebusaway.org). OneBusAway provides real-time transit information and commuter tools for Seattle-area bus riders through a variety of interfaces, including web, phone, SMS, and mobile devices (see Figure 1).

S. Ystad et al. (Eds.): MobiCASE 2009, LNICST 35, pp. 92–106, 2010.

Fig. 1. The current OneBusAway mobile interfaces show real-time arrival information for both Nokia (left) and iPhone (right) platforms, among others

In the remainder of this paper, we first describe related work in the area of mobile transit tools. We then describe the OneBusAway system as it currently exists, and conclude with directions for current and future research that build on it.

2 Related Work

Displays that provide real-time arrival information for buses, subways, light rail, and other transit vehicles are now available in a significant number of cities worldwide at places such as rail stations, transit centers, and major bus stops. However, it is likely to be prohibitively expensive to provide and maintain such displays at (for example) every bus stop in a region.

With the increased availability of powerful mobile devices and the public availability of transit schedule data in machine readable formats, there have been a significant number of tools developed to improve the usability of public transit, especially mobile tools. One motivation is that, as noted above, it is unlikely that real-time transit information will be available on a public display at every stop. Another is that personal mobile devices can also support additional, personalized functionality, such as customized alerts.

One of the first online bus tracking systems, BusView, was developed by Daniel Dailey and others at the University of Washington [8]. More recently, Google Transit, which was started as a Google Labs project in December 2005 [7], is now directly integrated into the Google Maps product and provides transit trip planning for more than 405 cities around the world [5]. In addition to providing trip planning through a web-based interface, interfaces to Google Transit exist on a variety of mobile devices as well, making use of location sensors such as GPS and WiFi localization on the device to improve the usability of the transit app.

While Google Transit has been useful to transit riders around the world, it is also significant for establishing a *de facto* standard for exchanging transit schedule data: the Google Transit Feed Specification [6], or GTFS for short. The upshot is that many of the transit agencies participating in the Google Transit program have also released their transit scheduling data in the GTFS format for third-party developers to work with. Development ecosystems have grown out of the public availability of this data. The Portland TriMet third-party applications page [10] lists over 20 applications using Portland's transit data, many targeted at providing transit data on mobile devices. Similar ecosystems exist in San Francisco and the Bay Area, Chicago, and other major cities.

An example of a mobile application that makes use of GTFS transit data is the Travel Assistance Device (TAD) developed at the University of South Florida [1]. The TAD uses the GPS on a mobile device to detect the current location of a bus rider and to prompt the rider when his or her stop is near. Routes and desired stops are manually entered into the system for later detection. The application is specifically targeted at riders with cognitive impairments to increase the usability of public transit for these users.

Another example of a mobile application to improve the usability of public transit can be found in previous work at the University of Washington. The Opportunity Knocks system [9] also provides a mobile application to provide cognitive assistance to transit riders. Like the TAD system, the Opportunity Knocks system uses GPS data to model a user's current location. Unlike the TAD system, the Opportunity Knocks system automatically detects the user's current mode of transportation from GPS traces and learns the important places a user travels to, such as home and workplace, without manual labeling. Based on these learned models, the application can automatically predict with high confidence where a user is headed given only a small amount of tracking data, correcting the predicted destination as more data becomes available. Additionally, the system can detect when the user does something unexpected, such as forgetting to get off the bus at the regular stop, and then automatically notify the user. The main drawback to the Opportunity Knocks system is that it does not currently run on-line. That is, a week or two of GPS data must be collected, at which point a model of the user's important places and travel patterns are learned and used by the system. The system does not update this model when new places are added or travel routines change.

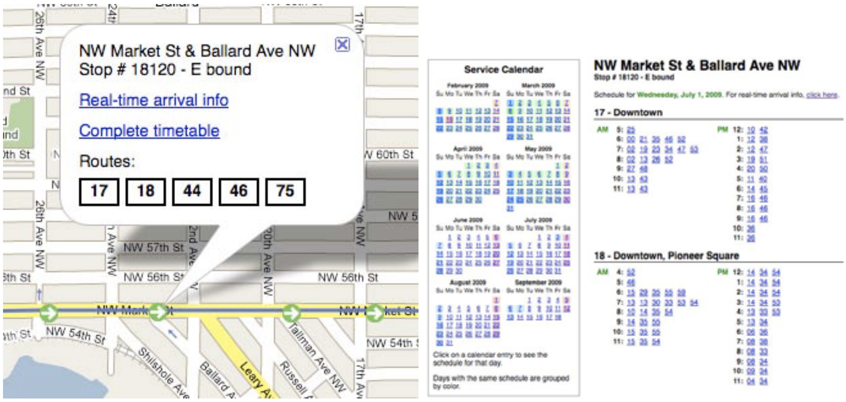

Fig. 2. Web-based route map and timetable interface, with routes and stops displayed on a map with direction of travel indicated. The full schedule for the stops is listed as well.

3 The OneBusAway System

In this section we describe the components that make up the OneBusAway transit traveler information system, along with its history and current usage.

3.1 Route Maps and Timetables

At a bare minimum, a modern transit website must provide static route maps and timetables to users. Our system is no different, though we enhance the usability of this static data by presenting it in novel ways, including a variety of Web 2.0 enhancements to make searching for stops, routes, and trips easier. Figure 2 shows an example route map and timetable. Routes and stops are displayed in a visual maps interface, with stop travel-direction indicated on the map, and routes servicing a stop shown in a pop-up dialog. When the timetable for a stop is examined, we display the complete service calendar, highlighting different service due to weekends and holidays. We also display specific route timetables in stem-and-leaf format to highlight frequency of service over the course of the day.

3.2 Real-Time Tracker

As with trip planners, real-time tracking has become accepted as an integral part of transit user information. The ability to determine when the next vehicle is coming brings travelers' perception of wait time in line with the true time spent waiting [3]. Transit users value knowing how long their waits will be or if they have just missed the last bus.

As noted in the related work section, one of the first online bus trackers was BusView, developed by Daniel Dailey and others at the University of Washington. We have built upon this system to develop a more user-friendly version of

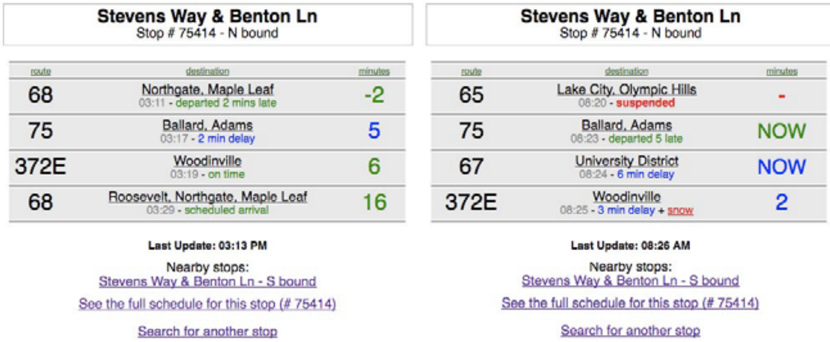

Fig. 3. Web interface to real-time arrival information. On the right, see example service alerts indicating cancellations and temporary reroutes due to snow-related adverse weather conditions.

the interface, available on the OneBusAway website. It includes various interfaces to real-time tracking data, including a telephone number users can call to have arrival information read to them, an SMS interface for receiving arrival information as text messages, a standard web interface (see Figure 3), and a website optimized for internet-enabled mobile devices.

3.3 Service Alerts

While fixed transit schedules change infrequently, the world in which those schedules must be kept is in constant flux. Temporary incidents such as construction, detours, accidents, severe weather, or special events often mean temporary service modifications in the forms of reroutes, skipped stops, or canceled service. Keeping users informed of these temporary service modifications is an essential task for any transit agency, but most traveler information systems do not use the full complement of communication modalities to notify their users. For this reason, service alert notification is a major component of our system, with an eye towards tight integration with the route timetables, trip planner and real-time tracker components so that service alerts are pushed to users across all communication channels.

Our current service alert infrastructure was put to the test during a major snowstorm in the Seattle area last winter. For more than a week, upwards of half the bus service in the Seattle area was cancelled due to icy road conditions. What service remained was often on detour to avoid iced-over hills or stalled vehicles. With conditions changing rapidly, it was difficult to keep riders up to date on cancellations and reroutes. We used OneBusAway to keep riders notified of service status when they accessed information for a particular stop on the web (see Figure 3) or through the phone system.

In the future, we wish to work on supporting better standardization of machine-readable service alert information from transit agencies, so that additional agencies can be more easily supported. For agencies that cannot provide automated machine-readable service alert information, we are also considering alternatives such as crowd-sourcing, which would allow riders to share information about service changes as they happen. As with any crowd-sourcing solution, issues of trust and verification would play an important role.

3.4 Trip Planner

Trip planners use an origin and destination address to search for one or more scheduled trips that travel between the two according to the desired time-frame of the traveler. Trip planners have become relatively common with larger transit agencies in the past several years. Although these trip planners are useful tools, they remain closed-source. This means that programs expanding and extending their content and functionality cannot be easily developed. As part of OneBusAway, we developed our own trip planner engine so that we might explore interesting planning applications.

One example of such an application is the Explore tool on the OneBusAway website (Figure 4). The Explore tool is a nearby attractions search tool, which

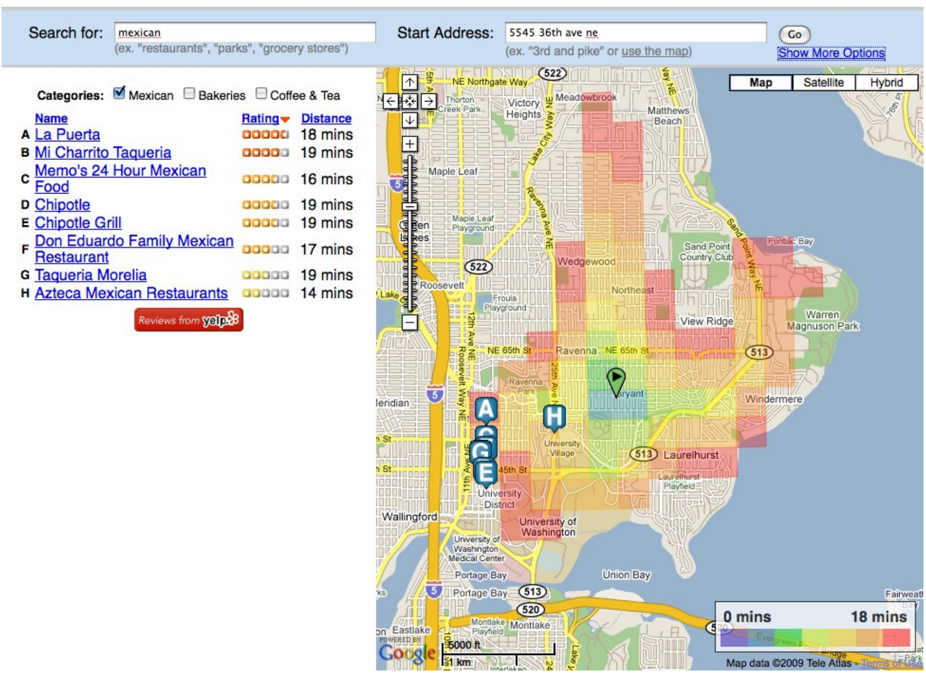

Fig. 4. The Explore tool. This search shows areas reachable in less than 20 minutes by transit from a given starting point, Mexican restaurants in that area, and the Yelp ratings for those restaurants.

attempts to answer another common transit rider question: “I’m looking for a nearby restaurant / park / library that’s close by when taking public transit. What are my options?”

For first-time and infrequent riders who are not familiar with what is accessible using their local routes, this can be a difficult question to answer. The Explore tool aims to make answering that question easy by combining the functionality of a trip planner with online databases of local restaurants, shopping, and other amenities. In our implementation, users specify their starting points along with what they are interested in searching for. Optionally, they might specify additional features, such as maximum trip length, number of transfers, or walking distance. When the search is submitted, we compute the total area reachable by transit given the specified constraints and then begin searching for local businesses and attractions as specified by the user in the target area.

In the interaction shown in Figure 4, the user has searched for nearby Mexican restaurants within 20 minutes by transit from home. The display of results includes the name of the restaurant, the average rating for that restaurant, and the minimum travel time to the restaurant, along with a display of all the results on a map.

Once a user has settled on a particular restaurant, he or she can select it for more information, including location and upcoming transit trip plans to that destination. Our prototype includes data from the Yelp (http://yelp.com) online database of reviews.

3.5 Interface Modalities

OneBusAway is accessible through a variety of interfaces. As mentioned in previous sections, the various tools described above can alternatively be accessed through the website, an interactive-voice-response (IVR) phone system, SMS text messages, mobile-optimized web pages, and native apps for popular mobile devices, including Nokia and iPhone (Figure 1). The native mobile apps are of particular interest because of the potential to integrate location sensing technologies, such as GPS and WiFi-localization, with the real-time transit information system (Figure 5). In the other interface modalities, much of the interaction involves trying to determine where the users currently are and what routes and stops they are interested in. With the mobile apps, the location information can make narrowing the context of interest much easier, so that relevant information can be found more quickly. As we describe in the next sections, the new affordances made possible by new mobile phone technology drive many of our new research applications.

3.6 History and Current Use

OneBusAway was originally developed by the first author after too many late nights wondering if the route 44, a notoriously unreliable bus in Seattle, would ever come. Despite not yet being an official service of the local transit agency, the website receives about 3500 visits per day, and the telephone interface about

Fig. 5. Using a GPS sensor to automatically find nearby stops

1500 calls. The system has also received considerable publicity, including news stories and discussion on local transit blogs. We are currently discussing a more official status for the system with transit providers in the Puget Sound region, and also adding additional functionality.

Other transit agencies in Washington State and elsewhere in the U.S. have expressed considerable interest in the system as well. The system is open source, under the Apache 2.0 license. This supports one of our longer-term goals for deployment, namely, making the system easily available and customizable by different agencies. A closely related goal is to facilitate standardizing on an open standard for real-time and other transit information, to allow a rich set of transit applications to be prototyped and deployed.

3.7 Implementation

OneBusAway follows a standard multi-tiered architecture common to many web application projects. There are a few wrinkles, however, due to some of the large data structures involved in order to support fast trip planning, which we describe below.

One of the biggest implementation challenges in building OneBusAway was the trip planner engine. The first challenging aspect is the basic algorithmic details of building a "correct" trip planner. We represent transit data as a directed graph structure. Nodes are instances where a user could board or exit a transit vehicle at a particular time and stop, while edges are transitions between nodes as a transit vehicle moves along its route. There is a similar graph for street network data used to provide walking directions between stops. At a high-level, trip planning is just a directed-graph shortest-path problem over these graphs, but the implementation details of handling the various nuances of transit systems are quite complex.

The second challenging aspect is achieving high performance in the trip planner. Reasonable response times for modern websites are measured in the hundreds of milliseconds, so our trip planner engine must be able to very quickly compute trip itineraries. A common approach to speeding up computation is to keep the entire transit graph structure in memory. Keeping the transit graph in a standard database would introduce too much latency as nodes and edges in the graph are visited. For single-origin to single-destination planning tasks an A* directed shortest-path search can very quickly find a solution. The single-origin to any-destination planning task, as illustrated by the restaurant example described in 3.4, is more computationally complex. While essentially Dijkstra's algorithm is used to explore the graph, careful pruning and accounting is performed to limit the required search space.

The space complexity of the memory resident transit graph limits the size of trip planning tasks that can reasonably be attempted. A larger transit agency, such as King County Metro in Seattle, might have order-of-magnitude one million nodes in its transit graph and a memory foot print of 50-100 MB. While it is reasonable to build a graph for all the transit agencies in the larger Puget Sound region of Washington to support integrated transit planning, building and maintaining a combined transit graph for larger areas, such as the entire United States, is currently not attempted.

So that OneBusAway might grow to support data from other transit agencies, whether they be in the same county or around the world, we have adopted a federated data architecture that separates transit data into related regions that can be housed separately. The various transit agencies that make up the Puget Sound region of Washington might be combined into one data region, while the agencies of the Bay Area of California might make up another.

These data regions are represented by individual server processes exposed by a standard API using remote method invocation. Additionally, the various user interface components, such as the web interface, the interactive voice response (IVR) phone interface, and the external API interface are broken up into individual server instances as well, as shown in Figure 6. Each interface instance communicates with the appropriate transit data instance based on the particular region a user is interested in. The strong decoupling of the various instances allows for robust replication and failover across a cluster of machines.

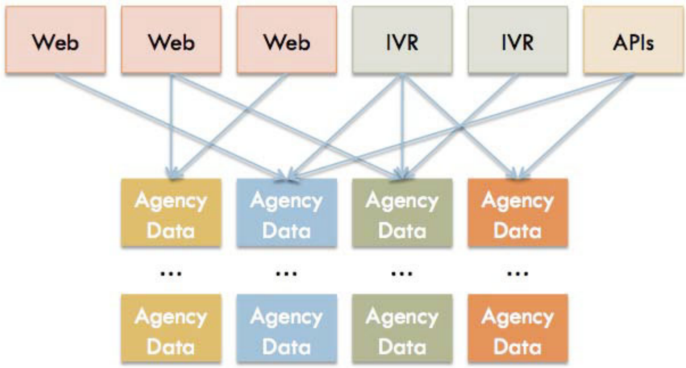

Fig. 6. OneBusAway architecture diagram. There is a strong decoupling between data sources and interface services, so that each service can be run as separate processes to support replication and failover across multiple machines.

OneBusAway is written in Java and uses a variety of standard open source development libraries and frameworks in its implementation. The system is composed of a number of service modules, each providing specific functionality, which are coupled together using the Spring inversion-of-control framework. Java object persistence to a relational database is handled by the Hibernate framework. The Tomcat servlet container combined with the Apache Struts MVC web framework does the bulk of the heavy lifting for web-based publishing. Client-side AJAX applications are written primarily using Google Web Toolkit, which compiles Java source code into optimized Javascript. For our telephony system, we use the Asterisk PBX server to pass incoming calls to handling code using the FastAGI interface. The only piece of non-open-source software in the entire system is our text-to-speech engine, which we license from Cepstral.

As mentioned, OneBusAway is open source software licensed under the Apache 2.0 license. The source code for the project, along with further implementation details, APIs, and documentation, can be found at our project site on the web: http://code.google.com/p/onebusaway/.

4 Current and Future Research

In addition to the goal of wider deployment of OneBusAway, we are actively working on a set of research questions opened up by the availability of this system as a base.

4.1 Automatic Learning of User Travel Patterns

We are working on two distinct types of travel pattern learning. The first type involves learning long-term models of riders' travel patterns as they interact with our tools, including which transit routes and stops they use on a regular basis, so that we might opportunistically notify them of service changes for future trips. The second type involves learning models for inferring the real-time transportation mode of a rider, so that we might advise them actively for a transit trip already in progress. This learning is similar to that of Opportunity Knocks [9], but we are extending the models to update continuously over time, so that changes in travel patterns can be integrated automatically.

To support both types of learning, we need data from riders about transit usage. To gather this information, we have instrumented the various tools in the OneBusAway system to support logging of key events. Collecting such data can of course reveal much personal information, so at this time we are only logging such event data for ourselves. In the future, we plan to also offer an instrumented version of OneBusAway to people who want to help us with user studies.[1]

As one example of data collection, every time a rider with the instrumented version of the system accesses real-time information at a particular bus stop, whether it be on the website, the IVR phone system, SMS, or a mobile app, we make a record. For our mobile app, we also log location and accelerometer data for supported devices. Riders also annotate their current activity, noting when they are walking, waiting for a bus, on a bus, biking, in a car, or stopped.

From this collected data, we can begin to build models of travel patterns. For long-term usage models, we collect simple frequency statistics about which stops and routes the rider typical uses. We also wish to build models of the places riders travel to and from, including where they live, where they work, and other places they frequently visit. These models are slightly more complex to build. While trip planner usage provides activity traces with clear start and end destinations, the high frequency trips that make up the bulk of travel for transit riders usually do not have trip planner activity traces because the rider is familiar enough with the trip to travel without explicit travel directions. Instead, we receive a number of real-time arrival data requests for the origin stop when a rider begins a trip, without any immediate indication of the final destination. Only by reasoning about sequences of these real-time arrival data lookups can we see the origin for a trip and then the destination when the rider takes the next trip.

Of course, there is no guarantee that the next stop accessed in a sequence is the actual destination, as a rider might have moved without leaving an activity trace in the meantime. Because of this ambiguity, we will make use of probabilistic inference to model the uncertainty in our modeling of travel goals. We will continue our work with directed and undirected graphical models such as dynamic Bayesian networks, conditional random fields, and Markov logic networks to tackle this modeling problem.

[1] We will follow standard Human Subjects protocols for such studies, including obtaining informed consent from participants and safeguarding the privacy of participant data. The version of OneBusAway offered for general download would of course not do any such logging, for privacy reasons.

In addition to long-term usage models, we also wish to build models for inferring the current transit activity of the rider. From the annotated activity traces of location and acceleration data, we build a simple activity model to recognize walking, waiting, riding a bus, riding a bike, riding in a car, and the default state of stopped. We are working to use a boosted classifier to take features computed from location and accelerometer data to build a local movement classifier. When we detect that a rider is in motion we will use high-level knowledge of transit routes, locations of streets, statistics about travel speed and path to infer the transportation type, jointly inferred and smoothed with a conditional random field.

All told, the focus of these models is to build and maintain an active model of how a rider typically uses transit and how he or she is actively using it right now. These models will be key to implementing our next application area.

4.2 Automatic Notification

When everything is going right, riders do not need a lot of help from their transit tools. Buses arrive on time and go where they are expected to. However, everything does not always go right. Buses and trains might be running late or not at all. Some routes might be redirected due to construction, weather, or accidents. A rider may not be familiar with a route, and may not know which stop to get off at, or what to do if he or she misses a bus. Ideally, when things go wrong, we can notify riders and help them pick appropriate actions.

Realizing this functionality requires two components: the models of travel, both historical and real-time, discussed in the previous section, and actual data from the transit agencies describing the status of their systems. The latter is actually trickier than it sounds. Only a handful of agencies in the United States have real-time tracking capabilities for their fleets. Even fewer push automated service alerts to their riders. We are actively working with agencies in our area, including King County Metro, Sound Transit, and Pierce Transit, to automate publication of services alerts in machine-readable standard-compliant formats so that they can be consumed by external systems such as OneBusAway.

Armed with accurate and timely status updates from the underlying transit systems, we can begin to integrate this information into trip-planner and real-time arrival information. However, additional services become possible when we consider pro-actively notifying riders about changes in the transit system. An example scenario would be automatically notifying a rider and suggesting an alternative when the bus he or she typically takes to work every morning is cancelled due to technical problems. This example combines our models of a rider's typical transit usage with real-time status updates in an innovative and useful way. We can also perform notifications once a trip has already begun, automatically detecting the rider's current mode of transit and providing assistance when we detect that the rider has missed the bus, did not get off at the proper stop, or got on the wrong bus.

4.3 Value Sensitive Design

At first glance, public transit might seem to be simply a tool for getting from point A to point B, and not have any particular relation to questions of human values. However, we believe this is not the case — rather, public transit touches on a number of important human values, including fairness, access, safety, health, community, social inclusion, and environmental sustainability. We will use Value Sensitive Design [4], a design methodology methodology for information technology systems that seeks to account for human values in a principled and comprehensive manner throughout the design process, as a way to help investigate this complex design space. The overall transit-using population (both current and potential) is a diverse group with diverse needs, many of which might be supported in different ways with technological solutions.

One specific topic that we are beginning to investigate is helping make transit information systems more accessible to blind and low-vision riders, and also deaf-blind riders. These riders are usually highly dependent on transit for travel within their communities, but transit traveler information systems can be difficult to use for the blind. We will start by conducting a set of focus groups and semi-structured interviews with blind and deaf-blind riders to help understand their values and needs, along with user studies of the OneBusAway interface, initially to simply improve the accessibility of OneBusAway for these riders, and subsequently to investigate tailoring information to their needs and interests. Combined with the increased capabilities of modern mobile phones, some truly innovative solutions may be possible.

Another topic is personal safety. This is an important value that influences the usage of transit: for example, some riders are uncomfortable waiting alone for a bus late at night. Our system might help support this value by providing real-time arrival information so that riders could minimize their wait time or by suggesting alternate stops that other riders report to be safer.

4.4 Transit Travel Behavior

Our primary goal in developing mobile transit tools is to help make transit more usable and enjoyable for a diverse range of riders. A secondary goal is to collect transit usage histories to help inform models of how people use both public transit and other transportation modes on a macro scale. One of the projects we are actively involved with is UrbanSim [2,11], a software-based simulation model for integrated planning and analysis of urban development, incorporating the interactions between land use, transportation, and public policy. UrbanSim includes a set of component models that simulate different actors and processes in the urban environment, such as residents looking for a place to live; real estate developers constructing new or renovated houses, offices, and other structures; and others. In its current implementation, UrbanSim is coupled with an external travel model, typically a commercial Four Step travel model.

One important area for improvement for UrbanSim is in the travel model. We want to replace the current external travel model with an activity-based travel model that we implement and maintain. An important aspect of calibrating and improving such models will be gathering richer information about people's travel behavior, including modeling under what circumstances a user will make a trip using transit, a personal vehicle, biking, or walking. This is one of the motivating reasons for being able to distinguish between travel using transit, bike, car, or foot in our travel classifier. The activity traces and transit models learned using our tool can help us better understand these transportation choices and build better models for our urban simulations.

4.5 Real-Time Arrival Prediction Accuracy

Although measures of travel time reliability on freeways and arterials are receiving increased attention, transit travel time reliability often continues to be viewed by transit agencies solely on the basis of overall on-time performance. We are currently working to increase our knowledge about the causes of travel time variability in transit. The objective of this research is to compare the on-time performance of routes based on specific characteristics of the service. Using historical Automatic Vehicle Location (AVL) data from King County Metro, we have begun an investigation of on-time performance and headway adherence on a segment by segment basis for routes throughout the transit system. We will develop a database of AVL data and link the data to characteristics of the segments, including:

1. Type of right-of-way exclusive right-of-way, exclusive lane, shared
2. Presence of transit signal priority
3. Underlying traffic volume
4. Route characteristics such as through-routing and stop spacing
5. Typical load factors on the route
6. Make-up of fare payment, such as percentage of monthly pass users

We will then analyze the effect of each of these factors on both the on-time performance and the deviation of transit travel times on that segment to determine which of these characteristics has the greatest impact on transit reliability. Ultimately, this information can then be used to improve the accuracy of the real-time arrival information from OneBusAway, as well as being useful to transit agencies in addressing bottlenecks in their systems.

5 Conclusion

Tools for enhancing the usability of public transit will continue to be an important application area for mobile application development. Our OneBusAway project is already providing a number of innovative transit tools, including providing access to real-time transit information for Seattle area bus riders through a variety of interfaces, including web, phone, SMS, and mobile devices. We have described the current suite of applications that constitute OneBusAway (including real-time arrival information, trip planning, and service alerts), and outlined a number of application research areas that we are actively pursuing to push the OneBusAway concept even further, using increasingly-powerful smart mobile devices to provide location and context-aware tools for navigating transit systems.

Acknowledgments. Many thanks to Nokia Research and to the National Science Foundation under Grant No. IIS-0705898 for their support of this research. Thanks also to Harlan Hile for his work on the Nokia native OneBusAway application, and especially to all of the users of OneBusAway for providing valuable feedback and suggestions.

References

1. Barbeau, S., Winters, P., Perez, R., Labrador, M., Georggi, N.: Travel Assistant Device, US Patent App. 11/464,079 (August 11, 2006)
2. Borning, A., Waddell, P., Förster, R.: UrbanSim: Using simulation to inform public deliberation and decision-making. In: Chen, H., et al. (eds.) Digital Government: E-Government Research, Case Studies, and Implementation, pp. 439–464. Springer, Heidelberg (2008)
3. Dziekan, K., Kottenhoff, K.: Dynamic at-stop real-time information displays for public transport: effects on customers. Transportation Research Part A 41(6), 489–501 (2007)
4. Friedman, B., Kahn Jr., P.H., Borning, A.: Value Sensitive Design and information systems: Three case studies. In: Human-Computer Interaction and Management Information Systems: Foundations. M.E. Sharpe, Armonk, NY (2006)
5. Google transit (June 2009), `http://www.google.com/intl/en/landing/transit/#mdy`
6. Google transit feed specification (June 2009), `http://code.google.com/transit/spec/transit_feed_specification.html`
7. Google transit partner program (June 2009), `http://maps.google.com/help/maps/transit/partners/faq.html`
8. Maclean, S., Dailey, D.: Wireless Internet access to real-time transit information. Transportation Research Record: Journal of the Transportation Research Board 1791(1), 92–98 (2002)
9. Patterson, D., Liao, L., Gajos, K., Collier, M., Livic, N., Olson, K., Wang, S., Fox, D., Kautz, H.: Opportunity Knocks: a system to provide cognitive assistance with transportation services. In: Davies, N., Mynatt, E.D., Siio, I. (eds.) UbiComp 2004. LNCS, vol. 3205, pp. 433–450. Springer, Heidelberg (2004)
10. Trimet 'unofficial' web and mobile applications (June 2009), `http://trimet.org/apps/index.htm`
11. Waddell, P., Wang, L., Liu, X.: UrbanSim: An evolving planning support system for evolving communities. In: Brail, R. (ed.) Planning Support Systems, Lincoln Institute for Land Policy (2008)