Location and Activity Modelling in Intelligent Environments

Robert Whitaker, Judy Kay

School of Information Technologies,
University of Sydney
{rwhitake,judy}@it.usyd.edu.au

Abstract. This paper describes ULAP, a framework for scrutable modeling and prediction of people's locations and activities, based upon a diverse collection of sensors, with varying reliability. It supports transformation and aggregation of sensor data, using this to build individual user models of location and activity. We propose an approach to indicate the certainty of predictions about users based upon unobtrusive data for location: it can be provided to applications and also serves as a form of explanation to users. We use this to report experiments involving 32 users, each with varying amounts of historic sensor data for machine activity, formal schedule and Bluetooth device detections. This is combined with group membership.

1 Introduction

Intelligent environments with ubiquitous computing need to exploit the large amounts of data from many, diverse sensors to build user models so that these can serve personalized applications. There are several approaches to modeling user location, for example Active Badge [1], BlueStar [2] and Lancaster Guide [3]. There has also been some recent work in machine learning to predict a user's future location, such as the Assisted Cognition project [4]. Corresponding work on modeling user's activity has had less attention, although there was early work by Orwant [5] and more recent work by Koile et al. [6]. We would like to go beyond these, combining sensor information about location and activity to model and predict both at the time of a request and into the future.

We explain our motivation in terms of the *Boris's Smart Office Door* Scenario; it was introduced in [7]. Boris is an academic, who always carries a Bluetooth enabled PDA. Natasha, a student, comes to his office to meet him. Unfortunately, he is not there. However, his smart door provides an interface which enables Natasha to request help in meeting him. The interface responds, according to Boris's context. Example responses include: Boris is nearby and interruptible so *Boris's Smart Office Door* sends him a message and he comes back to his office to talk with her; Boris is at a seminar and not interruptible but normally returns to his office after seminars so *Boris's Smart Office Door* tells Natasha he is likely to be here in 20 minutes (after the seminar); Boris is at home so *Boris's Smart Office Door* tells Natasha he is unavailable today.

We have determined the following requirements for a framework to support applications like *Boris's Smart Office Door*. It should: support modeling and prediction of location and activity over time, with flexibility in the time granularity of modeling; support multiple applications; make use of multiple, heterogeneous sensors; be easy to manage new, lost or altered sensors; support scrutability, meaning that it can explain its reasoning; protect the user's privacy through a permission system; make use of data for individuals and groups.

Section 2 gives an overview of ULAP and Section 3 describes our approach to representing certainty. We use this in the Section 4 report of evaluation. Section 5 has related work Section 6 has conclusions and future work.

2 ULAP Framework

The ULAP (User Location and Prediction) framework is shown in Figure 1. Its design has been influenced by the architecture of systems like Doppelganger [5], Web Guide [8], and MyPlace [7]. ULAP has three core components: the environment; the core of ULAP; and the applications which use it.

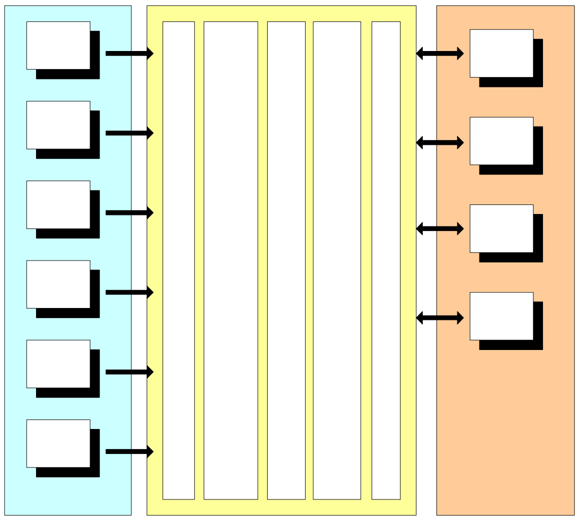


Fig. 1. ULAP Framework

The environment, shown at the left of Figure 1, can include arbitrary numbers of heterogeneous devices/sensors. In our implementation, there were six different types of sensors. These sensors and their purpose are summarized in Table 1. To ensure decoupling of the sensors from the ULAP core, we use publish/subscribe messaging to transmit data from the sensor to the central framework, as was done in MyPlace [7], although that work integrated just two sensors types. The ULAP approach enables sensors to collect data which is forwarded to all applications with subscriptions at the server. The sensor software does not need to know about those applications, decoupling the sensors, and hence, the environment, from the core framework.

Table 1. Summary of the different types of sensors

Sensor Type	Description/Purpose
BSpy	A Bluetooth based indoor positioning system. Determines location by querying all the Bluetooth enabled devices in range of its sensors.
BlueStar	Uses a combination of indoor and outdoor positioning systems to determine a person's location. The indoor positioning systems used Bluetooth technology.
Windows Activity	Focused on collecting information on the processes and machine a user was using at regular time intervals. Determined if a user was active at the machine or not through analyzing the times between keyboard and mouse events.
Login Sensor	Aimed at tracking a user's machine sessions on a network. It records a user's session information as well as the machine they are logged onto. This information can then be used to determine the location of the user.
Finger Sensor	This sensor collected location and activity information through the use of the *who* and *finger* commands. Location was determined based on the machine name, and activity by the value of the idle field from the finger command.
PDA	Enables a user to log activities and whether interruptible or not.

We now describe the elements of the ULAP core. Leftmost in Figure 1 is the data converter/filter. This must deal with two tasks: aggregation of data from multiple sensors and the conversion of data to a form suitable for the user models.

First consider aggregation. Each sensor can record different types of data and can represent the same data in different ways. For example, the BSpy sensor represents a location using the MAC address of the sensor (00:01:0E0:41:E0:10), while the login sensor represents the location as the machine name (pg-g62-1). In such cases, data from the two sensors cannot be merged directly to give the correct symbolic location[1]. ULAP must map from the raw values from each sensor to consistent symbolic values.

The importance of this issue may not immediately be obvious: much of this functionality could be handled inside the user modeling component or by an application using the user model. However, this is impossible where sensors have different ways to identify users. The data converter/filter component must ensure data is added to the correct user model. It maps the user ID for each sensor to the internal representation used by the ULAP framework.

A similar problem relates to handling multiple devices for the same user. For example, the BSpy sensor identifies users by the MAC address of their device. Where a user carries two devices, a phone and a PDA, both must map to the same symbolic value.

The implementation of this process is based on an approach similar to that of XSLT transformations of XML documents. It builds an internal representation of the XML formatted conversion file. Using this representation it attempts to find an appropriate mapping and apply the conversion. If no mapping is found the original raw value is used.

As data is collected, it must be stored and modeled. This component of the ULAP core uses PersonisLite, a light weight version of Personis [9]. The user model has two contexts, one for the modeled components of location and the other for components of the user's activities. This part of the framework supports group modelling, by

[1] Symbolic location refers to the human representation of a location eg. the name of a room

dynamically generating required group models at runtime, based upon the individual models for each member of the group.

The next part of the ULAP core is the resolvers: these are responsible for interpreting sensor evidence within the user models. Resolvers are selected, at runtime by the application. Different resolvers provide variable granularity of location and activity prediction, as needed for the different subcases of the scenario.

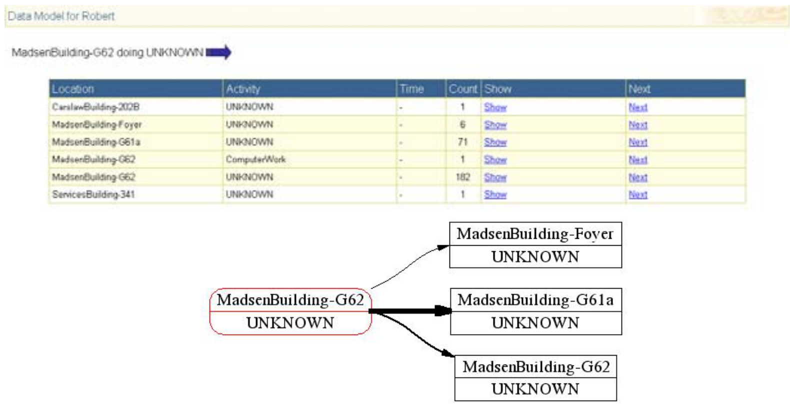


Fig. 2. Example of the ULAP generic interface supporting user scrutiny of results.

ULAP predictions of the user's future location and activity are based upon Markov Chain models, a choice based on its simplicity and the potential for intuitive explanations of the system operation. This means that ULAP can enable users to scrutinize the user modeling processes. Each location/activity pair is represented as a node and possible path in the chain.

The rightmost part of the ULAP core shown in Figure 1 is the interface support enabling the user to see the user model. The Markov model gives a natural visualization of the system's reasoning on the person's movement between locations and activities. An example of a model visualization is shown at the bottom of Figure 2. ULAP supports variable length models. The figure also shows the interface that enables a user to dynamically iterate through the models to see how predictions were determined and to explore additional predictions into the future.

The last main part of the ULAP architecture is the applications, such as *Boris's Smart Office Door*. Shown at the right of Figure 1, three applications we have built to evaluate ULAP are: first, ULAP Modeler, for individual users; second, Group Modeler; and third, Last Location/Last Activity, which query the user's current or last known location and activity (as a basis for prediction into the future as in the scenario where Boris was at a seminar).

The framework has been implementated in a combination of Perl and Python scripts which interact with and manipulate the data stored in the user models. Through the use of system hooks it was possible to monitor mouse and keyboard events to ensure accuracy in the assumptions made by the activity sensors for the activity sensors. The scrutable interface is a Perl based web interface which uses *dot* [10].

3 Modeling Certainty and accuracy

Ideally, we would have had a set of gold standard training and test data: then we could have used various resolvers to query the user models and then compare the results with the known correct result. Indeed, we built tools to collect such data, based upon users maintaining a log of their actual location/activity. Various paper schemes as well as a PDA application were tried. It is unsurprising that people found it too difficult to remember to keep the record (or too irritating to be reminded).

Accordingly, we decided that a different approach was needed. Our approach was partly motivated by our goal of scrutability: we wanted to be able to inform both users and applications of the certainty of a prediction. We identified two elements of this:

- The *consistency* of the available evidence;
- The *nature* of the evidence available.

To determine a consistency value for a prediction, ULAP calculates $\{w_i\}$, a set of weights, where each w_i is the weight of the evidence for the i-th location/activity supported by any of the evidence. ULAP then determines $max\{w_i\}$, meaning that i is the value with the highest weight. This value is the result of the query. Its consistency is calculated as $max\{w_i\}/sum\{w_i\}$. If there is no evidence for a query, we return a consistency value is 0. With one piece of evidence, it is 1.0.

This can be calculated at the time of the user model query. Then, ULAP applies the appropriate location/activity granularity. So, for example, if an application asks if the user is interruptible or not, there are two values and each piece of evidence is interpreted to contribute to the weight of support for one. If, on the other hand, a query specifies a resolver with several location/activity values, ULAP calculates the total evidence weight for each of these. There are many ways to calculate the weights. A review of a range of such algorithms has been described for ubiquitous computing [11]; any of these could be applied within ULAP. Notably, since we want to deal with multiple sensors of varying reliability, an algorithm can exploit knowledge to adjust the weight according to sensor reliability.

To illustrate the process, suppose 180 pieces of evidence support location A and 20 support location B. An algorithm that treats all evidence equally returns the value A, with consistency 90%. Taking another example, if there are 10 equal-weight pieces of evidence for each of 20 different location/activity values, each is equally likely. The resolver returns one of them, with accuracy 5%.

Clearly, there are serious limitations to this consistency measure. The second element of certainty relates to the *nature of the evidence* and has to help deal with this. For example, consider the case in the paragraph above for locations A and B. One very simple indication is the total number of pieces of evidence. This measure is what we have used.

In summary, in lieu of an accuracy measure we use consistency and the amount of evidence. This is clearly inferior to a measure of true accuracy, calculated by comparing a ULAP prediction against a known correct result. However, in our experiments, that was unavailable. Moreover, in general, it will be important for user modeling predictions for ubiquitous applications to include a prediction of the accuracy of the result [12]. So, it is important to define a practical way to indicate the certainty of a prediction, as our approach does.

4 Evaluation

Our evaluation tested the effectiveness of the ULAP framework by implementing it and then using it to build a range of models. We now report its use in:

- modeling individual users, based upon a variety of sensors for location and activity, with historic data used to support predictions and comparing the effect on certainty from the evidence of additional sensors;

- modeling groups by aggregating individual models, comparing the effect on certainty of predictions, where this had the potential to provide predictions for individuals even when no sensor data was available for them but there was data for people in the same group.

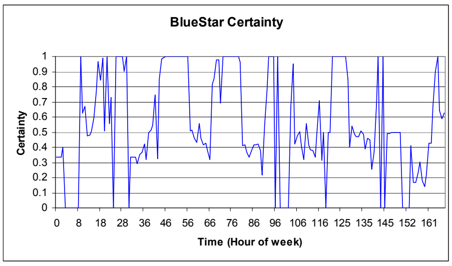
As already discussed, individual model certainty is based upon consistency and the amount of evidence for predictions. This section summarizes results for multiple heterogeneous sensors, individual and group modeling. For fuller results as well as scalability experiments, see [13].

Our experiments have been based upon data for 32 users. A summary of the data for four of the more interesting users is summarized in Table 2. Data was collected over 4 months for the BlueStar (Bluetooth) sensor types, and 6 weeks for the other sensor types. This is of a similar order to much of the published work, such as the Assisted Cognition project [4, 14-17] which had 6 months of a single data type, GPS, to model an individual's movements around a large city. We used this to build individual models.

Table 2. Details sensor readings or detections for 7 users with relatively rich collections.

User	Number of detections recorded						
	BSpy	BlueStar	Login	Activity	Finger	PDA	Timetable
A1	4,819	-	285	5,717	5,111	250	YES
B1	6,464	-	0	-	10,939	-	-
E1	-	-	159	5987	131	-	YES
H1	-	163,392	0	-	0	-	-

Figure 3 shows the contrasting levels of consistency in two extreme cases. The graph on the left is for User H1 and is built from 163,392 pieces of BlueStar data collected over four months, covering every hour of each day of the week. Consistency values less than 1.0 are due to detection of H1 by multiple sensors at different locations. This graph on the right is for User A1, based upon 16,182 pieces of sensor evidence, representing data collected over each hour of the week. The zero points occurred when there was no data for the user. A comparison of these graphs shows that both return similar consistency readings, even though in the right hand graph we have increased the number and type of sensors used, as well as increasing the number of possible of combinations a user can be detected in a single hour through the observation of activity in addition to location.



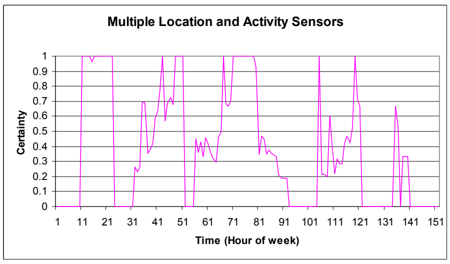
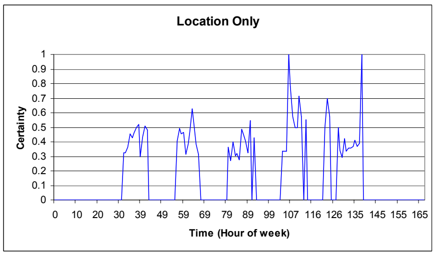


Fig. 3. The graph on the left shows the consistency of models based on 4 months of BlueStar data. Compare this with the graph on the right, which shows consistency of 6 weeks of data collected from multiple heterogeneous sensors.

Figure 4 indicates the relative effect of *activity* sensors in addition to multiple location sensors. The left graph, for user A1's location alone tends to have consistency around 50% for each of the 5 days of the week and no other data. The right graph is for the same user with activity sensors as well. This visually gives a higher consistency. There are many reasons for these differences: the types of sensors, activity sensors usually have a finer location granularity; and the use of additional data captured by these sensors when determining certainty.



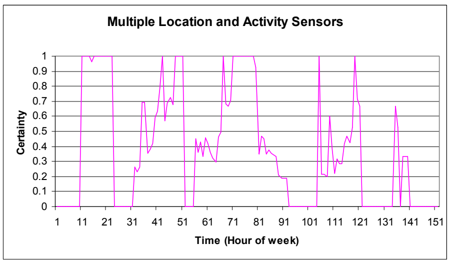


Fig. 4. The graph on the left shows consistency with multiple location sensors. That on the right also has activity sensor data.

In this evaluation, models were constructed for a range of groups of people. Using this calculation time periods where the user mainly performs one event will clearly stand out through a certainty close to one. This can then be compared to those times when many different events have been observed over the user, in this case the certainty will be lower dependent on the number of different events seen and how often each event was observed. We now look at two of those profiles in detail with those being: the profile of a university academic; and that of honors students teaching various courses.

Figure 5 shows the consistency graph for User B1, a university academic. As shown in Table 2, their model is based on substantial data sets from two sources, BSpy and Finger. This person also tends to keep a fairly consistent schedule over the four months: for example the very consistent period around hour 70 of the week is their research group weekly seminar and other meetings. When shown this graph, B1 could identify their various regular activities in the week.

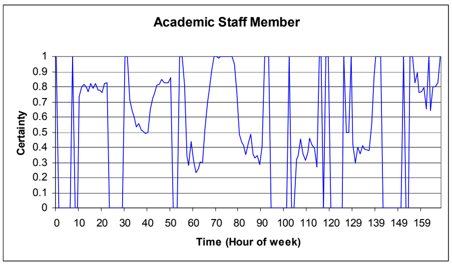


Fig. 5. Prediction consistency for B1

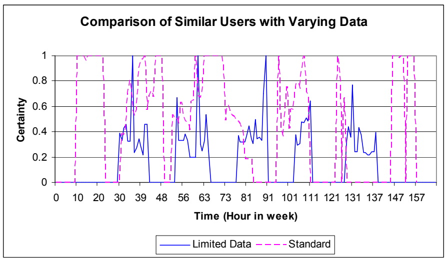


Fig. 6. Consistency for A1 compared with E1.

Figure 6 shows consistency measures for two Honours students, A1 in broken lines and E1 in solid lines. Both have similar schedules but, as can be seen from Table 2, A1 had six sources of sensor evidence where E1 had just four. Notably, A1 had Bspy data but E1 did not. E1, with limited data has consistency values around 0.4 and the five days of the week can be seen clearly. These trends are quite strong, taking account of the 6 week period that provides them. For A1, there are many more periods where predictions have higher consistency, including periods on weekends and nights.

The similarity of the two users of Figure 6 suggests the potential value of exploiting group membership or user similarity to support predictions even for users for whom we have no data. We performed group modeling experiments; these are similar to communities described in Doppelganger [5] although this work does not report results of user experiments as we do below. The group modeling functionality allows a person to be associated with every relevant group. So, for example, an Honours student who tutors and has a desk in Lab 1 can be assigned to multiple groups: Honours which includes people in many labs, tutors which overlaps the Honours group and includes others, Lab 1 group which includes students and research staff in that lab. Table 3 shows the groups identified for experiments.

Table 3. Number of detections per group from each sensor.

Type	Group	Number of detections recorded						
		BSpy	BlueStar	Login	Activity	Finger	PDA	Timetable
Hons	Honours	4819	0	20223	17873	11428	250	2
	Hons Group 1	4819	0	1633	16341	5788	250	2
Tutors	Tutors	10903	0	10009	16341	10564	250	2
	Tutors SOFT2001	4819	0	586	11704	5413	250	2

As seen in the left hand graph of Figure 7 a substantial confidence improvement was obtained for most time periods, as the number of conflicts or possible locations for each time period had been reduced. However, through the modeling of tutors for one particular course no substantial certainty improvement could be gained, nor any conclusive prediction be made about this group because of the group diversity. The certainty results can be seen in the right hand graph of Figure 7.

To identify useful groupings, we created group models which systematically explored each grouping. We then used the consistency measure as a basis for selecting useful groupings. This identified groupings that were unhelpful, such as that of tutors, where different people are allocated to different classes, meaning that data for one person is generally not consistent with data for others in the group.

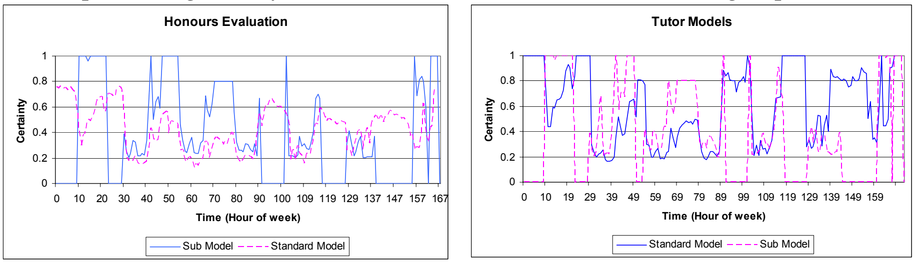


Fig. 7. In the Left graph shows use of subgrouping consistency. In the right graph, this approach was not successful because of subgroup diversity.

4 Related Work

Two projects were particularly important for the design of ULAP. Doppelganger [5] also aimed for a general framework for gathering and processing heterogeneous sensor data and community modeling, but had a different architecture and did not report results of experiments for long term user data. The more recent Assisted Cognition Project [4] models movement paths to assist the mentally disabled. One of its prototype systems, the Activity Compass [14], uses PDA and GPS location sensors. A second project is an application, called Opportunity Knocks [16], designed to run on a mobile phone, models a person's path in a city based on GPS data. There has been some work in using activity sensors, such as Activity Zones [6] and considerable work on location sensing, such as Active Badge [1], Lancaster Guide [3], Web Guide Project [8] and Multiple User Detection [18]. ULAP has explored a different dimension of the problem of modeling user location and activity, with a focus on far more heterogeneity of sensors than is the case in these projects. Several others have also explored the use of Markov models, for example, Assisted Cognition [4], Multiple User Detection [18] and Doppelganger. And there has been work on other learning approaches, for example Web Guide Project [8], Assisted Cognition [4] as well as Doppelganger. Importantly, at this stage in the area of location and activity modeling much of the evaluation has been based upon synthetic data or special test data. Other work that has collected authentic sensor data for normal or near normal users has been done in projects like MyPlace [7], Doppelganger [5], Activity Zones [6], Assisted Cognition [4] and Multiple User Detection [18]. The scale, diversity and time period of our sensor data is broader than these projects.

5 Conclusion

This report has described a framework for modeling location and activity based ondata collected from ubiquitous environments. We demonstrated the effectiveness of this framework through its implementation and analysis of the models generated by it. We have reported consistency results demonstrating ULAP's ability to refine its model by using multiple heterogeneous sensors and the modeling of groups.

This work provided an initial investigation into the modeling and prediction of location and activity information for an individual and group. The implementation and evaluation of a framework is the first step to the development and support of personalized applications for the user and their environments.

6 References

1. Want, R., et al., *The Active Badge Location System*. ACM Transactions on Information Systems, 1992(January 1992).
2. Quigley, A., et al. *BlueStar, a privacy centric location aware system*. in *IEEE Position, Location and Navigation Symposium 2004 (PLANS 2004)*. 2004. Monterey CA, USA.
3. Distributed Multimedia Research Group, *The Guide Project*. 2004.
4. Kautz, H., et al. *An Overview of the Assisted Cognition Project*. in *AAAI-2002 Workshop on Automation as Caregiver: The Role of Intelligent Technology in Elder Care*. 2002.
5. Orwant, J. *Heterogeneous Learning in the Doppelganger User Modeling System*. in *User Modeling and User-Adapted Interaction*. 1995.
6. Koile, K., et al. *Activity Zones for Context-Aware Computing*. in *Ubicomp*. 2003.
7. Kay, J., B. Kummerfeld, and D. Carmichael, *Consistent Modelling of users, devices and environments in a ubiquitous computing environment*. 2004, University of Sydney.
8. Fink, J. and A. Kobsa, *User Modeling for Personalized City Tours.* Artifical Intelligence Review, 2002. **18**(2002): p. 33-74.
9. Kay, J., B. Kummerfeld, and P. Lauder. *Personis: A Server for User Models*. in *Adaptive Hypertext 2002*. 2002: Springer.
10. Ganser, E., E. Koutsofios, and S. North, *Drawing Graphs with dot*. 2002.
11. Hightower, J. *From Position to Place*. in *2003 Workshop on Location-Aware Computing*. 2003. Seattle, Washington.
12. Antifakos, S., A. Schwaninger, and B. Schiele. *Evaluating the Effects of Displaying Uncertainty in Context-Aware Applications*. in *Ubicomp 2004*. 2004. Nottingham, England.
13. Whitaker, R., *Location and Activity Modelling in Intelligent Environments*, in *School of Information Technologies*. 2004, University of Sydney: Sydney, NSW. p. 131.
14. Patterson, D.J., O. Etzioni, and H. Kautz. *The Activity Compass*. in *First International Workshop on Ubiquitous Computing for Cognitive Aids*. 2002. Gothenberg, Sweden.
15. Patterson, D.J., et al. *Inferring High-Level Behavior from Low-Level Sensors*. in *Fifth International Conference on Ubiquitous Computing*. 2003.
16. Patterson, D.J., et al. *Opportunity Knocks: a System to Provide Cognitive Assistance with Transportation Services*. in *Ubicomp*. 2004. Nottingham, England.
17. Kautz, H., et al., *Foundations of Assisted Cognition Systems*. 2003, UW CSE Technical Report.
18. Ashbrook, D. and T. Starner, *Using GPS to learn significant locations and predict movement across multiple users.* Personal and Ubiquitous Computing, 2003. **7**: p. 275-286.