Platys: From Position to Place-Oriented Mobile Computing

Laura Zavala, Pradeep K. Murukannaiah, Nithyananthan Poosamani, Tim Finin, Anupam Joshi, Injong Rhee, Munindar P. Singh

■ *The Platys project focuses on developing a high-level, semantic notion of location called place. A place, unlike a geospatial position, derives its meaning from a user's actions and interactions in addition to the physical location where it occurs. Our aim is to enable the construction of a large variety of applications that take advantage of place to render relevant content and functionality and, thus, improve user experience. We consider elements of context that are particularly related to mobile computing. The main problems we have addressed to realize our place-oriented mobile computing vision are representing places, recognizing places, and engineering place-aware applications. We describe the approaches we have developed for addressing these problems and related subproblems. A key element of our work is the use of collaborative information sharing where users' devices share and integrate knowledge about places. Our place ontology facilitates such collaboration. Declarative privacy policies allow users to specify contextual features under which they prefer to share or not share their information.*

Mobile applications that automatically adapt to their surrounding circumstances will lead to an enhanced user experience. Emerging mobile applications exploit a user's location to deliver personalized services. In current practice, the user's location is captured at the level of position, that is, geospatial (latitude-longitude) coordinates. However, what often matters for experience is the user's place — a location in conceptual terms, such as home, work, gym, or grocery shopping, that combines positions with the user's activities, properties of the user's environment, and the activities of people surrounding or interacting with the user.

The Platys project seeks to realize the above notion of place and enable the construction of a rich variety of applications that take advantage of place to render relevant content and functionality and, thus, improve user experience. Examples include proactively (1) changing phone settings (for example, turn ringer off during a meeting and turn it back on at the end of the meeting); (2) downloading relevant information (for example, the map of an amusement park, museum, or any place the user visits); (3) annotating images or other media; (4) filtering content such as alerts, notifications, and customized ads; (5) changing the ambiance (for example, playing music); (6) showing (place-dependent) reminders

 ISSN 0738-4602

from to-do lists; and (7) pushing recommendations to the user when the situation seems appropriate.

In this article, we report on our efforts pertaining to the Platys project. A semantic model of user-centered places, the Platys ontology enables the mapping of positions to places. In the model, places and activities can be represented at different levels of granularity using subsumption hierarchies. We want to determine a user's place at any given time. Place recognition has been addressed with standard machine-learning classifiers as well as a semisupervised expectation-maximization algorithm. The recognition is based on data captured from a user's smartphone: location, sensor readings, Wi-Fi, Bluetooth scannings, and phone settings. Location is an essential part of place and therefore place recognition relies on location sensing. Since frequent location sensing by a mobile device depletes power, we have also investigated energy-efficient techniques for maintaining a sufficiently accurate location model.

We study not only private places specific to each user, but also public places that are shared by a community or an affinity group. A key element of our work is the use of collaborative information sharing where users' devices share and integrate knowledge about places. By providing a common semantic model, the Platys ontology facilitates such collaboration. Declarative privacy policies using the ontology allow users to specify contextual features under which they prefer to share or not share their information. Co-occurrences of users at particular places are used to learn the social circles of users.

Place-aware proactive mobile applications will be capable of proactively performing actions or making recommendations according to the user's current place. Available frameworks (for example, Locale for Android[1] and Nokia Situations[2]) allow development of the former. A situation and the action to be taken in it must be specified with fixed rule patterns such as: WHEN [in meeting] SET [ringtone=off]. Situations must be clearly defined through specific values of phone status attributes such as date, time, location, and battery level. A place such as "work" or "in meeting" could be specified as a situation by using a combination of date, time, and location.

This approach is clearly limited and rigid. Our approach recognizes place at different levels of granularity and capturing nuances in how a user perceives them. The user need not specify fixed attribute values that define the *in meeting* place. Consequently, if there are changes in those values (for example, a change of the normal meeting room), our approach may still be able to recognize the place.

While decades of research in context-awareness has addressed similar issues and made progress solving particular problems (see sidebar), nontrivial context-aware applications are still unavailable to everyday users as are frameworks that facilitate their creation. This is especially true for frameworks supporting a general, complex, and all encompassing notion of context.

The remainder of this article is organized as follows. In the next section we provide a formal definition of a user-centered place and consider elements of context particularly relevant to mobile computing. We then discuss the different approaches we have used to address the problems identified in realizing our place-oriented vision. Our techniques are user-centered and attempt to recognize places in a privacy-preserving manner. In other papers (Murukannaiah and Singh 2015; Zavala et al. 2011) we discuss architectures on which place-aware applications can be engineered. Currently, prototypes and experiments have been run in several university campus scenarios.

Context-Aware Computing

Research in context-aware computing (Schilit, Adams, and Want 1994) aims to enable computing systems that acquire and maintain context data and use it to adapt their behavior. It originated with Weiser's vision of ubiquitous computing (Weiser 1999) where human activities are enhanced with devices that are all around but unnoticeable to the user and that provide services that adapt to the circumstances in which they are used. Papers by Want and colleagues and Schilit and colleagues (Want et al. 1992; Schilit, Adams, and Want 1994; Schilit et al. 1993) are early works in context-aware computing and dealt with tracking a user's location and using it to provide better services or sharing it with others. Research in the field has addressed a range of problems, including the formal definition and categorizations for context, context representation, context recognition (user location, user activity, user mood, and so on), and context sharing. Several software frameworks have been proposed to facilitate the development of context-aware applications (Korpipaa et al. 2003; Gu, Pung, and Zhang 2004; Fahy and Clarke 2004; Salber, Dey, and Abowd 1999; Román et al. 2002; Dey, Abowd, and Salber 2001; Chen, Finin, and Joshi 2005). At a minimum, they all comprise context-recognition services (usually distributed) and a context manager (usually centralized) that allows client applications to query and register for context information. Some also include formal context modeling to share contextual information among heterogeneous entities, security and privacy, inference mechanisms, and agent capabilities. Chen and Kotz (2000) and Baldauf, Dustdar, and Rosenberg (2007) provide surveys of developments and applications in the field.

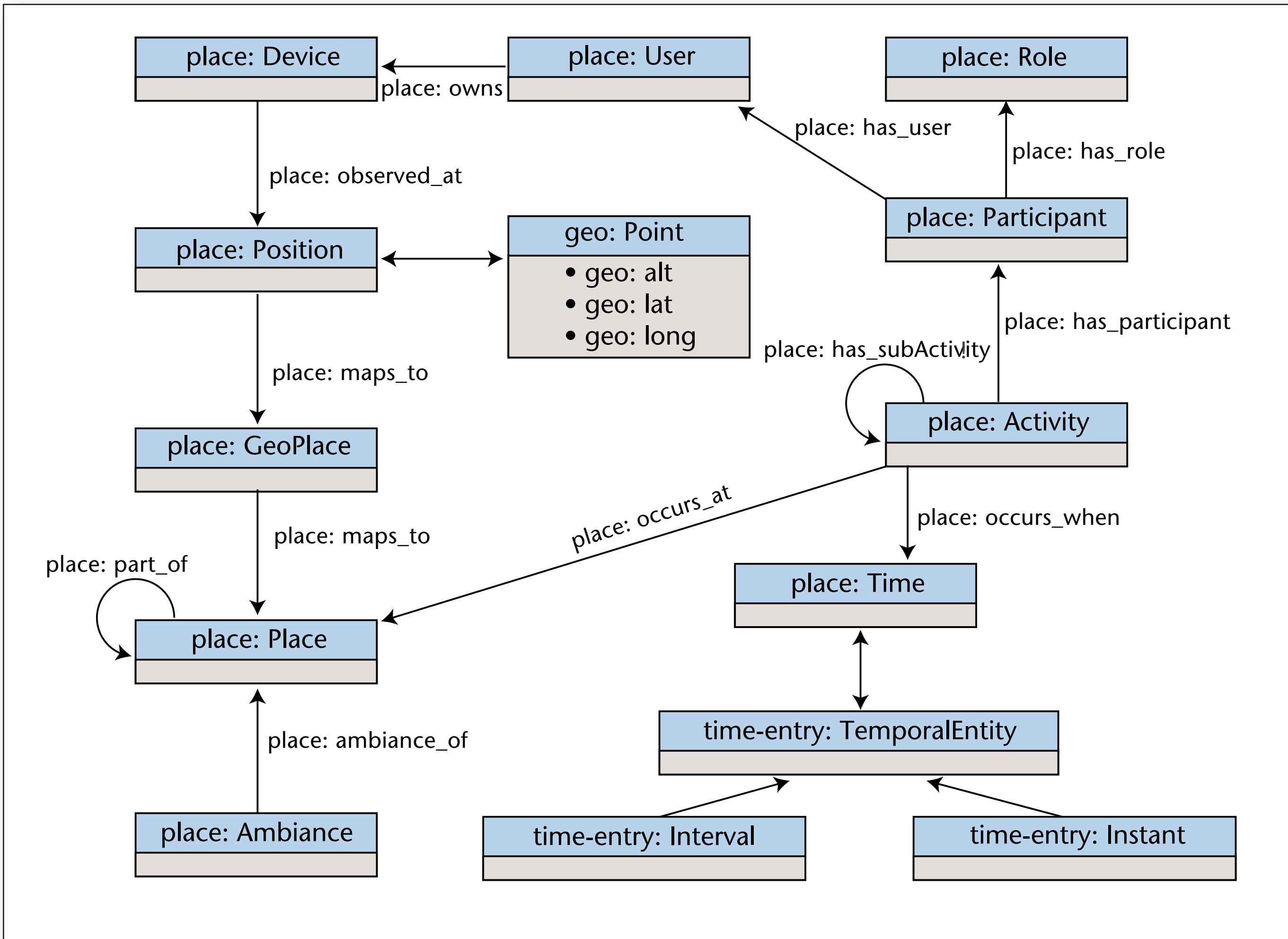

Figure 1. The Place Ontology Models the Concept of Place in Terms of Activities That Occur There.

Semantic Place Model

We define a (user-centered) place as a conceptually well-delineated set of positions associated with a user and alternatively combined with contextual information such as user activities, environmental properties, and nearby people and their ongoing activities. Using this user-centered, contextual notion of place it is possible to (1) capture nuances in how a user perceives places; (2) have a place that includes disjoint spatial regions (the set of positions that delineate a place need not be contiguous). For example, each workplace of a user has its own spatial region, but a user (for a specific purpose) may conceptualize all workplaces as a single place; (3) map a spatial region to more than one place, each associated with a different user. For example, a coffee place can be café for some users, but workplace for others; and (4) map a spatial region to more than one place for the same user, varying contextual information. For example, a shopping mall can be mall as well as workplace for a user who works at the mall. The contextual information would be used to know when the user is at one or the other.

Place Ontology

We developed a light-weight, upper-level ontology to model the concept of place in terms of activities that occur at that place. We adopt description logics (Baader et al. 2003), specifically the web ontology language OWL (Bechhofer et al. 2007), and associated inference mechanisms to represent the model. OWL supports the specification and use of ontologies that consist of terms representing individuals, classes of individuals, properties, and axioms that assert constraints over them.

Figure 1 shows the ontology's core classes and their relationships. A *User* is associated with a *Device* whose *Position* maps to a geographic place (GeoPlace) such as "UMBC" and to a conceptual place (Place) such as "at work." Some *Geoplaces* are part of

```
@prefix rdf: <http://www.w3.org/1999/02/22--rdf--syntax--ns#>.
@prefix place: < http://ebiquity.umbc.edu/ontologies/platys#>.
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>.
@prefix kb: <http://example.org/kbdevice/>.
@prefix gn: < http://www.geonames.org/ontology#>.
```

```
kb:droid1 place:observed_at kb:anon01f
kb:anon01f rdf:type place:Position
kb:anon01f geo:lat 39.253525
kb:anon01f geo:long -76.710706
```

```
[partof:
(?a gn:parentFeature ?b)
->
(?a platys:place_part_of ?b)
(?b platys:spatially_contains ?a)
]
```

Figure 2. KB Assertions in Turtle and a Jena Rule Used to Integrate Knowledge from GeoNames.

KB assertions (left); Jena Rule (right).

others through spatial containment defined by the transitive (part_of) relationship. The mapping from Positions to GeoPlaces is many to one and the mapping from Positions to Places is many to many, that is, the same Position may map to multiple Places, even for the same User; and, many Positions map to the same Place. Mapping from Positions to Places is done through GeoPlaces (maps_to is a transitive property). An *Activity* involves Users under certain Roles, and occurs at a given Place and Time. Activities have a compositional nature, that is, fine-grained activities make up more general ones. Ambiance encapsulates concepts describing the environment of the User (for example, noise level, ambiance light, and temperature).

The representation of activities is crucial to mapping positions to places. This approach reflects our pragmatic philosophy that the significance or meaning of a place for a given user depends largely on the activities that occur there, specially the patterns of lower-level activities. The idea applies at both the individual and collaborative level. For a user individually, the patterns of actions can help identify a place from that user's perspective. The patterns of actions common to users can help identify a place in a collaborative manner. For example, a park or a library would see similar patterns from multiple users.

The Knowledge Base

The knowledge base (KB) on each device aligns with the Place ontology. Using this ontology, devices can share information about their context. Given the position of the device (that is, geospatial coordinates) and the user's activity (if available), we assert the corresponding facts in the KB. In this section we focus on how we populate the KB with geoplace information. Activity and place inference are covered in the next section.

We use the Android location API to obtain the position of the device. Position on Android phones is determined through location providers such as the device's GPS and the network (which is based on availability of cell tower and Wi-Fi access points). Given the Position of the user's device, we assert the corresponding triples into the KB (figure 2). Then, we use additional online resources, specifically GeoNames' spatial KB (RDF version) and its associated services, to infer the user's GeoPlace in five steps: (1) using reverse geocoding services to find the closest GeoNames entity to the current position; (2) querying GeoNames through SPARQL to get further information about that entity; (3) applying transformation rules to the data obtained from GeoNames (figure 2); (4) using OWL inference to obtain the triples corresponding to the spatial containment of entities (transitivity of the part_of relationship); and (5) using ad hoc property chains (figure 3) to infer

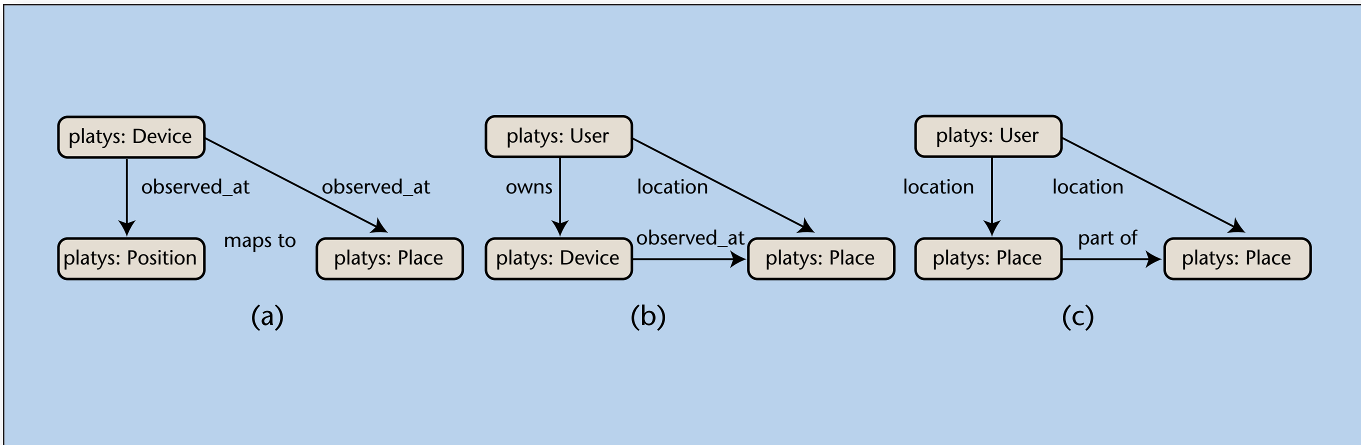

Figure 3. Property Chain Axioms Asserting Facts About a User's Location.

(a) Device is observed at the place whose position it maps to. (b) User's location is the place where his or her associated device is observed. (c) Generalization of user location based on spatial containment (part of).

knowledge about a user's geoplace based on the places his or her associated device is observed.

Recognizing User-Centered Places

We wish to determine a user's place at any given time using data captured from her smartphone: location, sensor readings, wifi, bluetooth scannings, and phone settings. We have addressed the problem using a semisupervised expectation maximization algorithm as well as standard machine-learning classifiers. In the former, we determine place based on unaligned historical sensor data and user labels. The focus is on place as a set of positions and we are able to recognize disjoint spatial regions as a single place. In the latter, contextual information is also taken into account. We recognize place and activity at different levels of granularity. Further, we are able to recognize the same spatial region as more than one place.

Semisupervised Expectation Maximization (EM)

We developed (Hang, Murukannaiah, and Singh 2013) a semisupervised expectation-maximization (EM) algorithm to recognize user-centered places. Our approach recognizes subjective places; does not require manual tuning of place radius and duration; and employs infrequent sensor readings from multiple sources. Each user is required to label places of his or her interest (at least once for each place). Given a user's place labels and historical sensor data from multiple sources, our algorithm operates as follows:

Build a data set consisting of a data instance for each sensor reading and user label. Further, consider a training set as a subset of the above data set, consisting of instances corresponding to place labels only.

Assign features to training instances. For each sensor type, add three features — a sensor reading at the time of labeling one immediately before and one immediately after.

Assign a place label to each unlabeled instance. For each unlabeled instance, find the similarity of the instance to each labeled instance and assign the label corresponding to the most similar instance.

Remove incorrect labels by establishing a similarity boundary for each place and iteratively shrinking it (using EM) until instances assigned to each place are sufficiently similar to each other.

We evaluated our approach in a study of six users. Each carried an Android phone installed with a data-collection program for at least three weeks. The program recorded the sensor data (including GPS, Wi-Fi, and Bluetooth) and prompted users to label places at random intervals. We compared our approach with two stay-point approaches (Hariharan and Toyama 2004; Zheng et al. 2012). Platys cannot be directly compared with a stay-point approach since the latter only recognizes whether a user is in some stay point or not (not the specific stay point as there are no labels). To enable a fair comparison, we implemented two versions of Platys: (1) Place-or-not, which only recognizes whether a user is in one of the labeled places or not, and (2) Which-place, which recognizes the specific place. Figure 4 shows the comparison. Our main findings are as follows. First, Platys (Place-or-not) performs better than both stay-point approaches used for comparison. Importantly, the F-measures for Platys, unlike those of stay-point approaches, are straight lines since they do not depend on place radius and duration. Second, stay-point approach with optimal place parameter values may perform better than Platys for some users. However, as shown, optimal place parameter values vary

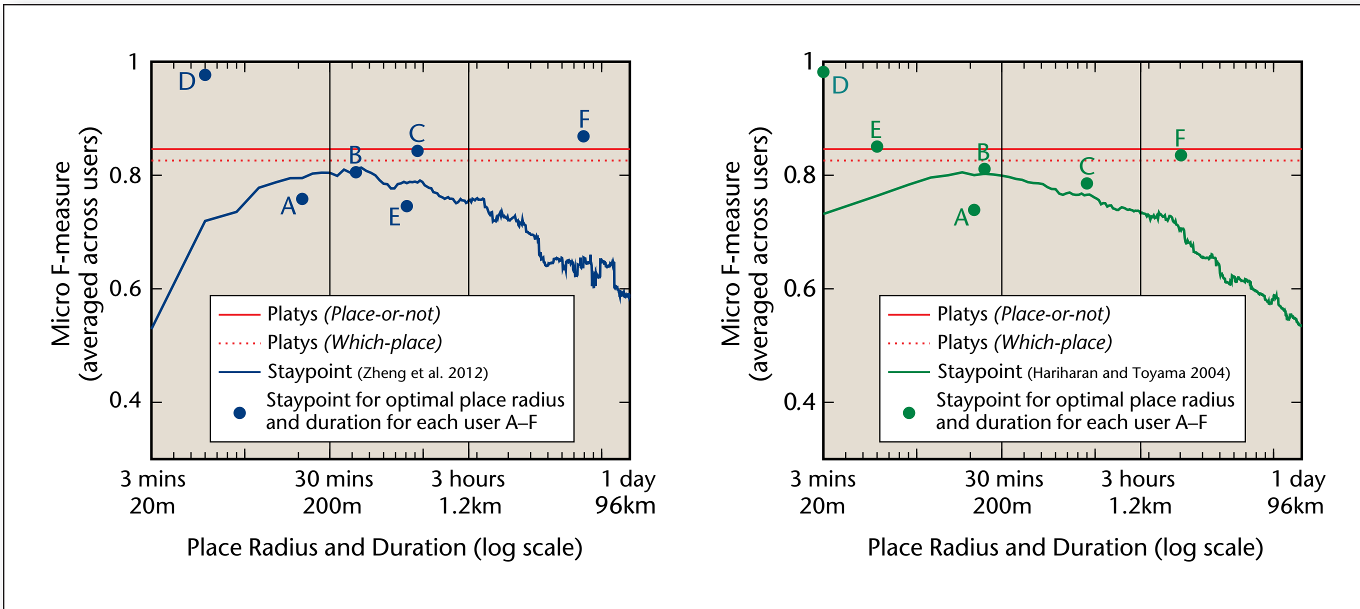

Figure 4. Comparing Platys with Two Stay-Point Approaches That Distinguish Place from Nonplace, but Do Not Identify a Place.

from user to user. Third, Place-or-not is an upper bound on Which-place. However, in most cases, the performance of Which-place is close to that of Place-or-not. Thus, once Platys identifies a user to be in one of the labeled places, in most cases it correctly identifies which place the user is in.

Supervised Classification

We used supervised machine-learning algorithms to recognize activity (for example, "sleeping," "walking," "sitting," "cooking") and place (for example, "at work," "at home") at different levels of granularity (Zavala et al. 2011). The current experiments are confined to a university domain and the users are students and faculty. Furthermore, the experiments are focused on learning to recognize an individual's context (activity and place). For high-level general activities, we obtained a high accuracy but with more fine-grained ones the accuracy drops. We expect this to improve as we incorporate more complex models that allow for collaborative context inference.

We evaluated our approach in a study of five users. Each user carried an Android smartphone installed with a data-collection program. The information collected includes location, ambiance light and noise, Wi-Fi scanning, Bluetooth scanning, current calendar event (if any), sensor readings (accelerometer, magnetic field, orientation, and proximity), call statistics (missed calls, answered calls, and duration), and phone state (idle, in use, and others). At the beginning of each collection, the user is asked to enter the current place and activity. This information is used as ground truth for the learning task. Multiple labels can be selected to capture different levels of granularity (for example, at work, in office, in meeting). Hierarchy is not specified in the collection program since we preprocess the data for each particular learning task we try and we know the hierarchy.

We compared the performance of different machine-learning algorithms in classifying the place and activity of the user given the particular readings from the phone after some preprocessing. We have conducted several experiments varying the classification task to different combinations of place and activity at different levels of granularity. We present here results for three algorithms: decision trees, naive Bayes, and support vector machines (SVMs). Table 1 shows the accuracy of the algorithms for a midlevel detailed activity recognition task for a particular user and nine everyday activities using 10 crossfold validation and 66 percent split validation testing options. Accuracy levels are comparable to those reported by Bao and Intille (2004), although their focus was mainly recognition of a limited subset of everyday activities consisting largely of ambulatory motions. Overall, recognition accuracy is highest for decision tree classifiers, which is also consistent with the paper by Bao and Intille (2004). This might be due to the fact that rule-based activity recognition appears to capture conjunctions in feature values. The naive Bayes approach assumptions of conditional independence between features and normal distribution of feature values may contribute to the weaker performance of the approach. Furthermore, to achieve good accuracy even when the assumptions are not met, the approach usually requires large volumes of training data.

Higher accuracy is observed for higher-level gen-

Classifier	10 Fold	66% Split
SVM (LibSVM)	76.9231%	79.5699%
Decision Tree (J48 Trees)	91.97%	93.3133%
Naive Bayes	47.9638%	50.5376%

Table 1. Accuracy of Different Algorithms for Activity Recognition of a Particular User and Ten Everyday Activities.

Activity	Accuracy
At Home, At Work/ School, Elsewhere	99.0%
In Meeting, In Class, Elsewhere	94.94%

Table 2. Recognition Accuracy for High-Level, General Activities Using Decision Trees.

Activities: Working/Studying, Sleeping, Walking, In Class, Outdoors, In Meeting Talk-Listening, Other/Idle, Shopping.

eral activities (table 2). Our 99 percent accuracy for "at home versus at work versus elsewhere" is higher than the one reported by Eagle and Pentland (2006) where they used a simple hidden Markov model conditioned on both the hour of day as well as weekday or weekend for the same classification task.

From Place to Social Circles

Often a place is associated with a social context. For example, a user interacts with his or her family *at home,* colleagues at a *workplace,* and friends at a *party.* Platys Social (Murukannaiah and Singh 2012) exploits this intuition to recognize social circles from places.

A social circle of a user (egocentric) is a set of contacts the user perceives as a logical group. Recognizing social circles could enable social network sites to deliver a high-quality user experience by reducing information overload (for example, by prioritizing updates from contacts), and enhancing privacy controls (for example, by providing a fine-grained control on who to share information with).

Currently, social network sites require users to manually create and maintain social circles (for example, circles on Google+ and groups on Facebook), which is tedious and time consuming (Lampinen et al. 2011). Alternatively, community detection algorithms are used to recognize social circles automatically. However, communities detected from a network of acquaintanceships (for example, "friendship" on Facebook and Google+) are coarser than social circles. Further, community detection presupposes that the global structure of the network is known.

Platys Social learns social circles by exploiting place information. It is implemented within the Platys middleware (Murukannaiah and Singh 2015) and employs information locally available on a user's mobile device. Platys Social operates as follows: (1) Construct a contact co-occurrence graph, whose nodes are the contacts of a user, and add an edge between two nodes if the user meets the two contacts at the same place. (2) Assign a weight to each edge proportional to the frequency with which the user meets corresponding contacts at the same place. (3) Find overlapping communities in the contact co-occurrence graph using Clique Percolation Method (Palla et al. 2005). Treat each community that corresponds to a social circle. Further, within each social circle, edge weights can be used to distinguish strong and weak contacts.

Evaluation

We evaluated Platys Social in a user study of six users. The users in the study carried a smartphone installed with Platys for the duration of the study. Each user recorded the places he or she visited and social circles (including strong and weak contacts) encountered on a daily basis. We measured the accuracy of Platys Social as the similarity between the social circles reported by users and those learned by Platys Social:

$$\text{accuracy} = \frac{|\text{learned circles} \cap \text{reported circles}|}{|\text{learned circles} \cup \text{reported circles}|}$$

We compared three variants of Platys Social depending on how edges are added to the contact co-occurrence graph. Stay points: add an edge between two contacts if two contacts are found (through Bluetooth devices) at the same stay point (determined from the Wi-Fi access point log). Interactions: add an edge between two contacts if a user's interaction includes both contacts (determined from email, call, and text logs). Place: add an edge in either of the above cases, according to the intuition that a place has both spatial and social attributes. As shown in figure 5, Platys Social performs best when places are defined using both spatial and social attributes.

Privacy Reasoning and Enforcement

A key element of our work is the use of collaborative information sharing where devices share and integrate knowledge about their place. Consequently, users must protect their privacy by controlling the release of information and how it is shared. Murukannaiah and Singh (2014, 2015) and Zavala et al. (2011) discuss architectures on which place-aware applications can be engineered. Devices might interact directly or through services on the Internet. Users specify privacy policies that regulate the disclosure of (1) sensor information to the server (for example,

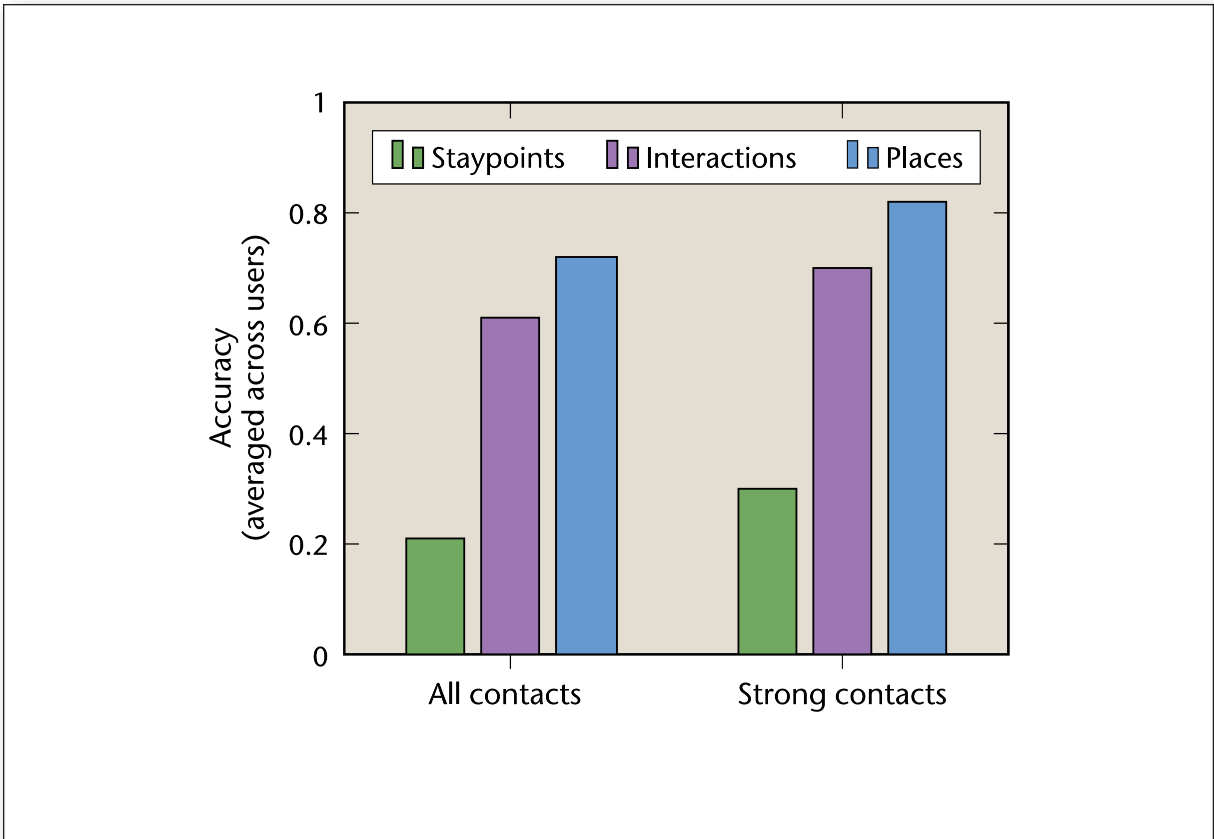

Figure 5. Comparing Accuracies of Social Circles Recognized Employing Stay Point, Interactions, and Place Information.

GPS information), (2) inferred context information to the server (for example, activity information), and (3) inferred context information to other users. Privacy policies are expressed as horn clause rules over the knowledge base.

Whenever a request is received, either at the server or at a device, the privacy-control module fetches the static knowledge about the user (for example, personal information and defined groups), the dynamic context knowledge, and the user-specified privacy preferences. Access rights are obtained by performing backward reasoning. Additionally, when access is allowed and according to the user-defined sharing preferences, certain pieces of the information might be obfuscated in order to protect user privacy. Privacy rules are defined as Jena rules (Carroll et al. 2004), and the Jena reasoning engine is used to perform the reasoning. For the devices, we use the AndroJena port of Jena for Android.[3]

Policies for Information Sharing

Privacy policies are represented as rules that describe which information a user is willing to share, with whom, and under what conditions. Conditions can be defined based on attributes such as a user's current location, current activity, or any other dynamic attribute. We rely heavily on the notion of group to define the subjects who are allowed to access certain information. A user can manage different networks of friends, and assign a variety of group-level privacy preferences accordingly. Example policies are "share detailed contextual information with family members all the time," "share my activity with friends all the time except when I am attending a lecture," and "do not share my sleeping activity with teachers on weekdays from 9 AM to 5 PM."

Policies for Obfuscating Shared Information

Users need to be in control of the release of their personal information at different levels of granularity, from raw sensed data to high-level inferred place information. Besides being able to specify which information a user is willing to share, we can specify how that information should be shared. A user can disclose information with different accuracy levels; for instance, he or she may be willing to reveal to close friends the exact room and building on which he or she is located, but only the vicinity or town to others. Furthermore, a user may decide not to disclose his or her location to advertisers.

We have built generalization models for location and activity that are based on hierarchies over location and activity entities. The models take advantage of the hierarchical nature of location and activity information, which is evident by the part-of or contained relations between location entities and the compositional nature of activities entities. The policies allow us to specify at which level the information is to be revealed. When a query for location or activity information is received, the reasoner will not only conclude whether the information can be shared or not, but also at what level in the hierarchy the information should be shared and only the corresponding triples are shared. For example, if location information should be shared at the *City* level, then triples containing location information with instances of entities below *City* in the hierarchy are not shared.

Energy-Efficient Location Sensing

Location-based services (LBSs) rely on global localization techniques such as GPS and Skyhook to obtain referenceable coordinates of the device. Pinpointing the location of a device on Earth with respect to an absolute reference point can be extremely challenging. GPS currently operates 31 satellites. Skyhook, which combines Wi-Fi and cellular signal fingerprints with GPS coordinates for indoor positioning, performs extensive war-driving in the cities where its services are provided.

Solving a much simpler problem can provide similar benefits to a particular class of LBS applications. We focus on what we call the location matching problem: “In an arbitrary location, can a smartphone efficiently detect whether or not it had previously visited this location?” This problem is much simpler than global positioning because it does not need to know the relative distance between two different locations on a geographical plane.

We approach the problem by pairing locations with an event (that is, a set of actions performed by the user at that location). For instance, when users are in a conference room, they mute the ring tone, or when they are in a gym, they play their favorite play list from a music app. By recording such events and corresponding cellular signal statistics at that location, we can easily identify the event for that location and reproduce that event (if needed) by matching the currently received cellular signals with those stored in the database. None of these applications require global positioning; they can function just as well with location matching.

Cellular Signal Signatures

We developed an Android application to collect cellular signal signatures per event. A signature is defined as the set of probability density functions (PDFs) of signal strengths from all observable base stations (BSs) when the smartphone is associated with that event. We collected data from around 40 volunteers from a university campus area for a period of more than three weeks.

The smartphones carried by the users receive signals from one or more cell towers (max seven) constantly. By analyzing the data set, we found that utilizing the detailed statistical information of cellular signals alone is sufficient to identify the event accurately. Cellular signals are received at no extra cost in mobile devices and have ubiquitous connectivity. Hence, we achieve continuous context sensing with minimal extra energy overhead.

An Autotuned Event-Sensing Algorithm

To better utilize the detailed statistical information recorded in our signatures, we design an event-sensing algorithm (ATiS) with autotuning capabilities. The idea is that the closer the input signal strengths match with the signature database, the more accurate the event estimate is. If the probability of seeing a particular signal strength within the PDF of a BS is high and the probability of the BS observed when performing an event is high, the total argument is maximized and hence we get a close match with the corresponding signature.

Finally, a signature threshold range C_L and C_U representing the lower bound and upper bound determines the event. The algorithm learns adaptively from its mistakes by evaluating itself against ground truth. Note that the values of $[C_L, C_U]$ are initialized with [1, 0] initially. During automatic tuning, C_L decreases and C_U increases, respectively, based on the ground truth to provide a tight bound for signature thresholds.

Accuracy and Energy Measurements

We evaluate for both accuracy and energy efficiency. For our analysis, we use active hour trace (AHT) of the user logs, which we assume to be from 07:00 hours to 23:00 hours because this is the time period during which most users will be active and mobile in general. We use the first 70 percent of the logs for training and the remaining 30 percent for evaluation.

We evaluate the output of the algorithms at each time instant (here 20 seconds), and compare it with the ground truth from the logs. False positive ratio (*FP*(%)) is defined as the percentage of cases when the algorithm detects an event when it is not available in the ground truth divided by the total number of cases. Similarly, false negative ratio (*FN*(%)) is defined as the percentage of cases when the algorithm does not detect an event when it is available in the ground truth divided by the total number of cases. As shown in figure 6a, we find that to achieve very low *FP*(%) values, base station set (BSSET; see Rahmati and Zhong [2007]) and mean squared error (MSE; see Prasithsangaree, Krishnamurthy, and Chrysanthis 2002; Varshavsky et al. 2007)–based algorithms need very

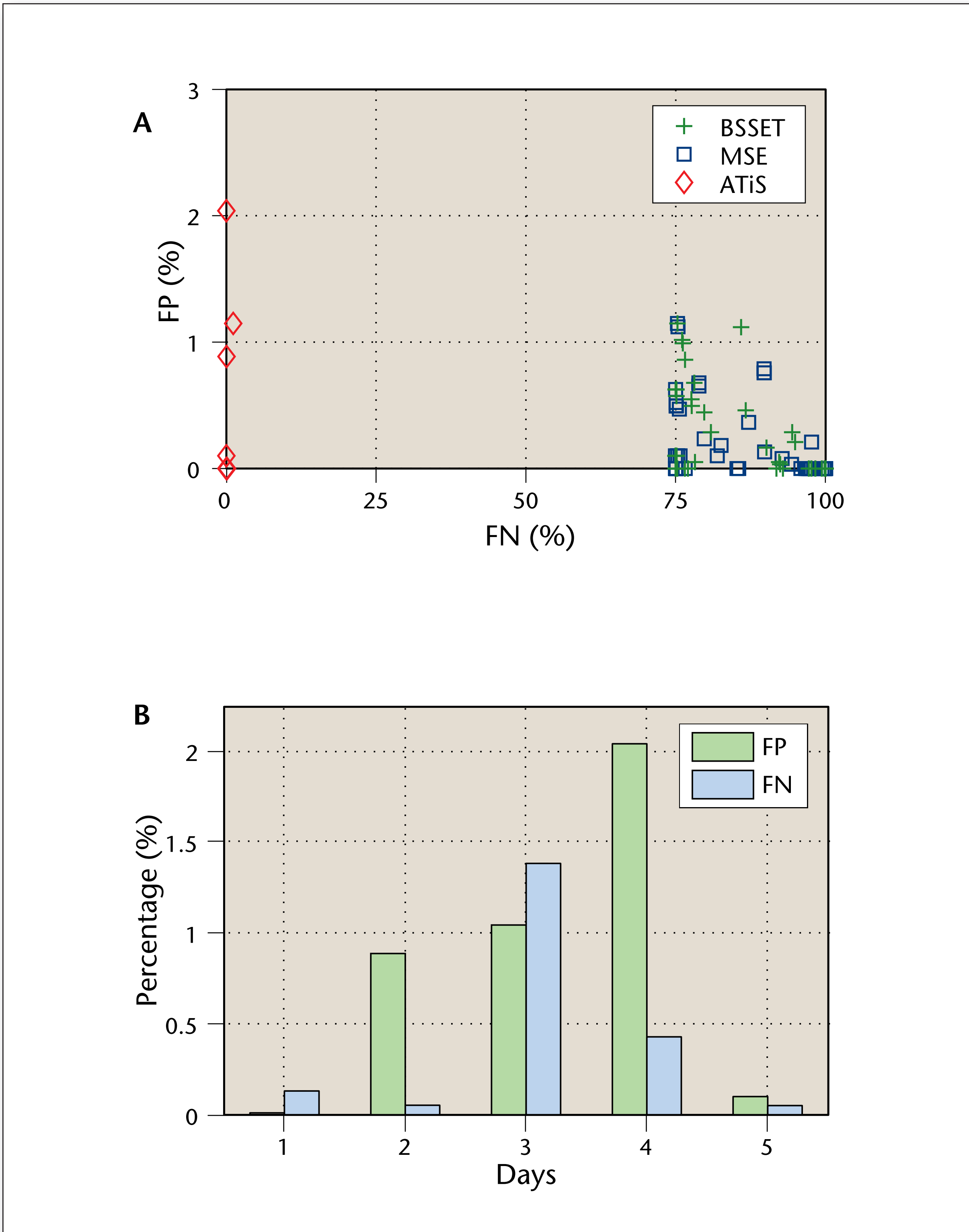

Figure 6. FP Versus FN Values.

(a) *FP* versus *FN* values for a random user. ATiS achieves very low *FP* and *FN* values simultaneously. (b) Variation in *FP* and *FN* values for 5 consecutive days for a random user in the data set.

Item	Energy Consumed (mWh)
Our System	0.0173
Wi-Fi Scan	0.1185
Accelerometer	0.6670
GPS	1.5800

Table 3. Energy Consumption Per Second Between Our System and Other Techniques for Continuous Location Sensing.

high threshold values, which results in high *FN*(%) values. But ATiS achieves low *FP*(%) and *FN*(%) values simultaneously. However, the values differ for every individual user and on a daily basis as shown in figure 6 (b). Overall, we achieved average *FP*(%) and *FN*(%) values of 1.10 percent and 0.19 percent, which is very close to the ideal case of zero *FP*(%) and *FN*(%) values.

We use a digital power monitoring device from Monsoon Solutions[4] to measure the energy consumptions for event sensing on Android smartphones (Google Nexus One). Extensive trials are done to avoid sensitive fluctuations in power consumption. Table 3 shows the general energy consumptions per second for location sensing by our system and other available techniques. However, the total amount of energy consumption varies differently depending on the application scenarios.

Discussion

The Platys project builds on a semantic concept of place to facilitate developing context-aware mobile applications that can enhance their users' experience. A place in Platys goes beyond location to include associated time spans, activities, people, roles, and objects. Our resulting context model is supported by an ontology in OWL.

Place recognition is performed using a semisupervised EM algorithm as well as standard machine-learning classifiers. Our approach allows to capture nuances in how a user perceives places and is able to recognize (1) user's place and activity at different levels of granularity; (2) disjoint spatial regions as a single place; and (3) the same spatial region as more than one place (for the same user and for different users). Performance for recognizing place at a general level (at home versus at work versus elsewhere) using machine-learning classifiers is higher than that reported in existing works. Performance for recognizing place using a semisupervised EM algorithm was generally better than two stay-point approaches used for comparison. Location plays an important role in place recognition. We have addressed the problem of energy-efficient location sensing.

A place is naturally associated with a social context. We have proposed an approach to recognize social circles by exploiting place information. Our approach performs best when places are defined using both spatial and social attributes.

To provide users with privacy to protect the personal information their mobile devices are collecting, we define privacy and information sharing policies. The policies are expressed in the SemanticWeb languages OWL and RDF. Our policies ensure context-dependent release and obfuscation of information in accordance to the user preferences.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants 0910838, 0910868, 0910846, and 1016216.

Notes

1. www.twofortyfouram.com.
2. betalabs.nokia.com/trials/nokia.
3. code.google.com/p/androjena.
4. www.msoon.com.

References

Baader, F.; Calvanese, D.; McGuinness, D. L.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. New York: Cambridge University Press.

Baldauf, M.; Dustdar, S.; and Rosenberg, F. 2007. A Survey on Context-Aware Systems. *International Journal of Ad Hoc and Ubiquitous Computing* 2(4): 263–277. dx.doi.org/10.1504/IJAHUC.2007.014070

Bao, L., and Intille, S. S. 2004. Activity Recognition from User-Annotated Acceleration Data. In *Pervasive Computing,* Lecture Notes in Computer Science volume 3001, 1–17. Berlin: Springer. dx.doi.org/10.1007/978-3-540-24646-6_1

Bechhofer, S.; van Harmelen, F.; Hendler, J.; Horrocks, I.; Mcguinness, D. L.; Patel-Schneider, P. F.; and Stein, L. A. 2007. Owl Web Ontology Language Reference. Technical Report, W3C. Boston, MA: The World Wide Web Consortium.

Carroll, J. J.; Dickinson, I.; Dollin, C.; Reynolds, D.; Seaborne, A.; and Wilkinson, K. 2004. Jena: Implementing the Semantic Web Recommendations. In *Proceedings of the 13th International World Wide Web Conference,* 74–83. New York: Association for Computing Machinery.

Chen, G., and Kotz, D. 2000. A Survey of Context-Aware Mobile Computing Research. Technical Report, Dartmouth College Computer Science Department, Hanover, NH.

Chen, H.; Finin, T.; and Joshi, A. 2005. *The SOUPA Ontology for Pervasive Computing,* 233–258. Berlin: Birkhauser Publishing Ltd. / Springer.

Dey, A. K.; Abowd, G. D.; and Salber, D. 2001. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction* 16(2): 97–166. dx.doi.org/10.1207/S15327051HCI16234_02

Eagle, N., and Pentland, A. 2006. Reality Mining: Sensing Complex Social Systems. *Personal and Ubiquitous Computing*

10(4): 255–268. dx.doi.org/10.1007/s00779-005-0046-3

Fahy, P., and Clarke, S. 2004. Cass: A Middleware for Mobile Context-Aware Applications. Paper presented at the MobiSys 2004 Workshop on Context Awareness, Boston, MA, 6 June.

Gu, T.; Pung, H. K.; and Zhang, D. Q. 2004. A Middleware for Building Context-Aware Mobile Services. In *2004 IEEE 59th Vehicular Technology Conference. Spring,* volume 5, 2656–2660. Piscataway, NJ: Institute for Electrical and Electronics Engineers.

Hang, C.-W.; Murukannaiah, P. K.; and Singh, M. P. 2013. Platys: User-Centric Place Recognition. In Activity Context-Aware System Architectures: Papers from the AAAI 2013 Workshop. Technical Report WS-13-05. Palo Alto, CA: AAAI Press.

Hariharan, R., and Toyama, K. 2004. Project Lachesis: Parsing and Modeling Location Histories. In *Geographic Information Science: 3rd International Conference*, Lecture Notes in Computer Science volume 3234, 106–124. Berlin: Springer.

Korpipaa, P.; Mantyjarvi, J.; Kela, J.; Keranen, H.; and Malm, E.-J. 2003. Managing Context Information in Mobile Devices. *IEEE Pervasive Computing* 2(3): 42–51. dx.doi.org/10.1109/MPRV.2003.1228526

Lampinen, A.; Lehtinen, V.; Lehmuskallio, A.; and Tamminen, S. 2011. We're in It Together: Interpersonal Management of Disclosure in Social Network Services. In *Proceedings of the 2011 ACM SIGCHI Conference on Human Factors in Computing Systems,* 3217–3226. New York: Association for Computing Machinery. dx.doi.org/10.1145/1978942.1979420

Murukannaiah, P. K., and Singh, M. P. 2015. Platys: An Active Learning Framework for Place-Aware Application Development and Its Evaluation. *ACM Transactions on Software Engineering and Methodology* 24(3): 1–33.

Murukannaiah, P. K., and Singh, M. P. 2014. Xipho: Extending Tropos to Engineer Context-Aware Personal Agents. In *Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems,* 309–316. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Murukannaiah, P. K., and Singh, M. P. 2012. Platys Social: Relating Shared Places and Private Social Circles. *IEEE Internet Computing* 16(3): 53–59. dx.doi.org/10.1109/MIC.2011.106

Palla, G.; Derényi, I.; Farkas, I. J.; and Vicsek, T. 2005. Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society. *Nature* 435(7043): 814–818. dx.doi.org/10.1038/nature03607

Prasithsangaree, P.; Krishnamurthy, P.; and Chrysanthis, P. 2002. On Indoor Position Location with Wireless LANs. In *Proceedings of the 13th IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications.* Piscataway, NJ: Institute for Electrical and Electronics Engineers.

Rahmati, A., and Zhong, L. 2007. Context-for-Wireless: Context-Sensitive Energy-Efficient Wireless Data Transfer. In *Proceedings of the 5th International Conference on Mobile Systems, Applications, and Services.* New York: Association for Computing Machinery.

Román, M.; Hess, C.; Cerqueira, R.; Ranganathan, A.; Campbell, R. H.; and Nahrstedt, K. 2002. A Middleware Infrastructure for Active Spaces. *IEEE Pervasive Computing* 1(4): 74–83. dx.doi.org/10.1109/MPRV.2002.1158281

Salber, D.; Dey, A. K.; and Abowd, G. D. 1999. The Context Toolkit: Aiding the Development of Context-Enabled Applications. In *Proceedings of the 1999 ACM SIGCHI Conference on Human Factors in Computing Systems,* 434–441. New York: Association for Computing Machinery.

Schilit, B.; Adams, N.; and Want, R. 1994. Context-Aware Computing Applications. In *First IEEE Workshop on Mobile Computing Systems and Applications*, 85–90. Los Alamitos, CA: IEEE Computer Society.

Schilit, B. N.; Adams, N.; Gold, R.; Tso, M. M.; and Want, R. 1993. The Parctab Mobile Computing System. In *Proceedings, IEEE Fourth Workshop on Workstation Operating Systems,* 34–39. Piscataway, NJ: Institute for Electrical and Electronics Engineers.

Varshavsky, A.; de Lara, E.; Hightower, J.; LaMarca, A.; and Otsason, V. 2007. GSM Indoor Localization. *Pervasive Mobile Computing* 3(6): 698–720. dx.doi.org/10.1016/j.pmcj.2007.07.004

Want, R.; Hopper, A.; Falcao, V.; and Gibbons, J. 1992. The Active Badge Location System. *ACM Transactions on Information Systems* 10(1):91–102. dx.doi.org/10.1145/128756.128759

Weiser, M. 1999. The Computer for the 21st Century. *Mobile Computing and Communications Review* 3(3): 3–11. dx.doi.org/10.1145/329124.329126

Witten, I. H., and Frank, E. 2002. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. San Francisco: Morgan Kaufmann.

Zavala, L.; Dharurkar, R.; Jagtap, P.; Finin, T.; and Joshi, A. 2011. Mobile, Collaborative, Context-Aware Systems. In Activity Context Representation: Papers from the 2011 AAAI Workshop. Technical Report WS-11-04. Palo Alto, CA: AAAI Press.

Zheng, K.; Zheng, Y.; Xie, X.; and Zhou, X. 2012. Reducing Uncertainty of Low-Sampling-Rate Trajectories. In *Proceedings of the 28th IEEE International Conference on Data Engineering,* 1144–1155. Los Alamitos, CA: IEEE Computer Society.

Laura Zavala is an assistant professor of computer science at Medgar Evers College of the City University of New York (CUNY). Previously, she was a research scientist at the University of Maryland, Baltimore County (UMBC). She has several years of experience in applications of artificial intelligence to information search, recommender, social, and mobile systems. She has a B.S. in computer science and an M.S. in artificial intelligence from the University of Veracruz, Mexico. She has a Ph.D. in computer science and engineering from the University of South Carolina.

Pradeep K. Murukannaiah is a Ph.D. student in computer science at North Carolina State University. His research interests include mobile and social computing, machine learning, and multiagent systems. Murukannaiah has a B.E. in information science and engineering from University Visveswaraya College of Engineering, Bangalore, and an M.S. in computer science from North Carolina State University.

Nithyananthan Poosamani is a Ph.D. student in computer science at North Carolina State University. He received his B.E. in electronics and communication engineering from PSG College of Technology, Anna University, Tamil Nadu, India. He worked as a software engineer in 3G Radio Network Controllers for two years before joining North

Carolina State University. His areas of research interests include human mobility pattern analysis, mobile systems, and energy-efficient localization techniques.

Tim Finin is a professor of computer science and electrical engineering at UMBC. He has more than 30 years of experience in applications of artificial intelligence to problems in information systems and language understanding. He received an S.B. degree in electrical engineering from the Massachusetts Institute of Technology and a Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign. He has held full-time positions at UMBC, Unisys, the University of Pennsylvania, and the MIT AI Laboratory. He is currently an editor in chief of the Elsevier *Journal of Web Semantics* and a coeditor of the Viewpoints section of the *Communications of the ACM.*

Anupam Joshi is a professor of computer science and electrical engineering at UMBC. He obtained a B. Tech degree in electrical engineering from IIT Delhi, and a master's and Ph.D. in computer science from Purdue University. He has published more than 50 technical papers and has obtained research support from NSF, NASA, DARPA, DoD, IBM, AetherSystens, HP, AT&T, and Intel.

Injong Rhee is a professor of computer science at North Carolina State University, Raleigh. His areas of research interests include computer networks, congestion control, wireless ad hoc networks, and sensor networks. He is editor of the *IEEE Transactions on Mobile Computing*.

Munindar P. Singh is a professor of computer science at North Carolina State University. His research focuses on the study of interactions among autonomous parties, with a special interest in multiagent systems and context-aware computing. Singh is an IEEE Fellow, the current editor in