Platys: From Position to Place-Oriented Mobile Computing

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

Abstract

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 they occur. 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: (a) representing places, (b) recognizing places, and (c) engineering place-aware applications. We describe the approaches we have developed for addressing these problems and related sub-problems. 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.

Introduction

Mobile applications that automatically adapt to their surrounding circumstances will lead to an enhanced user experience. Emerging mobile applications exploit a users location to deliver personalized services. In current practice, the user's location is captured at the level of *position*, i.e., 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 – descriptions that combine 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: (a) changing phone settings (e.g., turn ringer off during a meeting and turn it back on at the end of the meeting); (b) downloading relevant information (e.g., the map of an amusement park, museum, or any place the user visits); (c) annotating images or other media; (d) filtering content such as alerts, notifications, and customized ads; (e) changing the ambiance (e.g., playing music); (f) showing (place-dependent) reminders from to-do lists; and (g) pushing recommendations to the user when the situation seems appropriate.

In this paper, we report on our efforts pertaining to the Platys project. A semantic model of user-centric 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 semi-supervised expectation maximization algorithm. The recognition is based on data captured from a user's smartphone: location, sensor readings, wifi, 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 an 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 (e.g., 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.

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1] http://www.twofortyfouram.com/

[2] https://betalabs.nokia.com/trials/nokia-situations

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 (e.g., 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 side bar), non-trivial 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 reminder of this paper is organized as follows. In the next section we provide a formal definition of a user-centric 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-centric and attempt recognize places in a privacy-preserving manner. In (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.

Sidebar: 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) were human activities are enhanced with devices that are all around but unnoticeable to the user and which provide services that adapt to the circumstances in which they are used. (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, etc.), 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/or register for context information. Some also include formal context modeling to share contextual information among heterogeneous entities, security and privacy, inference mechanisms, and/or agent capabilities. (Chen and Kotz 2000; Baldauf, Dustdar, and Rosenberg 2007) provide surveys of developments and applications in the field.

Semantic Place Model

We define a (user-centric) 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-centric, 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 (e.g., *work office*, *home office*, *lab* or a *café*) 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.
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 others via 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, i.e., 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, i.e., fine-grained activities make up more general ones. $Ambiance$ encapsulates concepts describing the environment of the $User$ (e.g., 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

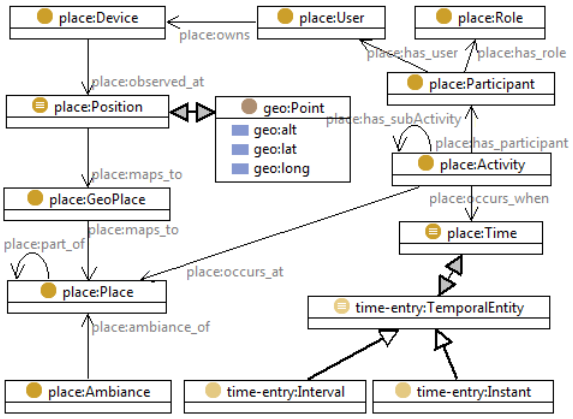


Figure 1: *The Place ontology* models the concept of *place* in terms of activities that occur there.

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 (i.e., 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 WiFi access points). Given the $Position$ of the user's device, we assert the corresponding triples into the KB (see Figure 2). Then, we use additional online resources, specifically GeoNames spatial KB (RDF version) and its associated services, to infer the user's $GeoPlace$ by:

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 (see Figure 2)
4. Using OWL inference to obtain the triples corresponding to the spatial containment of entities (transitivity of the $part\_of$ relationship)
5. Using ad-hoc property chains (Figure 3) to infer knowledge about a user's geoplace based on the places her associated device is observed.

```
@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 ?b)
]
```

Figure 2: KB assertions (left) in Turtle and a Jena rule used to integrate knowledge from GeoNames (right).

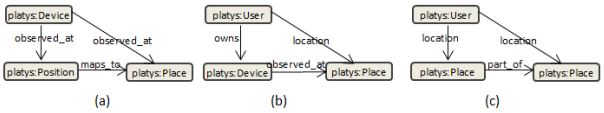


Figure 3: Property chain axioms asserting facts about a user's location: a) Device is *observed_ at* the place whose position it *maps_ to*; b) User's *location* is the place where her associated device is *observed_ at*; c) Generalization of user *location* based on spatial containment (*part_ of*).

Recognizing User-Centric 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 semi-supervised expectation maximization (EM) 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.

Semi-supervised expectation maximization (EM)

We developed (Hang, Murukannaiah, and Singh 2013) a semi-supervised expectation maximization (EM) algorithm to recognize user-centric places. Our approach: (a) recognizes subjective places, (b) does not require manual tuning of place radius and duration, and (c) 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 dataset consisting of a data instance for each sensor reading and user label. Further, consider a training set as a subset of the above dataset, 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 user 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, WiFi, and Bluetooth) and prompted users to label places at random intervals.

We compared our approach with two staypoint approaches (Hariharan and Toyama 2004; Zheng et al. 2012). Platys cannot be directly compared with a staypoint approach since the latter only recognizes whether a user is in some staypoint or not (not the specific staypoint as there are no labels). To enable a fair comparison, we implemented two versions of Platys: (a) *Place-or-not*, which only recognizes if a user is in one of the labeled places or not, and (b) *Which-place*, which recognizes the specific place. Figure 4 shows the comparison.

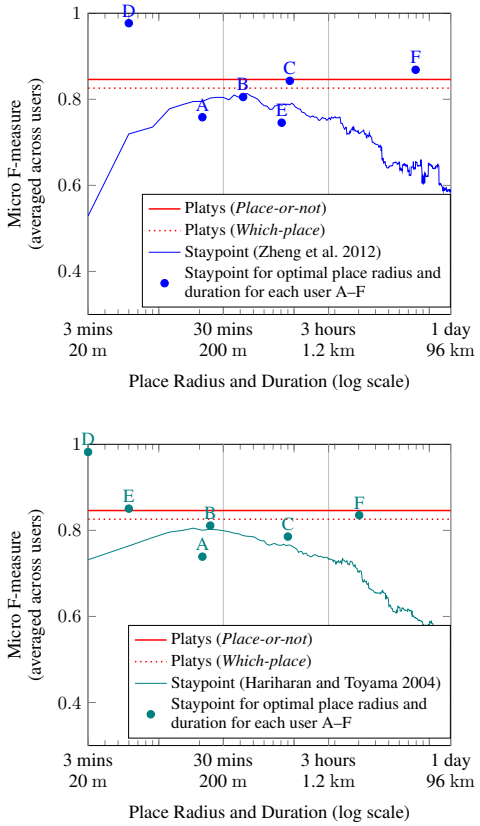


Figure 4: Comparing Platys with two staypoint approaches that distinguish places from nonplace, but do not identify a place.

The following are our main findings.

1. Platys (Place-or-not) performs better than both staypoint approaches compared with. Importantly, the F-measures for Platys, unlike those of staypoint approaches, are straight lines since since they do not depend on place radius and duration.
2. A staypoint approach with optimal place parameter values may perform better than Platys for some users. However, as shown, optimal place parameter values vary from user to user.
3. Place-or-not is an upper bound on Which-place. However, in most case, 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.

Activity Classification

We used supervised machine learning algorithms to recognize activity (e.g., “sleeping”, “walking”, “sitting”, “cooking”) and place (e.g., ”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 collected data for five users over the course of two weeks using Android smartphones and an interactive data collection program. The information collected includes location, ambiance light and noise, wifi scanning, bluetooth scanning, current calendar event (if any), sensors readings (accelerometer, magnetic field, orientation, and proximity), call statistics (missed calls, answered calls, and duration), and phone state (idle, in use, etc.). We collect the data every two, five, or twelve minutes (set by the user according to current activity duration) for a period of one minute. 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 (e.g., *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 have 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. Using the Weka Machine Learning Algorithms Toolkit (Witten and Frank 2002), 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 mid-level detailed activity recognition task for a particular user and nine everyday activities using 10 cross fold validation and 66% split validation testing options. Accuracy levels are comparable to those reported on

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

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.

(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 (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, general activities (see Table 2). Our 99% accuracy for "at home vs. at work vs. elsewhere" is higher than the one reported in (Eagle and (Sandy) 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 (ego-centric) 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 (a) reducing information overload (e.g., by prioritizing updates from contacts), and (b) enhancing privacy controls (e.g., 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 (e.g., *circles* on Google+ and *groups* on Facebook), which is tedious an 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 (e.g., "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 places 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.

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.

Assign a weight to each edge proportional to the frequency with which the user meets corresponding contacts at the same place.

Find overlapping communities in the contact co-occurrence graph using Clique Percolation Method (Palla et al. 2005). Treat each community 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 smart phone 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,

$$\text{accuracy} = \frac{|\text{learned circles} \bigcap \text{reported circles}|}{|\text{learned circles} \bigcup \text{reported circles}|}$$

We compared three variants of Platys Social depending on how edges are added to the contact co-occurrence graph.

Staypoints: Add an edge between two contacts if two contacts are found (via Bluetooth devices) at the same staypoint (determined from WiFi 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. In (Murukannaiah and Singh 2014; 2015;

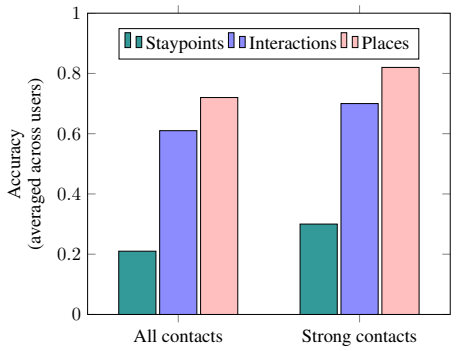


Figure 5: Comparing accuracies of social circles recognized employing staypoint, interactions, and place information.

Zavala et al. 2011) we 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 (a) sensor information to the server (e.g., GPS information), (b) inferred context information to the server (e.g., activity information) and (c) inferred context information to other users.

Privacy policies are expressed as horn clause rules over the knowledge base. In our prototype system, the focus is not on the protocol used by devices to exchange information, but on the privacy control mechanisms. Therefore, requests are simple messages with the required information embedded in them. Whenever a request is received, either at the server or at a device, the privacy control module fetches the static knowledge about the user (e.g., personal information and defined groups), the dynamic context knowledge and the user specified privacy preferences. Access rights are obtained by performing backward reasoning confirms conclusions by verifying conditions. 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 Jena reasoning engine is used to perform the reasoning. For the devices, we use the AndroJena (Lorecarra 2009) port of Jena for Android.

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 9am to 5pm."*. Figure 6 shows the representation of the first rule as a Jena rule (left) and the results on a test screen we provide to observe

```
@prefix kb: <http://example.org#device>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.

[AllowFamilyRule:
(?requester kb:contextAccess kb:userPermitted)
<-
(?requester rdf:type kb:requester)
(?groupFamily foaf:member ?requester)
(?groupFamily foaf:name "Family")
]
```

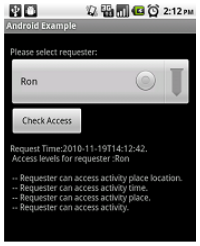


Figure 6: Left: Jena rule for expressing the policy *"share detailed contextual information with family members all the time."* Right: Android device screen with reasoning results. It shows access levels for requester *"Ron"* who is a member of the group *Family*.

the results of the reasoning engine (right).

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, she may be willing to reveal to her close friends the exact room and building on which she is located, but only the vicinity or town to others. Furthermore, a user may decide not to disclose her location to advertisers.

We have built generalization models for location and activity which 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 (LBS) rely on global localization techniques such as GPS and Skyhook to obtain reference-able 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 their 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 smart phone 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 (i.e., a set of action(s) performed by the user at that location). For instance, when a user is in a conference room, they mute the ring tone, or when they are in a gym, they play their favourite play list from a music. 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 (BS) when the smart phone is associated with that event. We collected data from around 40 volunteers from a university campus area for over a period of three weeks.

The smart phones carried by the users receive signals from one or more cell towers (max seven) constantly. By analyzing the dataset, we found that utilizing the detailed statistical information of cellular signals alone is sufficient to accurately identify the event. 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.

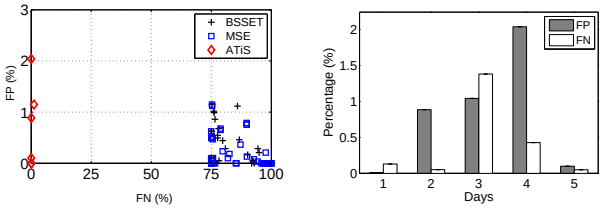
An Auto-tuned Event Sensing Algorithm

To better utilize the detailed statistical information recorded in our signatures, we design an event sensing algorithm (ATiS) with auto-tuning 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 hrs to 23:00 hrs because this is the time period during which most users will be active and mobile in general. We use first 70% of the logs for training and remaining 30% for evaluation.



(a) (b)

Figure 7: (a) FP Vs. 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 dataset.

Table 3: Energy consumption per second between our system and other techniques for continuous location sensing.

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

We evaluate the output of the algorithms at each time instant (here $20\,sec$), 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 7 (a), we find that to achieve very low $FP(\%)$ values, Base Station Set: BSSET (Rahmati and Zhong 2007) and Mean Squared Error: MSE (Prasithsangaree, Krishnamurthy, and Chrysanthis 2002; Varshavsky et al. 2007) based algorithms need very 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 7 (b). Overall, we achieved an average $FP(\%)$ and $FN(\%)$ values of $1.10\,\%$ and $0.19\,\%$ which is very close to the ideal case of zero $FP(\%)$ and $FN(\%)$ values.

We use a digital power monitoring device from Monsoon Solutions[3] to measure the energy consumptions for event sensing on Android smart phones (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 appli-

[3] http://www.msoon.com/LabEquipment/PowerMonitor/

cation scenarios.

Conclusions

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, activates, people, roles and objects. Our resulting context model is supported by an ontology in OWL.

Place recognition is currently performed on individual activity recognition but we make use of information about nearby devices (through Bluetooth and WiFi scanning) and are working on a collaborative approach. Performance for recognizing place at a general level (home vs. work vs. elsewhere) is higher than that reported in existing works. Compared to unsupervised staypoint approaches, the F-measures for Platys, unlike those of staypoint approaches, are straight lines since since they do not depend on place radius and duration. 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 places information. Our approach performs best when places are defined using both spatial and social attributes.

In order 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 Semantic Web languages OWL and RDF. Our release policies ensure context dependent release of information in accordance to the user preferences as well as obfuscation of shared information.

Acknowledgments

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

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, NY, USA: Cambridge University Press.

Baldauf, M.; Dustdar, S.; and Rosenberg, F. 2007. A survey on context-aware systems. *Int. J. Ad Hoc Ubiquitous Comput.* 2(4):263–277.

Bao, L., and Intille, S. S. 2004. Activity recognition from user-annotated acceleration data. In *Pervasive*, 1–17.

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.

Carroll, J. J.; Dickinson, I.; Dollin, C.; Reynolds, D.; Seaborne, A.; and Wilkinson, K. 2004. Jena: implementing the semantic web recommendations. In *Proc. 13th Int. World Wide Web Conf.*, 74–83. New York, NY, USA: ACM.

Chen, G., and Kotz, D. 2000. A survey of context-aware mobile computing research. Technical report, Dartmouth College Computer Science, Hanover, NH, USA.

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

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. *Hum.-Comput. Interact.* 16(2):97–166.

Eagle, N., and (Sandy) Pentland, A. 2006. Reality mining: sensing complex social systems. *Personal Ubiquitous Comput.* 10:255–268.

Fahy, P., and Clarke, S. 2004. Cass a middleware for mobile context-aware applications. In *Workshop on Context Awareness, MobiSys*.

Gu, T.; Pung, H. K.; and Zhang, D. Q. 2004. A middleware for building context-aware mobile services. In *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, volume 5, 2656–2660Vol.5.

Hang, C.-W.; Murukannaiah, P. K.; and Singh, M. P. 2013. Platys: User-centric place recognition. In *Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Hariharan, R., and Toyama, K. 2004. Project Lachesis: Parsing and modeling location histories. In *3rd Int. Conf. on Geographic Information Science*, 106–124. 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.

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 SIGCHI Conference on Human Factors in Computing Systems*, 3217–3226. New York: ACM.

Lorecarra. 2009. Androjena : Jena android porting.

Murukannaiah, P. K., and Singh, M. P. 2012. Platys Social: Relating shared places and private social circles. *IEEE Internet Computing* 16(3):53–59.

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. Paris: IFAAMAS.

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.

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.

Prasithsangaree, P.; Krishnamurthy, P.; and Chrysanthis, P. 2002. On indoor position location with wireless lans. In *IEEE PIMRC*.

Rahmati, A., and Zhong, L. 2007. Context-for-wireless:

Context-sensitive energy-efficient wireless data transfer. In *ACM Mobisys*.

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.

Salber, D.; Dey, A. K.; and Abowd, G. D. 1999. The context toolkit: Aiding the development of context-enabled applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, 434–441. New York, NY, USA: ACM.

Schilit, B.; Adams, N.; and Want, R. 1994. Context-aware computing applications. In *First Workshop on Mobile Computing Systems and Applications*, 85–90. Washington, DC, USA: IEEE Computer Society.

Schilit, B. N.; Adams, N.; Gold, R.; Tso, M. M.; and Want, R. 1993. The parctab mobile computing system. In *Workshop on Workstation Operating Systems*, 34–39.

Varshavsky, A.; de Lara, E.; Hightower, J.; LaMarca, A.; and Otsason, V. 2007. Gsm indoor localization. *Pervasive Mobile Computing* 3:698–720.

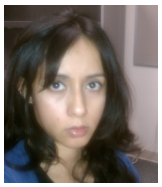
Want, R.; Hopper, A.; Falcão, V.; and Gibbons, J. 1992. The active badge location system. *ACM Trans. Inf. Syst.* 10(1):91–102.

Weiser, M. 1999. The computer for the 21st century. *Mobile Computing and Communications Review* 3(3):3–11.

Witten, I. H., and Frank, E. 2002. Data mining: practical machine learning tools and techniques with java implementations. *SIGMOD Rec.* 31:76–77.

Zavala, L.; Dharurkar, R.; Jagtap, P.; Finin, T.; and Joshi, A. 2011. Mobile, Collaborative, Context-Aware Systems. In *Proceedings of the AAAI Workshop on Activity Context Representation: Techniques and Languages*. AAAI.

Zheng, K.; Zheng, Y.; Xie, X.; and Zhou, X. 2012. Reducing uncertainty of low-sampling-rate trajectories. In *28th IEEE Int. Conf. on Data Engineering*, 1144–1155. Washington, DC: IEEE Computer Society.



Laura Zavala is an Assistant Professor of Computer Science at Medgar Evers College of 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 a 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 PhD 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 BE in information science and engineering from University Visveswaraya College of Engineering, Bangalore, and an MS in computer science from North Carolina State University.

Nithyananthan Poosamani is a PhD 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 the University of Maryland, Baltimore County (UMBC). He has over 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 MIT 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 co-editor of the Viewpoints section of the Communications of the ACM.

Anupam Joshi is a Professor of Computer Science and Electrical Engineering at the University of Maryland, Baltimore County (UMBC). He obtained a B. Tech degree in Electrical Engineering from IIT Delhi, and a Masters and Ph.D. in Computer Science from Purdue University. He has published over 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 in 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-Chief of ACM Transactions on Internet Technology, and a former Editor-in-Chief of IEEE Internet Computing.