Preprint draft of 'Laura Zavala, Radhika Dharurkar, Pramod Jagtap, Tim Finin and Anupam Joshi, Mobile, Collaborative, Context-Aware Systems, Proceedings of the AAAI Workshop on Activity Context Representation: Techniques and Languages, AAAI Press, August 2011.'

Mobile, Collaborative, Context-Aware Systems

Laura Zavala, Radhika Dharurkar, Pramod Jagtap, Tim Finin and Anupam Joshi
University of Maryland, Baltimore County, Baltimore, MD 21250 USA
{rzavala,radhika1,pramod1,finin,joshi}@umbc.edu

Abstract

We describe work on representing and using a rich notion of context that goes beyond current networking applications focusing mostly on location. Our context model includes location and surroundings, the presence of people and devices, inferred activities and the roles people fill in them. A key element of our work is the use of collaborative information sharing where devices share and integrate knowledge about their context. This introduces a requirement that users can set appropriate levels of privacy to protect the personal information being collected and the inferences that can be drawn from it. We use Semantic Web technologies to model context and to specify high-level, declarative policies specifying information sharing constraints. The policies involve attributes of the subject (i.e., information recipient), target (i.e., the information) and their dynamic context (e.g., are the parties copresent). We discuss our ongoing work on context representation and inference and present a model for protecting and controlling the sharing of private data in context-aware mobile applications.

Introduction

Applications for smartphones are rapidly evolving to take advantage of features available on the devices, especially localization capabilities. While location awareness is an important aspect of context-aware systems, context encompasses more aspects because other things of interest are also mobile and changing. Examples include ambiance, nearby people and resources, and the activities in which they are engaged. In previous work (Chen, Finin, and Joshi 2005; Chen and Joshi 2003) we presented an ontology to represent various types of contextual information in pervasive computing environments, specifically, smart meeting rooms. We have further generalized the model to a light-weight, upper-level context ontology (the *Place ontology*) that can be used to reason about a general notion of context, as well as to share contextual knowledge.

Our work is motivated by our vision of collective context determination where devices share and integrate knowledge about their context. In-situ P2P communication among (fixed and mobile) wireless devices based on opportunistic gossiping is used for sharing place information. Fixed devices such as sensors and APs can be used to summarize statistically the place information overheard from passing-by mobile devices. Collaborating participants cannot always be statically pre-identified; they frequently form dynamic adhoc coalitions. This paper includes a general architecture for these type of systems. Under these settings, users need support for appropriate levels of privacy to protect the personal information their mobile devices are collecting including the inferences that can be drawn from the information. For example, in a healthcare scenario, if a user has an accident, it might be right to disclose relevant information (medical records, history, etc.) to the paramedics on the scene and only while they are providing their services.

We advocate the adoption of semantic Web technologies in mobile, collaborative context aware systems for two main purposes: (i) creation of models for representing and reasoning about a high-level notion of context and (ii) specification of expressive policies to control the sharing of contextual information. We are developing a system to integrate all these ideas together.

We built a prototype system for a university environment which aggregates information from a variety of sensors on the phone (Google Android platform), online sources, as well as sources internal to the campus intranet, and individually infers the dynamic user activity using existing machine learning algorithms. For high-level, general activities, the accuracy of our system is better than existing works. For fined-grained, lower-level activities our system accuracy lowers. We expect this to improve as we incorporate models that allow for collaborative context inference. As a first step, the system allows sharing of contextual information directly between devices or through a server. To achieve this, each device has a knowledge base (KB) that aligns with our *Place ontology*. The system also implements a model for specifying and enforcing privacy through declarative policies. The policies allow users to specify situations under which they allow sharing of their context information as well as the level of accuracy at which such information should be shared. In the next section we describe in more detail a general architecture for mobile collaborative context aware systems on which we base our work, including a semantic model of context. In section 3, we discuss our work on individual activity recognition. In section 4, we present a model for specifying and enforcing privacy through semantic Web-based declarative policies. Finally, we discuss related work and future directions of our work.

General Architecture

Our focus is on semantic context representation for collaborative, mobile context-aware systems where the devices, i.e., smartphones, share and integrate knowledge about their context. Peer to peer communication among co-located nearby wireless devices based on opportunistic gossiping is used for sharing place information. Fixed devices such as sensors and access points (APs) can be used to reposit, share, and even summarize statistically the place information overheard from passing-by mobile devices.

Figure 1 depicts a general interaction architecture for this type of systems. Sensors on devices sense the local context of the user, using mobility tracking and ambient sensing such as light, sound, and motion. The network component opportunistically gathers and disseminates local context information to neighboring fixed or mobile wireless devices. Its policy engine verifies the release policies to ensure context dependent release of information in accordance to the user preferences. Devices might interact directly or through services on the Internet. Inferences such as current activity can be drawn from the information collected by the sensors, the context information gathered, and additional resources (e.g., the user calendar and open geolocation KBs). The sensor's raw data as well as the inferred context knowledge is stored in a local knowledge base on the device. Context-aware applications and network components may use this context knowledge to enhance their functionality. The locally inferred context knowledge can also be sent to context-aware services located on the Internet. These services on the Internet, verify, if needed, the statements (proof) of the clients against the access policies. Depending on their functionality, these services could also provide context information of the user to other users.

Users need to be able to specify policies that provide appropriate levels of privacy to protect the personal information their mobile devices are collecting including the inferences that can be drawn from the information. Furthermore, the policies need to be expressive, flexible, and allow for context-dependent release of information. Semantic technologies represent a key building block for supporting expressive context policy modeling, reasoning and adaptation (Weitzner et al. 2004). We use semantic Web technologies to model a higher-level notion of context and to specify high-level, declarative policies that describe users' information sharing preferences under given contextual situations.

Semantic Context Model

We consider elements of context that are particularly related to mobile computing, and which can be exploited in many applications including personal agents that proactively control activities on the phone such as being turned off during a meeting, downloading relevant information, and enforcing relevant privacy policies.

In current practice, the user's location is captured at the level of position, i.e., geospatial (latitude-longitude) coordinates. A particular position can be mapped to a place or geographic entity, such as a region, political division, populated place, locality, and physical feature. Although position and geographic place information are potentially valuable on their own, from the standpoint of context, place is a more inclusive and higher-level notion: a location in conceptual terms such as "*at work,*" "*in a study group meeting,*" "*at lunch,*" "*in class*" –descriptions that combine a geographic place with the users activities and surroundings, and the presence and activities of nearby people and devices.

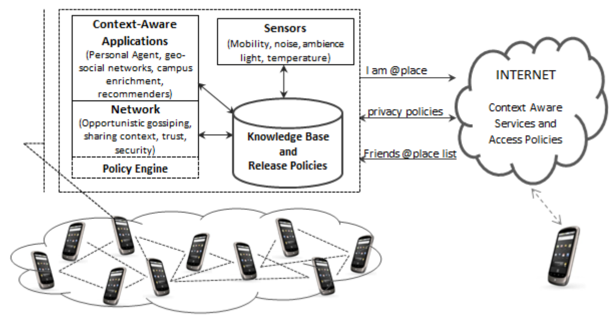

Figure 1: Interaction among entities in a collaborative information sharing, context-aware system.

We built 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 inferencing mechanisms to develop 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 2 shows the core classes in the ontology 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 due to spatial containment and such relationship ($part_of$) is transitive. The mapping from $Positions$ to $GeoPlaces$ is many to one and the mapping from $Positions$ to $Places$ is many-to-many (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). To support the mapping of positions to places, we rely crucially on activities. This approach reflects our pragmatic philosophy that the meaning of a place depends mainly upon 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 users 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

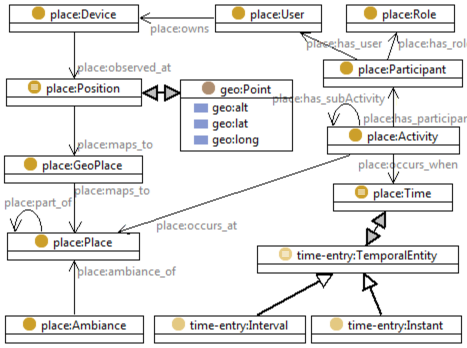

Figure 2: *The Place ontology* models the concept of *place* in terms of activities that occur there.

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 users activity (if available), we assert the corresponding knowledge 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 3). 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 3)
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 4) to infer knowledge about a user's geoplace based on the places his/her associated device is observed.

Activity and Place Inference

The system uses machine learning algorithms to recognize activity (e.g., "sleeping", "walking", "sitting", "cooking"), coarse-grained geographic place, and conceptual place (e.g., "at work", "at home") at different levels of granularity. 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). The general architecture (see section 2 —General Architecture), however, is planned for collaborative context inference. For high-level, general activities, we obtained a high accuracy but with more coarse-grained ones the accuracy drops. We expect this to improve as we incorporate more complex models that allow for collaborative context inference.

```
@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 3: An excerpt of the assertions made to the KB (left) in Turtle syntax and an example of a Jena rule used to integrate knowledge from GeoNames (right)

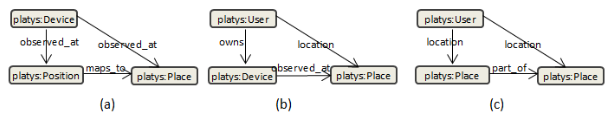

Figure 4: Property chain axioms to assert knowledge about a user's location. a) Device is *observed at* the place whose location *maps to*; b) User's *location* is the place where his/her associated device has been *observed at*; c) Generalization of user *location* based on spatial containment (*part of*).

The Dataset

We collected data for five users over the course of two weeks using Android smartphones and an interactive data collection program (Figure 5. Three users are students and two are faculty in the UMBC Computer Science Department. The information we are collecting 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 location (which includes coarse-grained geographic place, and conceptual 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 currently not specified since we preprocess the data for each particular learning task we try and we know the hierarchy. Figure 6 shows

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

Table 1: Recognition accuracy for high-level, general activities using Decision Trees.

a user's activity pattern on a Sunday, a weekday, and during a week. Such visualizations help us understand the range of activities performed by a user and could also be used to help people plan their activities.

Experiments and Results

Different machine learning algorithms were used to classify the place and activity of the user given the particular values for location, ambiance attributes, wifi and bluetooth scannings, calendar data, etc.; in general, all the information obtained 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. High accuracy is observed for high-level, general activities (see Table 1). 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.

Table 2 shows the accuracy for a mid-level detailed activity recognition task for a particular user and ten everyday activities using different classifiers and with 10 cross fold validation and 66% split validation testing options. Accuracy levels are comparable to those reported on (Lu et al. 2010) and (Bao and Intille 2004), although the focus on those works is mainly recognition of a limited subset of nine or fewer 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 Bayesian assumptions of conditional independence between features and normal distribution of feature values may contribute to the weaker performance of the approach. Furthermore, Bayesian algorithms usually require more training data to accurately model feature value distributions.

We are currently studying to what extent activity and place recognition can be generalized across users by training the classifier with one person's data and testing it with other's. However, this requires more data than we currently have which we are in the process of collecting.

Privacy Reasoning and Enforcement

In our prototype system, the context is shared among devices by means of queries sent directly between them or through a server. The integration occurs at each device and is currently a simple operation where the results are added to the knowledge base. Our prototype system has three privacy enforcement points. Users specify privacy policies that regulate the disclosure of (i) sensor information to the server (e.g., GPS information), (ii) inferred context information to the server (e.g., activity information), and (iii) inferred context information to other users.

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 2: Accuracy of different algorithms for activity recognition of a particular user and ten everyday activities.

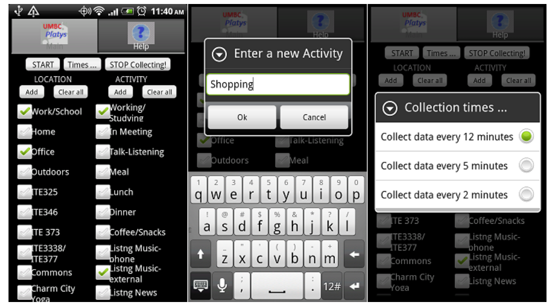

Figure 5: Our in-house Context Data Collection Program

A central part of our policies is the definition of groups. A user defines groups of contacts such as friends and family which are stored in the KB too. The user also specifies context dependent privacy policies and sharing preferences for each group. Privacy policies are expressed as logic rules over the KB. Our focus is currently 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 mioght be obfuscated in order to protect user privacy. The implementation makes use of Jena semantic web framework (Carroll et al. 2004). Privacy rules are defined as Jena rules and Jena reasoning engine is used to perform the reasoning. For the devices, we use AndroJena (Jena Android porting) (Lorecarra 2009) which is a porting of Jena to the Android platform.

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 like 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

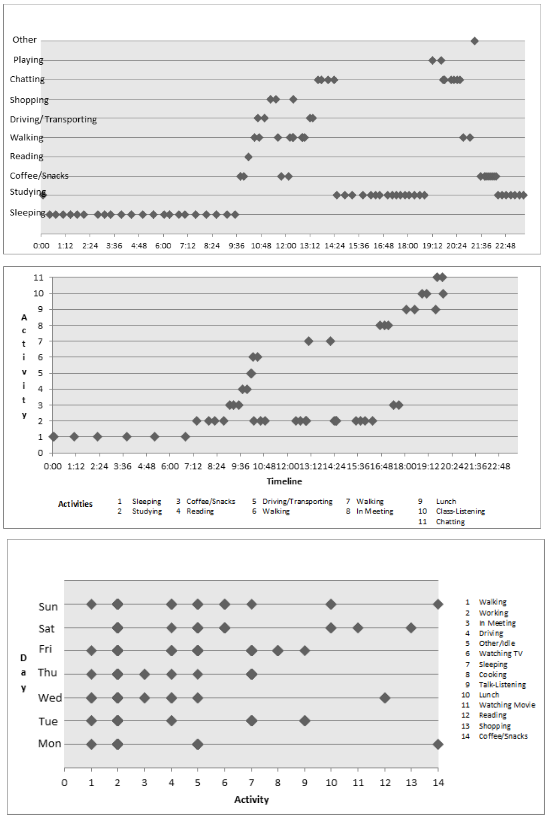

Figure 6: These three graphs show data collected about a user's activites on a Sunday, a weekday and during a week respectively.

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 7 shows the representation of the first rule as a Jena rule (left) and the results on a test screen we provide to observe the results of the reasoning engine (right).

Policies for Obfuscating Shared Information

In a context-aware system 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 context 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 simple sumbsumption hierarchies over location and activity entities (e.g., *City* is subclass of *State* which is subclass of *Country*).

```
@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")
]
```

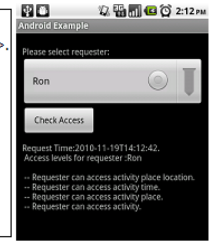

Figure 7: Left: Jena rule for expressing the policy *"share detailed contextual information with family members all the time."* Right: Android device screen with reasoning results. It shows access levels for requester *"Ron"* who is a member of the group *family*.

The generalization models for location and activity take advantage of the hierarchical nature of location information, which is evident by the part-of or contained relations between location entities. The policies specify at which level they which their location information to be revealed. When a query for location 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.

Related Work

The works in [1][2][3][4] present probabilistic approaches to recognize user activity based on observations from different types of sensors. We refer the reader to [5] for an extensive survey of various context-aware systems seen in literature. Most of those works, rely on the use of special equipment such as sensors embedded in the user's body (i.e., accelerometers) and in objects (RFID sensors). Specific applications in mobile phone devices are focused mainly on the localization and include location-aware tour guides [6] [7], which provided tourists with information depending on their location, and universities mobile applications [8][9], which provide location-based services aimed at enhancing everyday campus life at a university.

There has been several works [10] to [20] that deal with declarative formalisms and semantic Web technologies for security, privacy policies representation, reasoning, and enforcement, role-based access control to control access to resources, proof checking, proof-carrying authorization, and related areas. The importance of adopting a high level of abstraction for representing the different components in policies (such as subjects, actions, and elements of context) has been widely recognized by al. these works. Three well-known policy frameworks are Rein [16], Kaos [18], and Ponder [20]. A comparison of these can be found in [17].

The work in [12] is probably the most closely related work to ours. They present a system that makes use of semantic technologies to enable dynamic adaptation of policies depending on context changes. In particular, the paper shows how ontologies and logic programming rules can be used to leverage policy adaptation. One difference with our work is that they do not have a model for a general notion of context. They present a model to represent context elements for the particular case of work meetings.

Furthermore, there is no collaborative sharing of knowledge about context, one of our main goals which has been contemplated in our design and is currenly being addressed. Finally, adaptation is done through specification of alternative situations and actions under those situations. In our case, context is inferred by observing sensed data on smart phones, as well as integrating context information from nearby devices. [21] presents a model to capture, represent and apply trust policies of an agent in the scenario of Semantic Web knowledge bases, while preserving real-world semantics of trust. In [22], the authors present an approach that extended traditional role-based access control to include the notion of an environment role. The focus was on solving the problem of securing context-aware applications in a ubiquitous computing environment.

Conclusion

We presented an architecture for collaborative context aware systems where devices share and integrate knowledge about their context. We addressed collective context inference and privacy issues related to it. Our current status is 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.

The inferred context knowledge is stored in a local knowledge base on the device and can also be sent to context-aware services located on the Internet. Context-aware applications, network components, and sensors may use this context knowledge to enhance their functionality. We plan to create a few simple applications for Android devices that will exploit this knowledge.

We built on existing work in policy languages to address the need for providing users with privacy to protect the personal information their mobile devices are collecting. Our release policies ensure context dependent release of information in accordance to the user preferences. Additionaly, we extended existing work by introducing the notion of policies for obfuscating shared information. Our policies are mainly centered on the concept of groups. We are extending our prototype implementation to allow for a more flexible way to specify the subjects (instead of fixed groups). We have used Jena rules in combination with OWL and SPARQL to achieve our goals. Our current implementation has some ad-hoc mechanisms to make it possible to integrate the ontology with rules and queries to open KBs. However, as we look to generalize the process, we raise the question whether a new policy language is needed to make policy declaration and enforcement integration more seamlessly. Further, we want to be able to express richer policies at the triple level. It seems that a mix of rich pattern matching such as SPARQL and rules, together with first order semantics of existing approaches is needed, which we are studying. Finally, we plan to address the issue of privacy beyond protecting only the sharing of information and including the inferences that can be drawn from the information that is shared.

Acknowledgments

Support for this work was provided by NSF grant 0910838, MURI award FA9550-08-1-0265 from the Air Force Office of Scientific Research, and a gift from Qualcomm. We thank Amey Sane and Nikhil Puranik for their work on data visualization.

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.

Bao, L., and Intille, S. S. 2004. Activity recognition from user-annotated acceleration data. In Ferscha, A., and Mattern, F., eds., *Pervasive*, volume 3001 of *Lecture Notes in Computer Science*, 1–17. Springer.

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, H., and Joshi, A. 2003. An ontology for context-aware pervasive computing environments. *Knowledge Engineering Review* 18(3):197–207.

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

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

Kagal, L.; Finin, T.; and Joshi, A. 2003. A Policy Based Approach to Security for the Semantic Web. In *Proc. 2nd Int. Semantic Web Conf.*, 402–418.

Lorecarra. 2009. Androjena : Jena android porting.

Lu, H.; Yang, J.; Liu, Z.; Lane, N. D.; Choudhury, T.; and Campbell, A. T. 2010. The jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, 71–84. New York, NY, USA: ACM.

Weitzner, D. J.; Hendler, J.; Berners-lee, T.; and Connolly, D. 2004. Creating a policy-aware web: Discretionary, rule-based access for the world wide web. In *Web and Information Security. IOS*. Press.

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