APPROVAL SHEET

Title of Thesis: Privacy Preservation in Context-Aware Systems

Name of Candidate: Pramod Jagtap
M.S. in Computer Science,
2011

Thesis and Abstract Approved: _____

Dr. Anupam Joshi
Professor
Department of Computer Science and
Electrical Engineering

Date Approved: _____

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2011		**00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Privacy Preservation in Context-Aware Systems	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Maryland,Department of Electrical Engineering,College Park ,MD,20742**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Recent years have seen a confluence of two major trends ? the increase of mobile devices such as smart phones as the primary access point to networked information and the rise of social media platforms that connect people. Their convergence supports the emergence of a new class of context-aware geosocial networking applications. While existing systems focus mostly on location, our work centers on models for representing and reasoning about a more inclusive and higher-level notion of context, including the user?s location and surroundings, the presence of other people and devices, feeds from social networking systems they use, and the inferred activities in which they are engaged. A key element of our work is the use of collaborative information sharing where devices share and integrate knowledge about their context. This introduces the need for privacy and security mechanisms. We present a framework to provide users with appropriate levels of privacy to protect the personal information their mobile devices are collecting including the inferences that can be drawn from the information. We use Semantic Web technologies to specify high-level, declarative policies that describe user?s information sharing preferences. We have built a prototype system that aggregates information from a variety of sensors on the phone, online sources, and sources internal to the campus intranet, and infers the dynamic user context. We show how our policy framework can be effectively used to devise better privacy control mechanisms to control information flow between users in such dynamic mobile systems.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**67**	

ABSTRACT

Title of Thesis: Privacy Preservation in Context-Aware Systems

Pramod Jagtap, Masters in Computer Science, 2011

Thesis directed by: Dr. Anupam Joshi, Professor
Department of Computer Science and
Electrical Engineering

Recent years have seen a confluence of two major trends – the increase of mobile devices such as smart phones as the primary access point to networked information and the rise of social media platforms that connect people. Their convergence supports the emergence of a new class of context-aware geosocial networking applications. While existing systems focus mostly on location, our work centers on models for representing and reasoning about a more inclusive and higher-level notion of context, including the user's location and surroundings, the presence of other people and devices, feeds from social networking systems they use, and the inferred activities in which they are engaged. A key element of our work is the use of collaborative information sharing where devices share and integrate knowledge about their context. This introduces the need for privacy and security mechanisms. We present a framework to provide users with appropriate levels of privacy to protect the personal information their mobile devices are collecting including the inferences that can be drawn from the information. We use Semantic Web technologies to specify high-level, declarative policies that describe user's information sharing preferences. We have built a prototype system that aggregates information from a variety of sensors on the phone, online sources, and sources internal to the campus intranet, and infers the dynamic user context. We show how our policy framework can be effectively used to devise better privacy control mechanisms to control information flow between users in such dynamic mobile systems.

Privacy Preservation in Context-Aware Systems

by

Pramod Jagtap

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
MS
2011

Dedicated to Ebiquity Lab

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Dr. Anupam Joshi. I thank him for the constant support and guidance, and for his continued belief in me throughout this thesis work. He has always given me full support and allowed me to explore new topics and research problems that drove my interests. I am thankful for his words of advice and many skills I have gained by working with him.

Thanks to Dr. Tim Finin and Dr. Laura Zavala for all the valuable suggestions throughout my work at Ebiquity lab. I would like to thank all Dr. Tim Finin, Dr. Yelena Yesha and Dr. Laura Zavala for graciously agreeing to be on my thesis committee. They have always made themselves available and accessible and I thank them for their time, suggestions and important advice.

All the Ebiquity group members have been extremely supportive in building an atmosphere conducive to research. It has been a great joy working with them throughout the year.

I extend my sincere thanks to National Science Foundation (award 0910838) and the Air Force Office of Scientific Research (MURI Grant FA9550-08-0265).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Content sharing on social networking websites has dramatically increased over the last few years. Popular services such as Facebook, Twitter and MySpace allow millions of individuals to create online profiles and share personal information with a huge number of friends. The increasing availability of extended geo-location technologies such as cell tower localization on Internet services and Assisted Global Positioning System (A-GPS) on phone devices, has changed the way people interact with each other on the web. It has enriched the social networking experience with additional social dynamics that emerge from allowing users to interact relative to location and time. Location awareness is one important aspect of context-aware systems. However, context encompasses more than just the user's location, because other things of interest are also mobile and changing (Schilit, Adams, & Want 1994). Other important aspects include the ambiance, resources and people nearby, and the activities in which they are engaged. The rise of online social networking systems along with recent improvements in mobile technology, smartphones, and sensor networks presents a unique opportunity for context-aware systems.

A very important but often overlooked issue in most social networking systems is that of privacy. The existing research addressing privacy issues (Acquisti & Gross 2006), (Dwyer, Hiltz, & Passerini 2007), (Gross & Acquisti 2005), (Jones & Soltren 2005), brings

out various concerns and emphasizes the need of strong privacy control mechanisms. Furthermore, the recent emergence of context-aware geosocial networking services demand more robust access control mechanisms. These systems face similar security threats as distributed and mobile applications but privacy and trust aspects are more prominent due to the sensitive nature of context information. Users need to protect the personal information that their mobile devices are collecting through automated collection of sensor data, as well as the inferences that are drawn from that information. Users can be understandably sensitive about how the sensor data is captured and used, especially if it is used to reveal a user's location, speech or sensitive images. In addition to such security considerations, people may simply be uncomfortable with others knowing their location, or even with their location being sensed in the first place. Mobile applications such as the Audio Loop (Hayes *et al.* 2004), which continuously record raw audio, also raise concerns and introduce issues about how (or even whether) to obtain consent to be recorded from others whose data might be captured by the user's device (Iachello *et al.* 2006). Such concerns could affect the adoption and use of devices that embed sensing and introduce problems into social relationships. Although there are existing approaches that can help with these problems (e.g., cryptography, privacy-preserving data mining), they are often insufficient (Kapadia, Kotz, & Triandopoulos 2009).

Ideally a context-aware system consists of heterogeneous and dynamic sensors which causes the constant changes in both owner's and requester's contextual information. This environment calls for better access controls with finer control over the context data to preserve user privacy. There is a need for privacy control mechanisms that consider the dynamic changes in user context relative to the location and time. The user needs to be in control of the release of her personal information at different levels of granularity, from raw sensed data to high level inferred context information. A context-aware infrastructure should provide the end user with a (logically) central place of privacy control and trust

management, contrary to point solutions within different, possibly not trusted, applications (van Sinderen 2006). Thus, users should be able to define their privacy policies and the context-aware system should be able to protect their information from illegal access as per privacy policies regardless of the application.

For instance, consider a healthcare context-aware system where sensor-enabled mobile phones can be used to collect in situ sensor data and context data such as patient's and caretakers' personal information, current location and current activity. In this case the user can specify privacy policies like *"allow Dr. Nash detailed information at all time"* and *"allow access to caretaker's location only in case of emergency"*. Consider another scenario of university campus; a student user may be willing to let her teachers see where she is between 9:00am and 6:00pm on weekdays but not over the weekend. Further, she may not be willing to let her teachers know about her sleeping activity during the daytime. Additionally, a user may want to control the granularity or accuracy of the answer, depending on current context of her and the requester. For instance, a user might want to share the room-level location to some people and city-level location to others. She might want to share the exact room number to anyone who is in same building as she. On the similar lines, a user may not want to disclose her location if she is at some sensitive place like a nightclub. To incorporate all such privacy policies, the system needs to generalize the contextual data and provide an option to specify policies over different granularity levels of the context data.

Privacy control mechanisms should be flexible enough to capture contextual information about their users subject to semantically rich privacy constraints. Besides flexibility on the level of granularity of the information and situation under which information can be shared, the incorporation of incentives can add even more richness to the policies. Consider for example, in the university campus scenario, that a particular restaurant offers discounts for groups of five or more students on a particular day. A student and a few of her acquain-

tances happen to be looking for lunch around that restaurant at the same time. The users might be more interested in sharing their locations under situations where they might get rewarded for doing so.

Overall, we are motivated by the need of privacy control models to control the information flow in collaborative context-aware geo-social networking applications based on the context of both owner and requester. None of the existing models allow users to specify the privacy preferences based on this information in a subtle way. Therefore, in this thesis we present a policy based framework to constrain the information flow based on the contextual information along with profile information. It can be extended and incorporated in existing social networks including location based mobile social networks. We validate our architecture in an on-campus context-aware prototype system that aggregates information from a variety of sensors on the phone, online sources, and sources internal to the campus intranet, and infers the dynamic user context. We show how our policy framework can be effectively used to devise better privacy control mechanisms to control information flow between users in such dynamic mobile systems.

Chapter 2

BACKGROUND AND RELATED WORK

Context-aware systems have been studied for a long time. The focus has been mainly on the location and activity inference. The Active Badge Location system (Want *et al.* 1992) used infrared technology to find the location of a user so that calls can be forwarded to phones nearby. The context-aware electronic tourist guide (Cheverst *et al.* 2000) contributed by developing location-aware tour guides which provided tourists with information depending on their location. Recently research about privacy controls in these systems has received the significant attention. AnonySense (Shin *et al.* 2010), a privacy-aware architecture for collaborative pervasive applications that use mobile sensing. Mobile sensor data is anonymized before its use by any of the applications. Project Aware Home (Kidd *et al.* 1999) captures, processes and stores data (collected by sensors) about home residents and their activities. It uses access control mechanism based on Role-based Access Control (RBAC) by defining environment roles similar to subject roles of RBAC and it is used to capture security-relevant aspects of the environment in which an application executes. Context Privacy Service (CoPS) (Sacramento, Endler, & Nascimento 2005) describes the design and implementation of a privacy service which control how, when and to whom you could disclose a user's context information. Using the end-user survey and results of other research groups, it has identified requirements for flexible and efficient pri-

vacy service. This system is most closely related to our work. However, it doesn't handle context-dependent privacy policies, which can be specified by users on dynamic context data. Overall, most privacy preserving works focus on location related aspects of context and deal with mechanisms to control access to such information.

The understanding of few background concepts such as Policies and Semantic Webs, Geo-social networking systems should make this thesis more easier to understand.

2.1 Policies and the Semantic Web

The Semantic Web refers to both a vision and a set of technologies. The vision was first articulated by Tim Berners-Lee as an extension to the existing web in which knowledge and data could be published in a form easy for computers to understand and reason with. Doing so would support more sophisticated software systems that share knowledge, information and data on the Web just as people do by publishing text and multimedia. Under the stewardship of the W3C, a set of languages, protocols and technologies have been developed to partially realize this vision, to enable exploration and experimentation and to support the evolution of the concepts and technology.

The current set of W3C standards are based on RDF (Lassila & Swick 1998), a language that provides a basic capability of specifying graphs with a simple interpretation as a "semantic network" and serializing them in XML and other popular Web systems (e.g., JSON). Since it is a graph-based representation, RDF data are often reduced to a set of 'triples' where each represents an edge in the graph ('Person32 hasMother Person45') or alternatively, a binary predication (e.g., 'hasMother(Person32,Person45)'. The Web Ontology Language OWL (Bechhofer *et al.*) is a family of knowledge representation languages based on Description Logic (Baader *et al.* 2003) with a representation in RDF. 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. The axioms can be realized as simple assertions (e.g., 'Woman is a sub-class of Person', 'hasMother is a property from Person to Woman', 'Woman and Man are disjoint') and also as simple rules.

The use of OWL to define policies has several important advantages that become critical in distributed environments involving coordination across multiple organizations. First, most policy languages define constraints over classes of targets, objects, actions and other constraints (e.g., time or location). A substantial part of the development of a policy is often devoted to the precise specification of these classes, e.g., the definition of what counts as a 'student' or a 'entertainment activity'. This is especially important if the policy is shared between multiple organizations that must adhere to or enforce the policy even though they have their own native schemas or data models for the domain in question. Second, OWL is based on description logic, a well understood subset of logic for which powerful and efficient reasoning systems are available. By constraining our use of OWL to the right subset, we can exploit existing OWL reasoners. A third advantage is that OWL's grounding in logic facilitates the translation of policies expressed in OWL to other formalisms, either for analysis or for execution. Finally, OWL is designed of and for the Web, making sharing policies and the ontologies they use both natural and easy.

There has been a lot of work done to develop access control frameworks (Moses 2005), (**?**), (Jajodia *et al.* 1997). Rein (Rei and N3) (Kagal & Berners-lee 2005) is a distributed framework for describing and reasoning over policies in the Semantic Web. It supports N3 rules (Berners-Lee & Connolly 2008), (Berners-Lee *et al.* 2005) for representing interconnections between policies and resources and uses the CWM forward-chaining reasoning engine (Berners-Lee), to provide distributed reasoning capability over policy networks. AIR (Kagal, Hanson, & Weitzner 2008) is a policy language that provides automated justification support by tracking dependencies during the reasoning process. It uses Truth

Maintenance System (Doyle 1978) to track dependencies. Policies and data are represented in Turtle (Beckett 2007), whereas the reasoning engine is a production rule system (Waterman & Hayes-Roth 1978) with additional features for improved reasoning efficiency such as goal direction. Rei and AIR consider rules defined over attributes of classes in the domain including users, resources, and the context.

2.2 Geo-social networking systems

A mobile social network identifies and tracks the geo-spatial locations of a user and other people in her social network and typically can display them on a map interface. So not only can a user share information, media and updates with her friends, but one can also find out exactly where everyone is. Facebook has recently launched a location-based feature "Places Check-in". It lets you check in on the place you are currently at and when you check in, it allows you to tag friends who are with you, just as you can tag a friend in a status update or photo. You can post an update along with your check-in to tell people more about what you are doing. The tricky part here is if you have set your privacy control to "Everyone", other Facebook users will know that you and your friend are in specific location at the current time. Brightkite, one of the popular mobile geo-social networking applications, provides two different modes. In public mode, information shared with everyone with full accuracy, and in private mode, it allows users to share information with people at three different trust levels (Trusted friends, friends and everyone else) and with three different levels of visibility (hidden, city, and exact). FourSquare, another popular service, shows a user's current geo-location to her friends, even when the application is not open. Google Latitude also has similar privacy settings wherein a user can hide her location, show exact location, or share the city where she is in with all the invited friends on Google Latitude. Privacy is an important issue with these services, they all have some opt-

in and information protection options, but they don't provide strong control while allowing a user to share her geo-location or current activity information.

Chapter 3

SYSTEM ARCHITECTURE

The proposed system architecture is shown in the Figure 3.1. The major components of this system are client devices, server side modules and the Internet services that provide social media. The client devices are location aware smartphones. Today's smartphones are programmable and come with a large set of cheap powerful embedded sensors, such as a camera, GPS, accelerometer, digital compass, gyroscope, microphone, and many more. These sensors are enabling the emergence of personal, group and community scale sensing applications. These client devices as well as the server side modules contain a user profiles repository, a privacy control module and content preferences. The server side also contains a content aggregator, a learn and share module and a privacy control module. The content aggregator combines social media like event updates, photos, and videos from Internet services like YouTube, Flickr, Facebook or university information portals. The learn and share module infers the user's dynamic context using sensor data collected by a variety of sensors on the phone, the information from the content aggregator and online sources such as user's calendar. The inferred context is shared with corresponding client device so that the device along with server can handle further context sharing queries from other clients. The requester queries are passed through the privacy control module to constrain the information flow and hence to protect the user privacy. The privacy control module

provides the access control mechanisms and aids in controlling the information flow within system. On the client device, it enables privacy sensitive and resource sensitive reasoning over sensed data along with privacy enforcement between peer devices sharing contextual information. The interaction between various components of our system can be described as follows:

- The system user has a client device to collect the sensor data periodically. This data is passed to the learn and share module residing on the server through the privacy control module of client device. The privacy control module decides the specific sensor data that can be shared with the server based on user-specified privacy policies.

- The learn and share module infers the user context using sensor data and information from the content aggregator and other online sources. This context consists of current location, activity and additional surrounding information like nearby people. This inferred knowledge is passed to the corresponding client device so that it can handle context access queries from other clients.

- The client device can request contextual information to another client device or server. The access requests are passed through the privacy control module of other client device or server, which in turn decides whether to allow or deny the access. If the requester is granted access then the system constructs the response compliant with user's privacy preferences.

- Figure 3.1 shows the three different ways in which information can be shared in our system, namely: (i) context information sharing between the client devices, (ii) sensor data sharing between a client device and the server, and (iii) context information sharing between a client device and the server. The information sharing is controlled by the privacy control module in order to preserve user privacy.

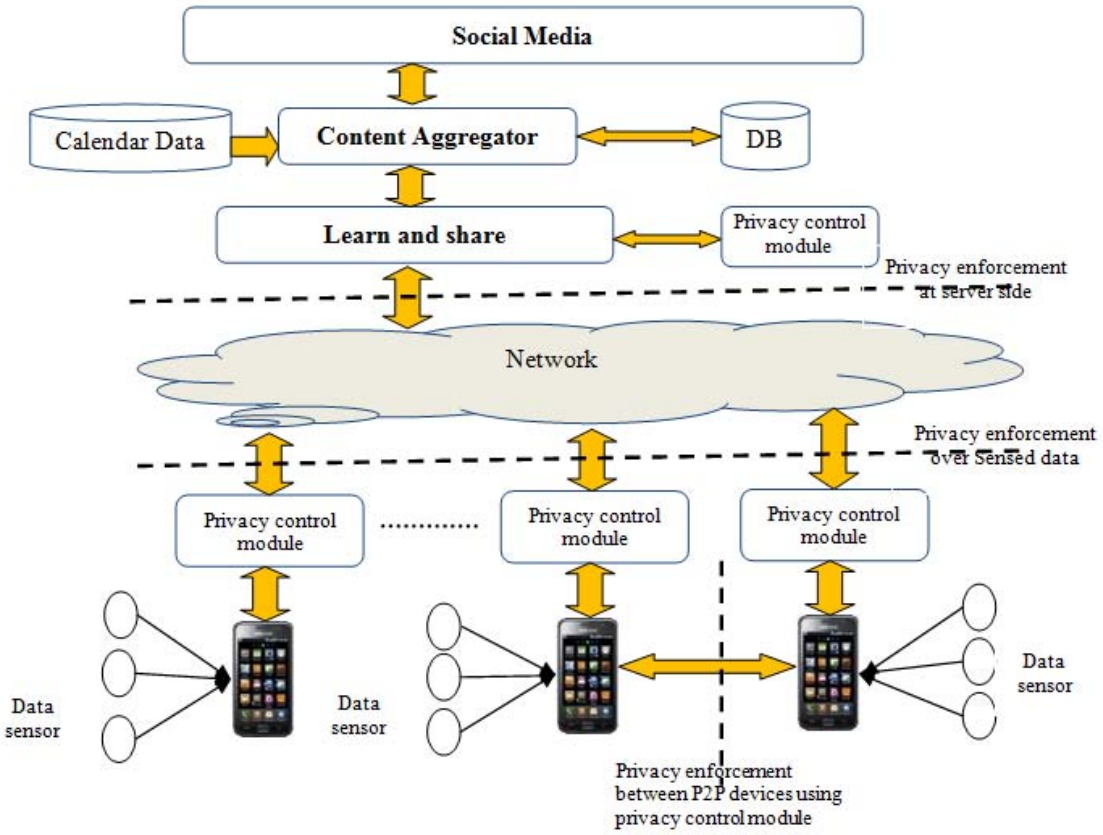

FIG. 3.1. The architectural view of the system

We will focus our discussion on our privacy mechanisms and the relevant system components which have most direct influence on the information flow in the system.

3.1 Privacy Related Components

The privacy control module aims to protect user privacy by performing reasoning over her context. It deals with the resource to be protected, the owner of a resource and the requester who wants to access it. It has access to owner's profile information and the group information along with specified privacy policies. It enforces user's privacy policies using static information about the user as well as dynamic information observed and inferred

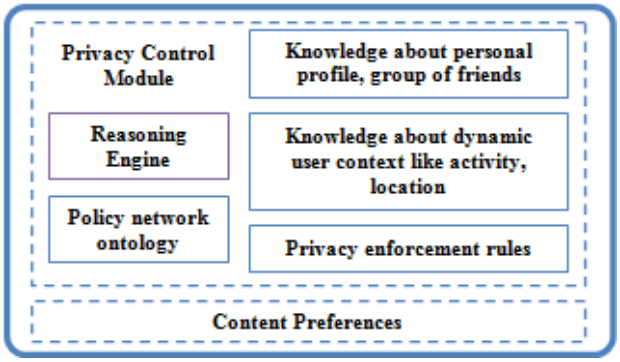

FIG. 3.2. Privacy control module

from her context. As shown in the Figure 3.2, it consists of, (i) a set of ontologies for describing policies and access requests, (ii) the knowledge about the owner, (ii) the privacy preferences, and (iv) a reasoning engine that accepts requests and performs the reasoning.

3.1.1 Context ontology

The context-aware systems raise the need of models for representing and reasoning about a more inclusive and higher-level notion of the context. Our context model ontology captures the user location and surroundings, the presence of other people and devices, and the inferred activities in which they are engaged. We adopt description logics (DL), specifically OWL (Web Ontology Language), and associated inferencing mechanisms to develop a model of context and policies. In the ontology model, the actions are in general lower level tasks and have no associated role. The activities are introduced as means to abstract multiple actions and further, to associate roles to the sets of actions. Places can be defined in terms of the activities that occur there. Ambiance includes concepts describing the environment of the principal (e.g., noise level, ambiance light, and temperature). The context ontology as shown in Figure 3.3, captures the semantic notion of context in a mobile context-aware system. Using the ontology, each device contains a declarative knowledge

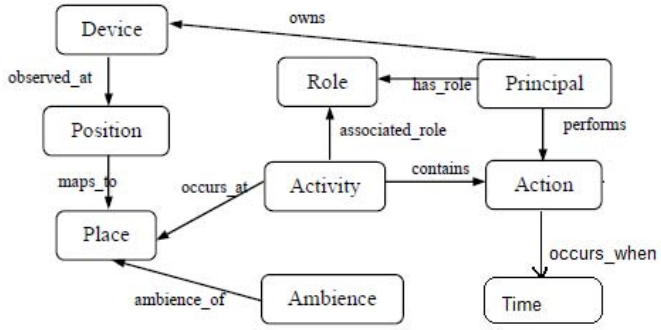

FIG. 3.3. The context ontology models the key concepts of context.

base with semantically rich information about user's information, activities, inferences, and further contextual information. The knowledge base aligns with the context ontology which defines the key context concepts used for making access control decisions. The ontology supports the generalization of context infomation by having hierarchical models for different aspects of context viz. activity and location. It helps the user to have finer control over her contextual information and hence to share information on different levels of granularity. The following section describes the location generalization and activity generalization in detail.

Location Generalization Usually the location information is sensitive and hence it should be shared with legitimate set of people as decided by user. It is achieved by allowing users to restrict the information sharing by specifying a set of privacy policies over this information. E.g. privacy policy such as "*Share my location with teachers on weekdays from 9am-5pm*" allows a group of people defined by the user as "teachers" to access user location. In this case "teachers" can access user's GPS coordinates on weekdays between 9am to 5pm. This approach has its own limitations as it doesn't allow sharing on different granularity levels of the location. In many cases the user might be interested to share the location but not in terms of GPS coordinates. E.g. the user can have privacy policy like

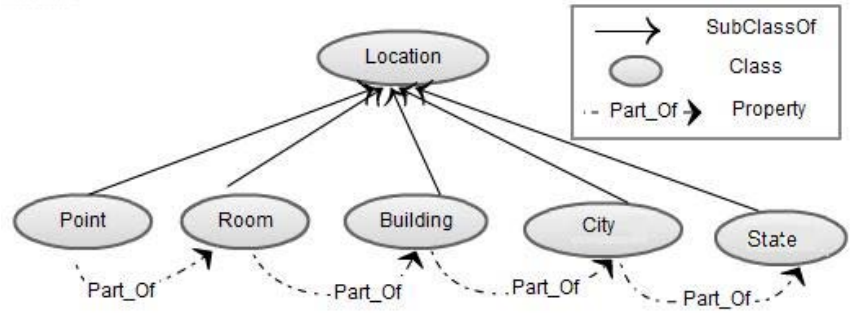

FIG. 3.4. The location hierarchical model.

"*Share my building-wide location with teachers on weekdays from 9am-5pm*" which allows location sharing but at the same time it doesn't reveal the exact location. The system will share the building names with "teachers" rather than exact GPS position of user. This way location generalization can be effectively used to protect user privacy.

In order to support the location generalization, our ontology uses hierarchical model for location as shown in Figure 3.4. The Location is a super class of Point, Room, Building, City and State classes. The Point class is used for denoting the GPS coordinates whereas Room and other subclasses are used to denote different levels of abstractions for the location. The transitive "Part_Of" property creates a location hierarchy based on some simple axioms like "Room is a part of Building". The reasoning engine will use this ontology to infer the different relations existing between instances of these subclasses.

Activity Generalization Along the similar lines of location generalization, we present the activity generalization for allowing users to share different precision levels of current activity to different set of requesters. Consider a policy like "*Share my activity with friends on weekends*"; it will share user's current activity to the people belonging to a "Friend" group. In many cases, the user is willing to share more generalized activity

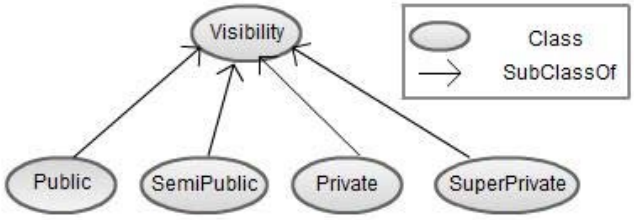

FIG. 3.5. The visibility options.

rather than precise one. E.g. if a user is attending a confidential "project meeting" then she might want to share it in a more generalized way as "working" or simply as a "meeting". In another scenario, if the user is out with someone on a "Date", then she might want to share it as a "Social Meeting". In both cases, the user clearly needs to obfuscate certain pieces of activity information to protect her context information. In other words, the user needs to differentiate between the set of activities by attaching a confidentiality parameter e.g. visibility option. The visibility option specifies the sensitivity level of activity from the user perspective. Our ontology supports different visibility options as shown in Figure 3.5. The Public option implies that the corresponding activity is least sensitive whereas Super-Private option indicates that the activity is at most sensitive. The SemiPublic and Private are listed in increasing order of sensitivity. These visibility options can be used to share more generalized/less sensitive/public activities instead of specific/sensitive/private ones.

Our ontology supports the activity generalization by using a hierarchical model of activities as shown in Figure 3.6. The Activity is a super class of all the activities whereas Working, Meeting and Studying activities has few subclasses. Each of these activities has a property called "has_visibility" which is pivotal for the generalization of activities. This property is used to associate a visibility option with the activity.

We have integrated these hierarchical models as part of our context ontology. With

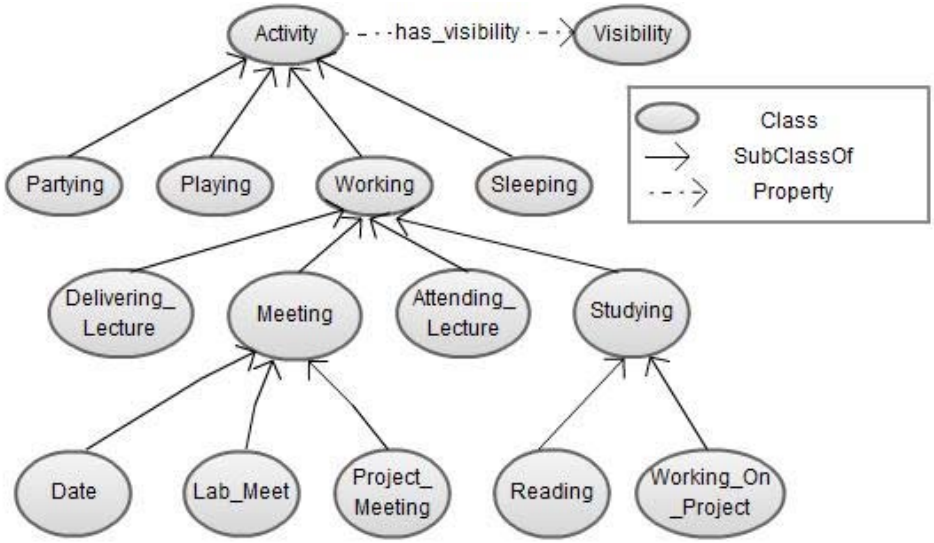

FIG. 3.6. The activity hierarchy model.

this ontology, our system effectively allows users to specify complex privacy policies with the notion of generalization.

3.1.2 Knowledge about the user

The user can create her personal profile with information like name, email address, hobbies and interests and, can manage different groups of her friends. Apart from that, the system has dynamic knowledge information about user including current activity and her recent location. Our context ontology defines the entities required to represent a user information in addition to the FOAF vocabulary. This knowledge is represented using N3 in our system. The context sensitive information such as a user's current location can be edited by the user and is accepted by the system with consent. All the attributes in a user's personal profile as well as data sensed by mobile devices are considered as resources to be protected. The profile and context information of some user "Alice" is shown in Table 3.1. The "platys" is our context ontology, "ex" is a namespace of user information file and foaf

Table 3.1. The user information.

```
ex:Alice a foaf:Person ;
    foaf:name “Alice” ;
    ex:systemUser “true” ;
    platys:has_role platys:Student .
platys:Professor_Meeting a platys:Activity ;
platys:is_performed_by ex:Alice ;
    platys:has_participant ex:Alice, ex:John ;
    platys:occurs_at platys:Class_LH1 ;
    platys:occurs_when “2010-11-19T14:12:42”.
platys:Class_LH1 a platys:Place ;
    platys:has_location “39.253525, -76.710706”.
platys:GPS a platys:Point ;
    platys:part_of platys:ITE_325 .
platys:ITE_325 a platys:Room ;
    platys:part_of platys:ITE .
platys:ITE a platys:Building ;
    platys:part_of platys:Baltimore .
platys:Baltimore a platys:City ;
    platys:part_of platys:Maryland .
platys:Maryland a platys:State .
```

represents the FOAF vocabulary. The information clearly specifies the user is a Person with name “Alice” and owner of client device. The current activity is Professor_Meeting and it occurs at some place Class_LH1. It has associated generalized location information which states that ITE_325 is a Room, ITE is a Building and so on. The Table 3.2 shows the snippet of user’s group information. It states that Harry is a person belonging to the Family group and Ron is a person belonging to the Friend group.

Table 3.2. Sample group information

```
ex:Harry a foaf:Person ;
    foaf:name “Harry” ;
    ex:memberOf ex:GroupFamily .
ex:Ron a foaf:Person ;
    foaf:name “Ron” ;
    ex:memberOf ex:GroupFriends .
ex:GroupFamily a foaf:Group ;
    foaf:name “Family” .
ex:GroupFriends a foaf:Group ;
    foaf:name “Friends” .
```

3.1.3 Privacy preferences

Privacy preferences are access control rules that describes how the user wants to share her information, with whom, and under what conditions. The user can disclose information with different accuracy levels; for instance, she may tell the exact building on the university campus to her close friends, but just the county or town she is in to others. The user may decide not to disclose her location to advertisers. She can manage different networks of friends, and assign variety of group level privacy preferences accordingly. For instance, a user can create a group of family members, a group of colleagues, or a group of teachers, and may define distinct privacy settings for each of them. Conditions can be defined based on the dynamic attributes such as context of the user or requester including current location, current activity or any other dynamic attribute. All the privacy preferences are represented as N3 rules in the system. Our system supports both user-level and system-level privacy rules. These rules have same representation but the latter overrides the former always. The user-level rules are specified by the user to protect her information whereas system-level

rules are defined by a system administrator for entire organization to conceal any sort of information leakage.

3.1.4 Reasoning Architecture

The reasoning engine handles the requester queries and performs reasoning for access control decisions. Our system uses the Jena Semantic Web framework(Carroll *et al.* 2004) for performing reasoning over the context data. Jena inference system allows the support of various inference engines or reasoners. These reasoners are used to infer additional facts from the existing knowledge base coupled with ontology and rules. In particular, Jena uses the generic rule reasoner which is included in Jena2 as a general purpose rule-based reasoner. It is used to implement both the RDFS and OWL reasoners. It needs at least a rule set to define its behavior. Its instance with a ruleset can be used like any of the other reasoners - that is it can be bound to a data model and used to answer queries to the resulting inference model. In our system, the reasoning engine uses the context ontology, context information of owner and requester, the owner's profile and group information along with privacy rules to generate an inference model. This inference model is used for generating response to the requester queries. This process is shown in the Figure 3.7 and summarized in the following steps:

1. Create the instance of OWL reasoner specialized for context ontology and then apply that to the user's static information to generate an inference model. This inference model consists of additional statements inferred from static knowledge and ontology. As the user information and ontology arent changed often, it is quite safe to save the model on external storage and reload it for subsequent queries rather than generating it each time.

2. The requester's contextual information is extracted from requester query and along

with user contextual information its added to inference model to generate a new model.

3. The system-level polices are executed against the inference model using an instance of generic rule reasoner. It is an optional feature and its used to enforce certain organization level policies. It will create a new model having SystemPermitted and SystemProhibited statements to enforce system policies over the users contextual information. If the user is a sole owner of client device then this step can be skipped. The detailed description of this feature is provided in the next section.

4. The user-specified privacy rules are executed against the inference model from step 3 to generate a new inference model having requester access levels.

5. The system will use the new model to decide what can be shared with requester and respond accordingly.

System Level Policies The context-aware systems are used by individuals to organization and from social-networking application to military domains. In case of military domains or organizations, the user may not be the sole owner of client device and there is a strong need of robust security mechanisms. It can be in the form of multi-level secure systems where the system-level policies must override the user-level policies. This highlights the need of system-level policies along with the user-specified policies. The system-level policies should be defined by the system-administrator to ensure that the sensitive resources are always protected from illegitimate access. Consider a system-level policy as "*Do not share the user's context if she is inside a military building BuildingXYZ.*" shown in Table 3.3 and a user-specified policy as "*Share my context with family members all the time.*" shown in Table 3.4. The system-level policy states that the user context won't be shared with anyone if she is inside BuildingXYZ whereas in latter policy user specifies to share

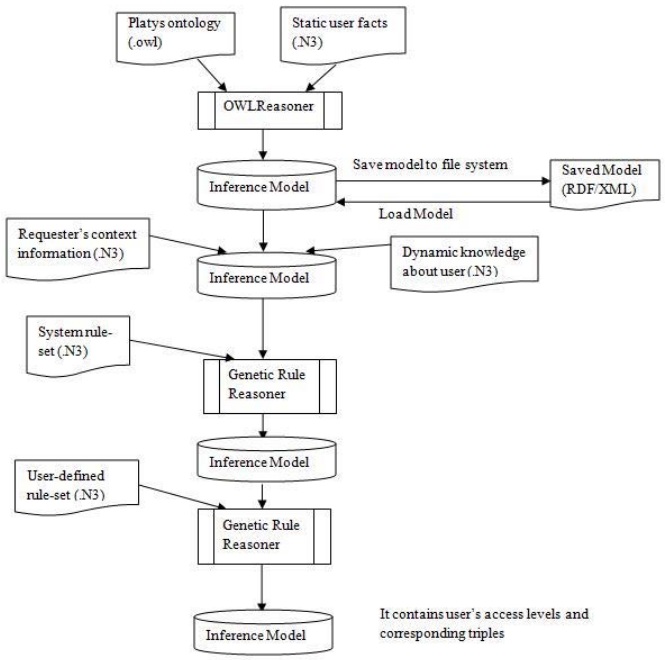

FIG. 3.7. The reasoning architecture.

her context with family members all the time. In this case the system-level policy should override user-specified policy and hence, if the user is inside BuildingXYZ then her context won't be shared to anyone including her family members. Consider the scenario where requester belongs to Family group and the user is currently inside BuildingXYZ. When requester queries the user context then the output inference model should have following statements resulted from reasoning: (i) (?requester ex:contextAccess ex:systemProhibited) and (ii) (?requester ex:contextAccess ex:userPermitted). The first statement is added to the model by system-level policy shown in the Table 3.3 whereas other is added by user-level policy shown in the Table 3.4. In this case, systemProhibited takes preference over userPermitted and hence, user context is not shared. The contextAccess property can have any value from the set {systemProhibited, systemPermitted, userPermitted, userProhibited}. The Table 3.5 shows the override preferences assumed by our system. Consider after

Table 3.3. Policy to not share user context if she is inside a BuildingXYZ

```
[Rule6:
    (?user ex:systemUser ?someValue)
    (?someActivity platys:is_performed_by ?user)
    (?someActivity platys:occurs_at ?userPlace)
    (?userPlace platys:has_location ?userLocation)
    (?userLocation platys:part_of ?userBuilding)
    (?userBuilding rdf:type platys:Building)
     equal(?userBuilding, platys:BuildingXYZ)
->
    (?requester ex:contextAccess ex:systemProhibited)
]
```

evaluation of the system rules and user rules, the contextAccess predicate has both SystemPermitted and UserProhibited values. In this case the context information is "Allowed" to share with the requester. Similarly if the contextAccess predicate has both UserPermitted and UserProhibited values then context information is "not allowed" to share with the corresponding requester.

3.2 Privacy Preservation

The user's personal information can be shared between a client device and the server or between two client devices. To constrain the information flow, privacy enforcement needs to be done on (i) client devices over sensed data, (ii) on peer client devices and (iii) at server side for contextual information. The subsequent sections will elaborate these points.

Table 3.4. Policy to share detailed contextual information with family members.

```
[AllowFamilyRule:
    (?requester a ex:requester)
    (?requester ex:memberOf ?groupFamily)
    (?groupFamily foaf:name “Family”)
->
    (?requester ex:contextAccess ex:userPermitted)
]
```

Table 3.5. Context access permission and preference table

Values	SystemPermitted	SystemProhibited	UserPermitted	UserProhibited
SystemPermitted	Allow	Deny	Allow	Allow
SystemProhibited	Deny	Deny	Deny	Deny
UserPermitted	Allow	Deny	Allow	Deny
UserProhibited	Allow	Deny	Deny	Deny

3.2.1 Privacy enforcement between client devices

The learn and share module on server side shares the owner's contextual information with corresponding client device. The client device further keeps track of this context and responds to queries made by other peer devices. Table 3.1 shows the sample contextual information for user "Alice". This contextual information needs to be protected and should be shared only with requesters having sufficient privileges. The user can provide detailed privacy policies specifying what context information can be shared with whom, when, and under what conditions. If the users are reluctant to provide any specific policies then they can opt for either default models of the system viz. (i) Optimistic Model - where the system can provide response to any query with all possible relevant information associated with a user's activity such as associated place, location and the timing details, or (ii) Pessimistic Model - where the system can refrain from revealing activity associated information. Apart from these default system settings the user can define her privacy rules with various degrees of accuracy levels. She can also use the system to obfuscate certain pieces of information to protect the context information. This way our system can protect the user's privacy by varying accuracy levels of activities, associated locations and timestamps.

Whenever any participant in the system tries to access any protected resource (activity, place, location or any additional information) the query is sent to the privacy control module. This module fetches the user knowledge, dynamic knowledge and user-specified privacy preferences to evaluate the query. As a result it will decide whether the participant is allowed to access to protected resource or not. In former case, it might obfuscate certain pieces of the information as per user-specified privacy policies to protect user privacy.

Consider sample privacy policies for different cases:

1. Policy to share context information based on user's group information: *Share detailed contextual information with family members all the time.* This policy checks

Table 3.6. Policy to share activity information with friends all the time except when a user is attending lecture.

```
[ShareActivityWithFriendsRule:
    (?requester a ex:requester)
    (?requester ex:memberOf ?groupFriends)
    (?groupFriends foaf:name "Friends")
    (?someActivity platys:is_performed_by ex:Alice)
    notEqual(?someActivity, platys:Listening_To_Lecture)
->
    (?requester ex:activityAccessRule _:policy5)
    (_:policy5 ex:activityAccess ex:userPermitted)
]
```

whether the requester is a part of family group defined by the user and then decides to share context information accordingly. The Table 3.4 shows the policy represented in Jena rule syntax.

2. Policy to share context information based on the user's context: *Share my activity with friends all the time except when I am attending a lecture.* This policy decides the activity information sharing based on the current activity of user. If the user is attending a lecture then it won't share the activity information with requester. The Table 3.6 shows this policy as a Jena rule.

3. Policy for sharing information based on temporal restriction: *Do not share my sleeping activity with Teachers on weekdays from 9am-9pm.* This set of policies shares the information based on time aspect of the context. In this case, user's activity information won't be shared with Teachers on weekdays from 9am-9pm. The Table 3.7 shows the corresponding Jena rule.

Table 3.7. Policy to not share sleeping activity with teachers on weekdays from 9am - 9pm.

```
[ShareActivityWithTeachersRule:
    (?requester a ex:requester)
    (?requester ex:memberOf ?groupTeachers)
    (?groupTeachers foaf:name "Teachers")
    (?requester ex:requestTime ?localTime)
    (?localTime time:dayOfWeek ?day)
    ge(?day, 1) le(?day, 6) (?localTime time:hour ?hour)
    ge(?hour, 9) le(?hour, 21)
    (?someActivity platys:is_performed_by ex:someUser)
    equal(?someActivity, platys:Sleeping)
->
    (?requester ex:activityAccessRule _:policy6)
    (_:policy6 ex:activityAccess ex:userProhibited)
]
```

4. Policy for information sharing based on requester's context : *Share my context with anyone attending same class as me.* This policy deals with the contextual information of both requester and owner. The requester can be anonymous but it need the requester context to decide information sharing. In this case, it checks the current location and activity of both user and the requester at given time. If the values are matching then user context information is shared with the requester. The corresponding Jena rule is shown in Table 3.8.

5. Policy using activity generalization for sharing : *Share my activity with friends if it's public.* This is an example of activity generalization to share activity information of specific granularity level. This policy allows the activity information sharing to friends iff the activity is declared as public by the user. In any other case it will just

Table 3.8. Policy to share context with anyone attending same class as user.

```
[Rule7:
   (?requester ex:requester ?someValue)
   (?requesterActivity platys:is_performed_by ?requester)
   (?requesterActivity platys:occurs_at ?requesterPlace)
   (?requesterPlace platys:has_location ?requesterLocation)
   (?requesterLocation platys:part_of ?requesterRoom)
   (?requesterRoom rdf:type platys:Room) (?user ex:systemUser ?userValue)
   (?userActivity platys:is_performed_by ?user)
   (?userActivity platys:occurs_at ?userPlace)
   (?userPlace platys:has_location ?userLocation)
   (?userLocation platys:part_of ?userRoom)
   (?userRoom rdf:type platys:Room)
   equal(?requesterRoom, ?userRoom)
   equal(?requesterActivity, ?userActivity)
   equal(?userActivity, platys:Listening_To_Lecture)
->
   (?requester ex:contextAccess ex:userPermitted)
]
```

Table 3.9. Policy to share current activity with friends if it's public.

```
[Rule4:
    (?requester ex:requester ?someValue)
    (?requester ex:memberOf ?groupFriends)
    (?groupFriends foaf:name "Friends")
    (?someActivity platys:is_performed_by ?someUser)
    (?someActivity platys:has_visibility ?visibility)
    equal(?visibility, platys:Public)
->
    (?requester ex:activityAccessRule _:policy4)
    (_:policy4 ex:activityAccess ex:systemPermitted)
    (_:policy4 ex:activityAccessLevel platys:Public)
]
```

deny the access queries. The corresponding Jena rule is shown in the Table 3.9.

6. Policy using activity generalization for sharing : *Share my public activity with friends.* This case of generalization can be used to share the less accurate context information with requester rather than current precise context information. This policy ensures that activity having public visibility is shared with requester even if current activity has any other visibility. E.g. if the current activity has Private visibility then system will traverse the hierarchical activity model to fetch the activity with Public visibility. If it succeeds to get such activity then it will be shared with requester. The corresponding Jena rule is shown in Table 3.10.

7. System-level policy : *Do not share user's context if she is inside BuildingXYZ.* This is an example of system-level policy enforced on system. This set of policies always override the user-defined policies. It outputs systemPermitted and systemProhibited values for contextAccess predicate. The sample policy is shown in the Table 3.3.

Table 3.10. Policy to share public activity with friends.

```
[Rule4:
    (?requester ex:requester ?someValue)
    (?requester ex:memberOf ?groupFriends)
    (?groupFriends foaf:name “Friends”)
->
    (?requester ex:activityAccessRule _:policyRule2)
    (_:policyRule2 ex:activityAccess ex:userPermitted)
    (_:policyRule2 ex:activityAccessLevel platys:Public)
]
```

The described policies are tried out in our system and system behaved as expected. The system results for these policies are described in System Evaluation section.

3.2.2 Privacy enforcement over the sensed data

The sensor data collected by client devices is sent to the server for inferring the dynamic context of an user. As users can be sensitive about how sensor data is captured and used, it is best to let them control how their sensor information is released. It can be done by providing users an option to specify privacy policies to protect the sensed data. Before the data is collected from sensors in continuous sensing or whenever there is a request for sensed data, the privacy control module evaluates the user-defined privacy policies and decides which sensor data can be shared. Only allowed sensors' data is collected and sent to the server for further context inferring. For instance, user can have policy like *“share GPS co-ordinates on weekdays from 9am-5pm only if he is in office”*. Table 3.11 shows it's corresponding Jena rule. This policy allows the sharing of GPS sensor information on weekdays during daytime iff user is in office.

Table 3.11. Policy to share GPS coordinates on weekdays from 9am-5pm only if user is in the office.

```
[ShareGPSRule:
    (?requester ex:requestTime ?localTime)
    (?user ex:systemUser ?true)
    (?localTime time:dayOfWeek ?day)
    ge(?day, 1) le(?day, 6)
    (?localTime time:hour ?hour)
    ge(?hour, 9) le(?hour, 17)
    (?user ex:Latitude ?latitude)
    (?user ex:longitude ?longitude)
    Equal(?latitude, ?officeLat)
    Equal(?longitude, ?officeLong)
->
    (?requester ex:canAccessGPSCoordinates “True”)
    (?requester ex:canAccessActivityPlace “True”)
    (?requester ex:canAccessActivityTime “True”)
    (?requester ex:canAccessPlaceLocation “True”)
]
```

Table 3.12. Policy to share accelerometer readings, WiFi AP ids and recorded audio.

```
[ShareAccelerometerRule:
    (?requester ex:requestTime ?localTime)
    (?localTime time:dayOfWeek ?day)
     ge(?day, 1) le(?day,6)
->
    (?requester ex:canAccessAccelerometerReadings “True”)
    (?requester ex:canAccessWiFiIds “True”)
    (?requester ex:canAccessAudioData “False”)
]
```

In another case, a user can have policy like *“Do not allow access to recorded audio but allow access to accelerometer and WiFi AP ids on weekdays”*. Table 3.12 shows corresponding Jena rule syntax.

3.2.3 Privacy enforcement at the server side

At server side, the learn and share module, infers the dynamic context of an user including current activity, associated place and location and nearby people. This contextual information needs to be protected and should be shared with requesters having sufficient privileges. The server has information about all the system users whereas a client device has information about it’s owner only. Due to this difference the server can handle requests for all the system users whereas the client device can handle requests about it’s owner only. The main distinction between the access requests made by a client device to a peer device and to a server is that the latter request contains a specific userId. This userId is used to retrieve information of specific user. Consider a privacy policy as shown in the table 3.13, which states *“allow location access to teachers on weekdays only between 9am*

Table 3.13. Policy to share location with teachers on weekdays only between 9am and 6pm.

```
[ShareActivityWithTeachersRule:
    (?requester ex:memberOf ?groupTeachers)
    (?groupTeachers foaf:name “Teachers”)
    (?requester ex:requestTime ?localTime)
    (?localTime time:dayOfWeek ?day)
    ge(?day, 1) le(?day, 6)
    (?localTime time:hour ?hour)
    ge(?hour, 9) le(?hour, 18)
    (?user ex:systemUser ?true)
    Equal(?user, ?userId)
->
    (?requester ex:activityAccessRule _:policy6)
    (_:policy6 ex:activityAccess ex:userProhibited)
]
```

and 6pm”. The system uses the userId to retrieve specific user information and uses it to verify whether the requester is a teacher or not. The example explained above involves the representation of a user’s personal resources such as list of friends, groups information, contextual attributes like current location and current activity.

Chapter 4

SYSTEM IMPLEMENTATION

Our primary goal for the prototype was to use Semantic Web based policy framework to demonstrate strong access control over the static and transient user information in a collaborative context-aware geosocial networking system. We have used location-aware devices such as iPhone or Google Android phone as client devices in our prototype implementation. Our mobile application collects the sensor data and sends it to the server for processing. The server side module has provision to collect data from various online sources such as Google Calendar or social networking sites such as Facebook. This module can collect user profile information and find networks of their friends. Also users can create their own networks and add people in that. Using this information and sensor data the module can infer contextual information of user. The user context is shared with corresponding client device so that both client and the server can respond to access queries from requester.

Our system uses a common protocol which defines a generic request and response formats. It ensures that the request and response are easily constructed and interpreted by servers and clients. It enables system to have heterogeneous components acting as both server and client. We have developed android-based applications to use Android phone as client and server respectively. The similar applications are developed for personal com-

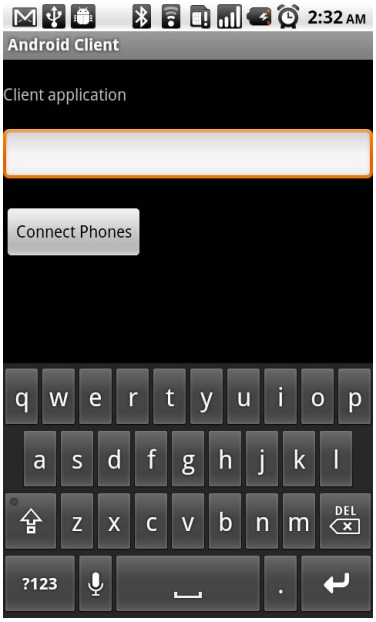

FIG. 4.1. Android client application user interface.

puters so that system can support all sorts of client and server combinations. The system uses sockets to establish two-way communication link between a server and clients. The Figure 4.1 and Figure 4.2 shows the Android client and server application respectively.

The requester can use client application and query for user context. This query is processed by the policy framework and it's result is shown to the requester with valid accuracy level. The Figure 4.3 shows the result obtained on to the server for context query whereas Figure 4.4 is a result returned to the requester on client device.

In the implementation, we have used contextual information as the resource that changes dynamically for the user, and have provided mechanisms to specify more expressive policies to control the sharing of contextual information. The user can create policies by using Policy Editor interface as explained below.

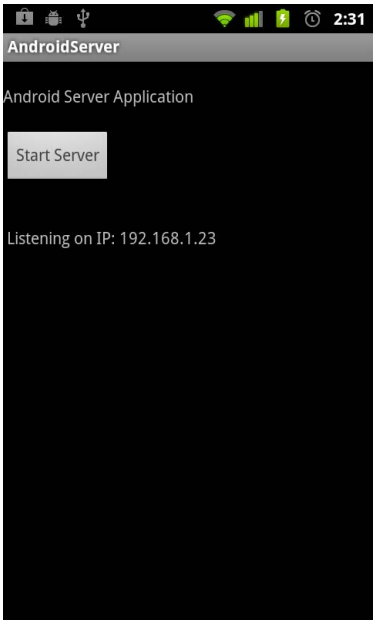

FIG. 4.2. Android server application user interface.

4.0.4 Specifying a privacy policy

The user can use the web interface on client device to specify and edit privacy preferences. It can be used to specify access control rule as - 'who' by selecting friends or groups of friends, 'what' by selecting resources such as location or activity, 'conditions' by selecting allowed days of the week or specifying the allowed time range during day or by specifying region on the map as sensitive. The user can also specify allowable type of activity like sleeping, eating, working, chilling. Figure 4.5 describes the sample privacy rule editor for client devices. The policies are created and stored in N3 format on both server and client sides in persistent memory and reloaded when required by reasoning engine. The current implementation does not provide user interface to generate policy required for the explain justification of the policies.

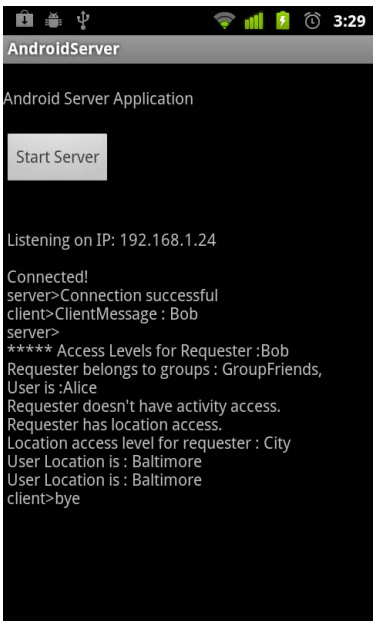

FIG. 4.3. Result on server side.

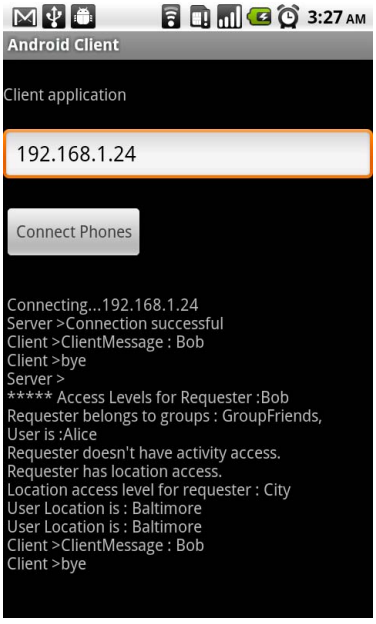

FIG. 4.4. Result on client device.

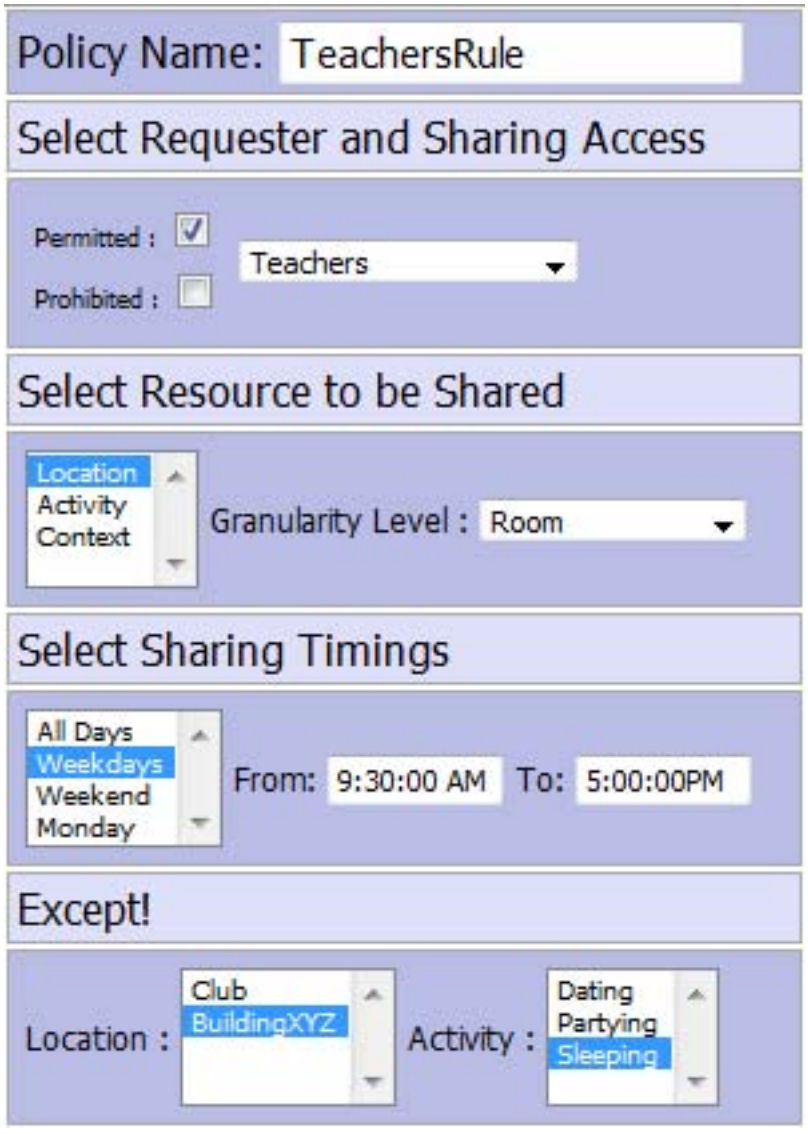

FIG. 4.5. Policy editor for a client device.

Chapter 5

SYSTEM EVALUATION

The goals of evaluation were (i) to see if the system satisfies a basic criteria by allowing access from privileged user and restricting illegal user, (ii) to test whether the actual computing time of reasoning over mobile devices is acceptable and (iii) to determine how it scales with different size of user information like number of users in group list. The following sections elaborates each of these goals and evaluation results.

5.1 System Validation

The main objective behind system validation was to verify whether the system allows access from privileged users and restricts illegal user as per privacy policies. The privileged user is a requester who is allowed to access user's context as per user-specified privacy rules whereas other's are modeled as illegal users. To perform the validation, we designed the use cases with sample user information, group information and privacy policies. We changed the requester or requester context in each of these use cases. The results were initially inferred manually and then compared with system results having same settings. The system behaved as expected by allowing information access to privileged users and denying access to illegal users as per user-defined privacy rules. Our sample use cases and information is described as below:

5.1.1 System Level Policies

1. *Share detailed context information with family members.*
2. *Share user's building-wide location with teachers on weekdays only between 9 am and 6 pm..*
3. *Share user's citywide location with everyone.*
4. *Do not share user's super-private activities with anyone.*

5.1.2 User Level Policies

1. *Do not share my context if I am in a meeting with Professor..*
2. *Share my Semipublic activity with friends.*
3. *Do not share my sleeping activity with teachers on weekdays between 9am-9pm.*
4. *Do not share my context when I am partying.*
5. *Share my working activity with my family.*
6. *Share my room-wide location with everyone in the same building as me.*
7. *Share my context with anyone attending same class as me.*

The user profile and contextual information is shown in the Table 3.1 and requesters are Jon - teacher, Bob - friend, Ron - family member and Pramod - user not belonging to existing groups.

- If the context access request comes from requester Ron (a family member) then he should be able to access user context as per system policy 1. The Figure 5.1 shows response on a client device of Ron for this request. It clearly shows the system

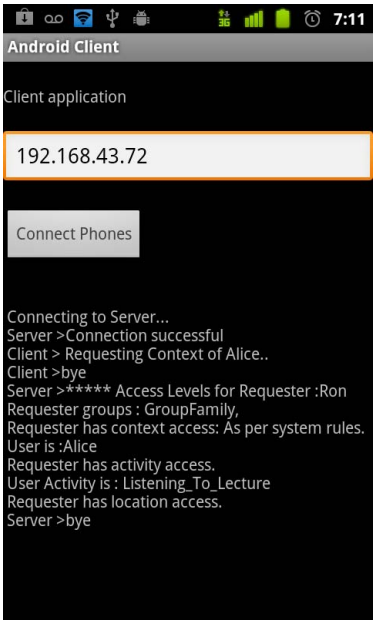

FIG. 5.1. The system response to a context access query made by a family member.

has recognized that the Ron has context access and hence, user's current context information is shared.

- Suppose the access request comes from Bob (a friend) then he should be able to access user's SemiPublic activity and citywide location only. He is not allowed to access user's detailed context. The Figure 5.2 shows the response for context access query, the Figure 5.3 shows the response for activity access query and, Figure 5.4 shows the location access query response.

- Consider, Jon (a teacher) queries the context of this user. As per system-level privacy policies, he cannot access detailed user context but can access building-wide location on weekdays between 9am-9pm and city-wide location all the time. The Figure 5.5 shows the response for context access query and Figure 5.6 shows the response for location access query made on weekend.

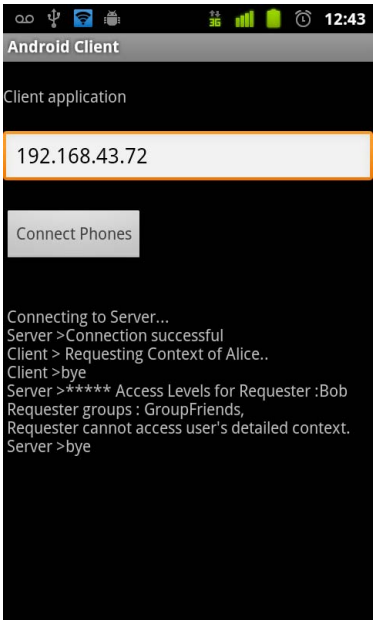

FIG. 5.2. The system response to a context access query made by a friend.

- This case deals with context of both requester and the user. Consider some user "Pramod" who is not a part of existing group lists, queries for the user context. As per user-level privacy rules, user context can be shared with such requesters only if these requesters are attending same class as user. The Figure 5.7 shows the system response for context query made by such requester whose attending same class as user and Figure 5.8 shows the response when requester is not attending same class as user. As shown in the response, system successfully identifies that the requester doesn't belong to existing groups.

5.2 System Performance

We have evaluated the system performance in terms of reasoning time taken for the requester query. It is measured when the access requests are made to server PC and to the android client device. To evaluate scalability of the system, we varied the number of users

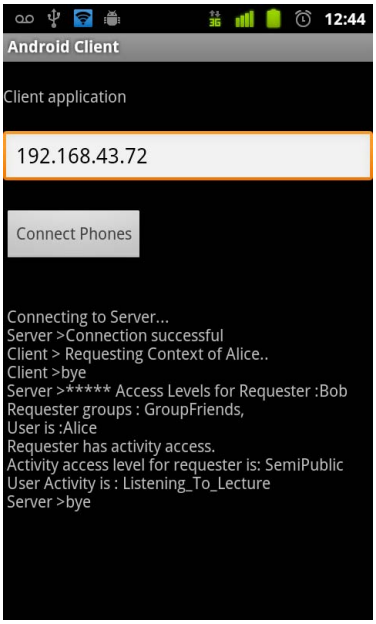

FIG. 5.3. The system response to a activity access query made by a friend.

in group list and noted the time taken (reasoning time) by the system to provide access levels for the requester. The Table 5.1 shows the results of evaluation where the obtained values are average of several computations. It clearly shows that reasoning on mobile devices can be done without any scalability issues and it can be efficiently used to enforce privacy over sensed and contextual data. Figure 5.9 shows the growth of reasoning time (in milliseconds) against number of users in the group list.

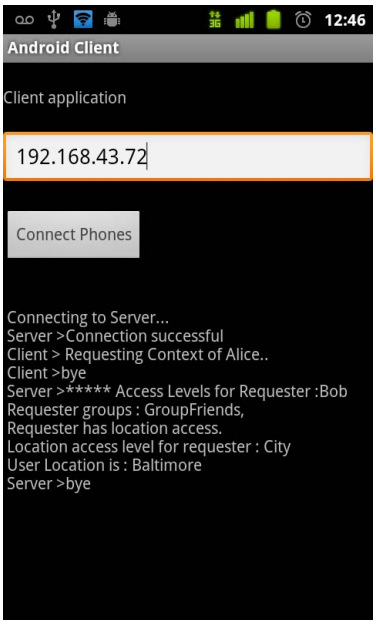

FIG. 5.4. The system response to a location access query made by a friend.

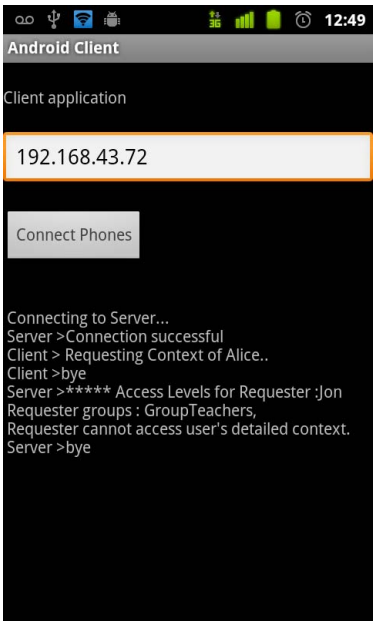

FIG. 5.5. The system response to a context access query made by a teacher.

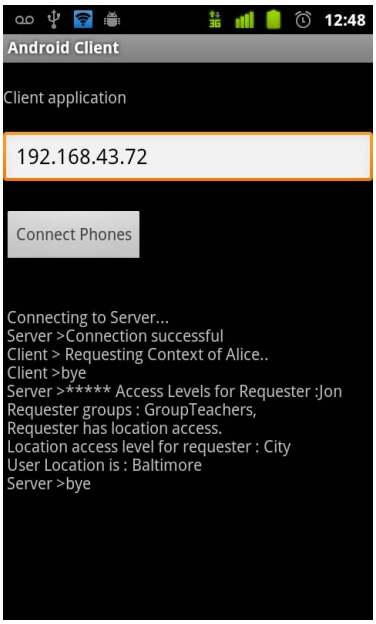

FIG. 5.6. The system response to a activity access query made by a teacher on weekend.

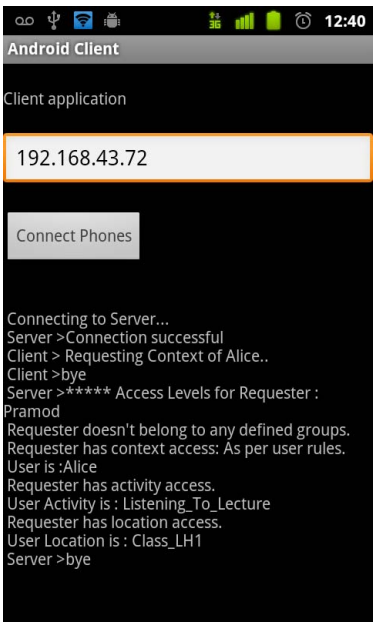

FIG. 5.7. The system response to a context access query made by a random requester attending same class as user.

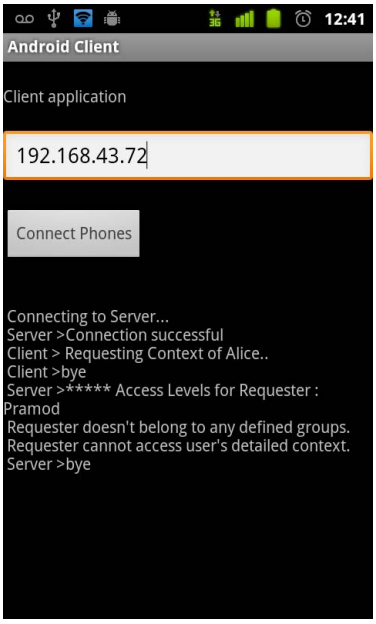

FIG. 5.8. The system response to a activity access query made by a random requester with different context.

Table 5.1. Reasoning time for different number of users.

Numbers of users	On server machine		On Android device	
	Reasoning time(ms)	Standard deviation	Reasoning time(ms)	Standard deviation
10	1177	142	1128	15
50	1246	74	1446	46
100	1993	26	1903	118
250	2448	184	2682	165
500	3042	108	4233	245
1000	3715	456	10896	393

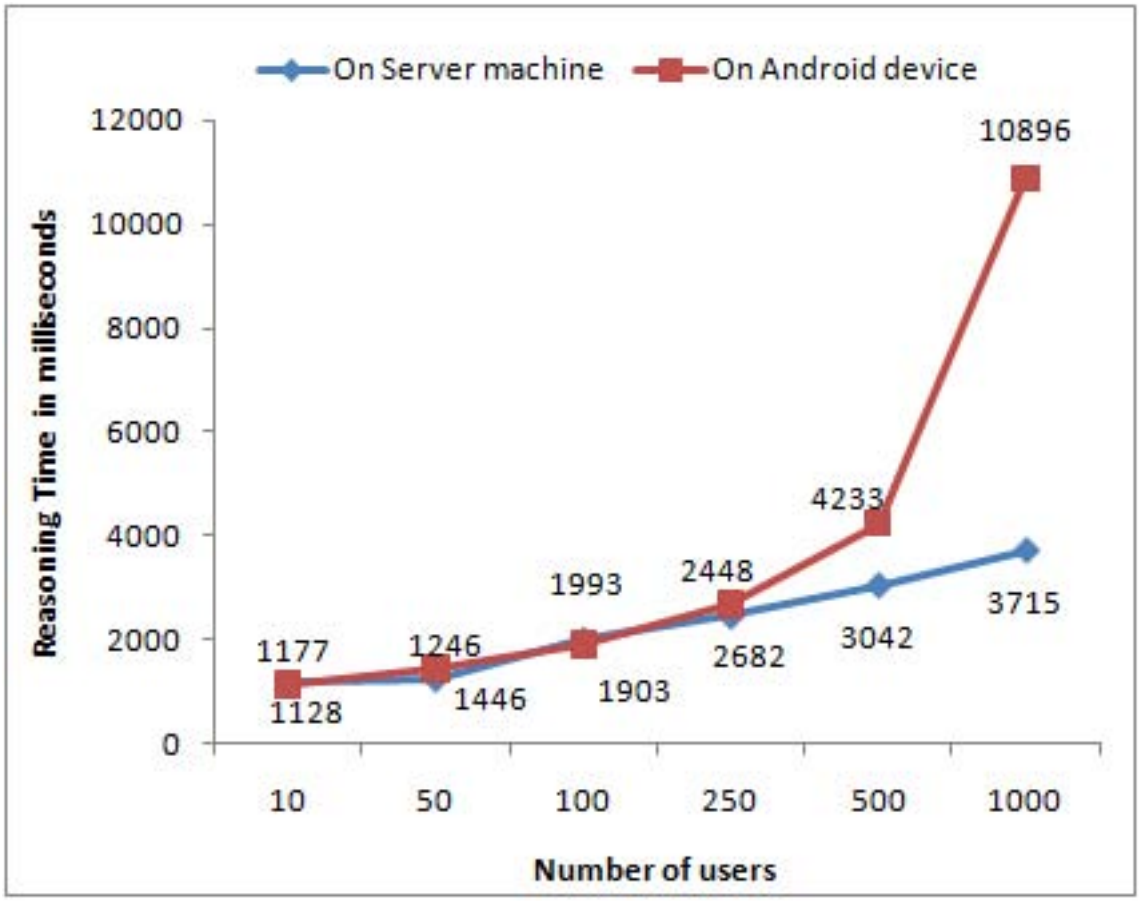

FIG. 5.9. Reasoning time (in milliseconds) for different number of users in owners group list.

Chapter 6

CONCLUSION AND FUTURE WORK

Our mobile devices are becoming the dominant way we communicate with people, access information, and consume services. As they become more intelligent, they can and will model our interests, activities and behavior in order to understand our current context and using it, better serve our needs. When appropriate, aspects of this learned context may be shared with other devices in order to collaborate and provide enhanced service. This development introduces a strong need to allow users greater control of what information is shared with who and with what level of detail.

We described a policy based framework to control information flow in collaborative context aware geo-social networking application. It allows users to specify a rich suite of privacy preferences that consider the static and dynamic knowledge about user, along with generalization rules to regulate the accuracy of results. Protected resources can be activities, location information, or media such as photos, videos posted by participants of the social network. We showed some example policies that state of the art systems do not support. Our privacy mechanisms constitute a baseline that can be extended and incorporated by any of the existing social networks including location based mobile social networks. We plan to extend the prototype implementation to address the engineering challenge of scalability. We plan to carry out user studies to evaluate the utility of the proposed privacy

control mechanisms. We also plan to address the issues of incorporating incentives to allow for even more flexibility in the definition of policies for context-dependent release of information.

REFERENCES

[1] Acquisti, A., and Gross, R. 2006. Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook. In *Proceedings of 6th Workshop on Privacy Enhancing Technologies*.

[2] 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.

[3] Bechhofer, S.; van Harmelen, F.; Hendler, J.; Horrocks, I.; Mcguinness, D. L.; Patel-Schneider, P. F.; and Stein, L. A. OWL Web Ontology Language Reference. Technical report, W3C.

[4] Beckett, D. 2007. Turtle - Terse RDF Triple Language. Technical report.

[5] Berners-Lee, T., and Connolly, D. 2008. Notation3 (N3): A readable RDF syntax. Technical report.

[6] Berners-Lee, T. Cwm - a general purpose data processor for the semantic web.

[7] Berners-Lee, T.; Connolly, D.; Prud'hommeaux, E.; and Scharf, Y. 2005. Experience with n3 rules. In *Rule Languages for Interoperability*.

[8] Carroll, J. J.; Dickinson, I.; Dollin, C.; Reynolds, D.; Seaborne, A.; and Wilkinson, K. 2004. Jena: implementing the semantic web recommendations. 74–83. New York, NY, USA: ACM.

[9] Cheverst, K.; Davies, N.; Mitchell, K.; Friday, A.; and Efstratiou, C. 2000. Developing a context-aware electronic tourist guide: some issues and experiences. In *CHI*, 17–24.

[10] Doyle, J. 1978. Truth maintenance systems for problem solving. Technical report, Cambridge, MA, USA.

[11] Dwyer, C.; Hiltz, S. R.; and Passerini, K. 2007. Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace. In *Proceedings of the Thirteenth Americas Conference on Information Systems (AMCIS)*.

[12] Gross, R., and Acquisti, A. 2005. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, WPES '05, 71–80. New York, NY, USA: ACM.

[13] Hayes, G. R.; Patel, S. N.; Truong, K. N.; Iachello, G.; Kientz, J. A.; Farmer, R.; and Abowd, G. D. 2004. The personal audio loop: Designing a ubiquitous audio-based memory aid. In *In Proceedings of Mobile HCI*, 168–179. Springer Verlag.

[14] Iachello, G.; Truong, K. N.; Abowd, G. D.; Hayes, G. R.; and Stevens, M. 2006. Prototyping and sampling experience to evaluate ubiquitous computing privacy in the real world. In *Proceedings of the SIGCHI conference on Human Factors in*, 1009. Press.

[15] Jajodia, S.; Samarati, P.; Subrahmanian, V. S.; and Bertino, E. 1997. A unified framework for enforcing multiple access control policies. In *In Proceedings of ACM SIGMOD International Conference on Management of Data*, 474–485. ACM Press.

[16] Jones, H., and Soltren, J. 2005. Facebook: Threats to privacy. ethics and the law on the electronic frontier course.

[17] Kagal, L., and Berners-lee, T. 2005. Rein : Where policies meet rules in the semantic web. Technical report, Laboratory, Massachusetts Institute of Technology.

[18] Kagal, L.; Hanson, C.; and Weitzner, D. 2008. Using dependency tracking to provide explanations for policy management. In *Proc. IEEE Workshop on Policies for Distributed Systems and Networks*, 54–61. Washington, DC: IEEE Computer Society.

[19] Kapadia, A.; Kotz, D.; and Triandopoulos, N. 2009. Opportunistic sensing: security challenges for the new paradigm. 127–136.

[20] Kidd, C.; Orr, R.; Abowd, G.; Atkeson, C.; Essa, I.; MacIntyre, B.; Mynatt, E.; Starner, T.; and Newstetter, W. 1999. The Aware Home: A Living Laboratory for Ubiquitous Computing Research. volume 1670. 191–198.

[21] Lassila, O., and Swick, R. 1998. Resource description framework model and syntax specification.

[22] Moses, T. 2005. *eXtensible Access Control Markup Language TC v2.0 (XACML)*.

[23] Sacramento, V.; Endler, M.; and Nascimento, F. N. 2005. A privacy service for context-aware mobile computing. In *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, 182–193.

[24] Schilit, B.; Adams, N.; and Want, R. 1994. Context-aware computing applications. In *In Proceedings of the Workshop on Mobile Computing Systems and Applications*, 85–90.

[25] Shin, M.; Cornelius, C.; Peebles, D.; Kapadia, A.; Kotz, D.; and Triandopoulos, N. 2010. AnonySense: A system for anonymous opportunistic sensing. *Journal of Pervasive and Mobile Computing*.

[26] van Sinderen, M. e. a. 2006. Supporting context-aware mobile applications: an infrastructure approach.

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

[28] Waterman, D. A., and Hayes-Roth, F., eds. 1978. *Pattern-Directed Inference Systems*.