Semantics for Privacy and Shared Context

Roberto Yus[1], Primal Pappachan[2], Prajit Kumar Das[2],
Tim Finin[2], Anupam Joshi[2], and Eduardo Mena[1]

[1] University of Zaragoza, Zaragoza, Spain
{ryus,emena}@unizar.es,
[2] University of Maryland, Baltimore County, Baltimore, USA
{primal1,prajit1,finin,joshi}@umbc.edu

Abstract. Capturing, maintaining, and using context information helps mobile applications provide better services and generates data useful in specifying information sharing policies. Obtaining the full benefit of context information requires a rich and expressive representation that is grounded in shared semantic models. We summarize some of our past work on representing and using context models and briefly describe Triveni, a system for cross-device context discovery and enrichment. Triveni represents context in RDF and OWL and reasons over context models to infer additional information and detect and resolve ambiguities and inconsistencies. A unique feature, its ability to create and manage "contextual groups" of users in an environment, enables their members to share context information using wireless ad-hoc networks. Thus, it enriches the information about a user's context by creating *mobile ad hoc knowledge networks.*

Keywords: Context, Semantic Web, ontology, OWL, reasoning

1 Introduction

A recent study by Cisco [5] predicts that the number of mobile devices in the world will exceed the population by 2014 and that 54% of them will be smart devices by 2018. In addition to smartphones, other mobile devices like tablets or wearable computing devices (e.g., smart watches and glasses) are also becoming ubiquitous.

These smart devices and their applications store and have access to a great deal of personal information about their users. Since communication is a primary use, they know their users' contacts, phone activity, email correspondents, and social media connections. They have a rich array of sensors that track location and activity and installed software applications can access the device's audio and video streams. Applications may also be able to extract information from stored content that includes users' calendars, email, and social media streams.

This data can be integrated, combined with background knowledge, and reasoned over to produce even more information. Capturing, maintaining, and using such information can help mobile applications provide better services by being aware of their user's context. However, much of the information is personal and sensitive and must be protected from undesired use or disclosure. As a result, there are two related problems inherent in the ecosystem of smart devices, sensors, applications, and users.

The first problem involves protecting the privacy and security of personal data. For example, Android, the most popular mobile operating system as of 2014, follows a single "take-it-or-leave-it" install-time permission acquisition model for handling user data security. This approach assumes that users are able to understand the various permissions requested by application and differentiate between what is acceptable and what is not. Given the volatile nature of personal data, mechanisms which allow flexible management are important to protect privacy and security of the users. To address this, we have been exploring the use of policy-based, context-dependent privacy and security frameworks that manage personal data and access to it. Our frameworks use semantic technologies including Semantic Web representation languages (OWL and RDF) and tools for reasoning with and enforcing context-dependent information sharing policies [4, 8, 12, 13].

The use of ontologies for defining contextual concepts such as activity and location enables us to apply our policies over generalized or specific instances of these concepts. It also allows us to define more fine-grained policies for controlling various aspects of information sharing such as what information is shared, with whom it is shared, and in which context. For example, enabling a user to define a policy rule stating that social network apps cannot access a recording device whenever she is in a research meeting with her colleagues. Additionally, in situations where information sharing cannot be stopped, it can also support obfuscation mechanisms, for example showing your location with a 100 kilometer accuracy when you are not at work or are outside normal working hours.

Since the framework requires a rich notion of context to enforce policies, the second problem is how to obtain precise information about a user's context. In previous work [22, 23] we explored how to recognize and distinguish high-level descriptions from low-level sensor data. For example, using supervised learning techniques we can determine a user's activity (e.g., *attending_class*) and the role she is playing in it (e.g., *student*). By accessing background knowledge available from resources like DBpedia and Geonames we can map GPS locations to typed places. For example, from coordinates *(39.253798,-76.714354)* we can learn that we are in the *ITE building* which is part of *UMBC* which is an *EducationalInstitution* and located in *Maryland*.

Such context information can help applications match user requirements to available services. For example, by using the location of a user, applications like Foursquare or Yelp can recommend restaurants. Activity tracking applications like Endomondo allow users to track their daily activity and help them stay fit. Systems like SHERLOCK [21], which is a distributed architecture for mobile devices, provide users with information about potentially interesting services in the vicinity taking the context of the user into account. Context awareness has been useful in community health care scenarios as well [16].

Research in context-aware computing has predominantly focused on context synthesizing [17] high-level context information from low-level information such as sensor or user data [9, 14] (e.g., to infer that the user is in a meeting after considering the data from sensors such as microphone, GPS, and the calendar entry as busy). However, not all mobile devices have all the sensors nor does the user provide their devices with all the data. The issue of data availability can be overcome by sharing relevant context pieces among users from devices nearby, which can help in creating a shared context model for these devices. For example, my device may know I am located at UMBC but not know what activity I am engaged in. Devices of nearby students may know and be willing to share the fact that they believe they are attending a class with name *Operating Systems* in room 231 of the ITE building. We can use this knowledge of commonality of context across devices for building a richer shared, semantic context model.

The remainder of this paper is organized as follows. In Section 2, we introduce Triveni, a system that allows collaborative information to be shared within a group of devices, and present a motivating use case. In Section 3, we show the high-level architecture of the Triveni system and the acquisition of context information and its integration performed by the system. Finally, Section 4 surveys related works and conclusions and future work are presented in Section 5.

2 Exploring Collaborative Context Discovery

Triveni[3] is an experimental framework that allows a group of devices to discover one another and share context information by combining existing techniques of ad hoc network management, data access control, and semantic data management. The potential is that all of the devices will benefit from a richer and more accurate model of their context. Information privacy is protected by policies running on each device that decide what information is made available to other devices.

Triveni builds knowledge ad hoc networks [11] to enable mobile devices to create "contextual groups" among them and exchange relevant knowledge in a secure and private manner. Our Triveni prototype implements methods to gather and integrate high-level context pieces from multiple devices to produce an enriched context which is available for all participating devices. The use of OWL ontologies and Description Logic reasoners lets it detect and resolve conflicts and inconsistencies in the shared context, which occurs due to lack or misinterpretation of low-level data and device failures.

2.1 Motivating Use Case

Jeff, Abed, and Annie are students at Greendale Community College (GCC) and members of a study group that meets on Wednesdays in the study hall of the

[3] Triveni Sangam is a confluence of three rivers and the point of confluence is said to flush away all of one's sins.

library. On a given Wednesday, Pierce, who is not part of the study group, is using his computer in the study hall.

There are five mobile devices on this particular Wednesday (see Figure 1): Jeff's tablet and smartphone, Abed's laptop, Annie's tablet, and Pierce's laptop. Notice that these mobile devices are equipped with different number of sensors (from smartphones that have compass, accelerometer, gyroscope, and GPS, among others, to laptops that do not have any of these sensors). In addition, the information that users provide can also be different and in varying detail. In our specific scenario, Annie's and Abed's calendar entries give information about meeting scheduled for that day (e.g., duration, topic, and participants) and Jeff's smartphone just used Foursquare to check in *Study room F*. With the information available on their devices, traditional context generation systems will create different high-level context information for each device (see Figure 1 where the current context of each user is shown in blue boxes).

Notice that some of the contexts obtained by the devices are wrong. For example, Annie disabled the location gathering mechanism of her tablet, while at home, to save battery and so, her device thinks that the location is still "home". Summarizing, the devices have some information about the context but most of them are not as rich in detail as it would be desired.

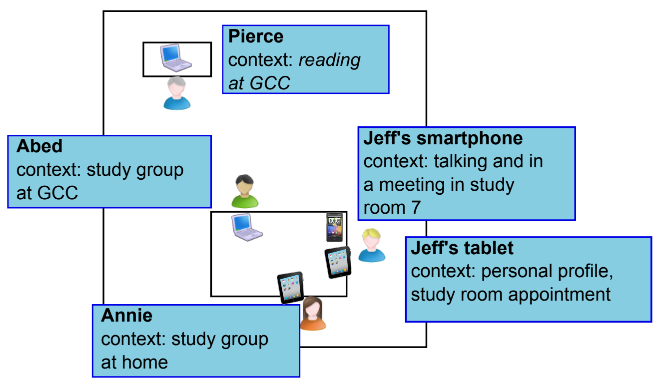

Fig. 1. Motivating use case: users being part of a study group.

In this scenario, the best possible context for these mobile devices would the following:

- For everyone in the study group (Jeff, Abed, and Annie) → *"study group about Spanish with a duration of one hour with three participants: Jeff, Abed, and Annie"*.
- For everyone in the library (Jeff, Abed, Annie, and Pierce) → *"Study room F inside Greendale Community College at* 25° *C and with the lights on"*.
- For every device of Jeff (Jeff's tablet and smartphone) → *"heart rate 70bpm"*.

However, not every mobile device has access to the information needed to compute the fully enriched context. Nevertheless, the collaboration among them can be used to address this problem.

3 Architecture of the System

Triveni's primary goal is to enrich the information about a user's context, obtained by context synthesizers, by leveraging the context of other users nearby[4]. This way, applications are able to make use of the enriched context provided by the system. Triveni has a decentralized architecture where mobile devices communicate among themselves using wireless ad hoc networks and exchange their context (see Figure 2 for the high-level architecture of each Triveni node). Therefore, Triveni:

1. Obtains the context information (see Section 3.1) from: 1) the available *Context Synthesizer(s)* (*Context Manager* module); and 2) devices discovered in the vicinity (*Communication* module).
2. Reconciles the context information collected (see Section 3.2) to generate the shared context models (*Integration* module) verifying the information integrated to resolve inconsistencies (*Inconsistency Resolving* module).

Triveni uses Semantic Web technologies (specifically OWL ontologies and semantic DL reasoners) for context modeling. This allows Triveni to detect inconsistent information by using the reasoner and also to infer additional information which has not been explicitly stated. Another benefit of using ontologies to model context is that it would be possible to reconcile different context models/definitions by using ontology alignment techniques [18]. Ontologies have been widely used before to define and extract context [3]. However, for the sake of simplicity, we consider that mobile devices in our system use a common ontology for context definition (see Figure 3 for an excerpt of the ontology defined for our use case). We advocate using a local reasoner, a program that infers logical consequences from a set of asserted facts or axioms [7], on each device. The use of semantic reasoners (and Semantic Web technologies in general) on mobile devices has been studied in [20] and our results show that today's mobile devices can handle semantic reasoning.

3.1 Context Acquisition

To acquire information about the context of a user, Triveni modules running on each device first obtain the high-level context of the user using context synthesizers, specifically those modeling the context using ontologies [3]. Such context synthesizers can have varying degrees of certainty with regards to the accuracy of the high-level context based on factors such as: 1) the sensors used in obtaining the data; 2) liveness inferred from sensor update frequency; and 3) the

[4] We refer to users and their mobile devices interchangeably along the paper.

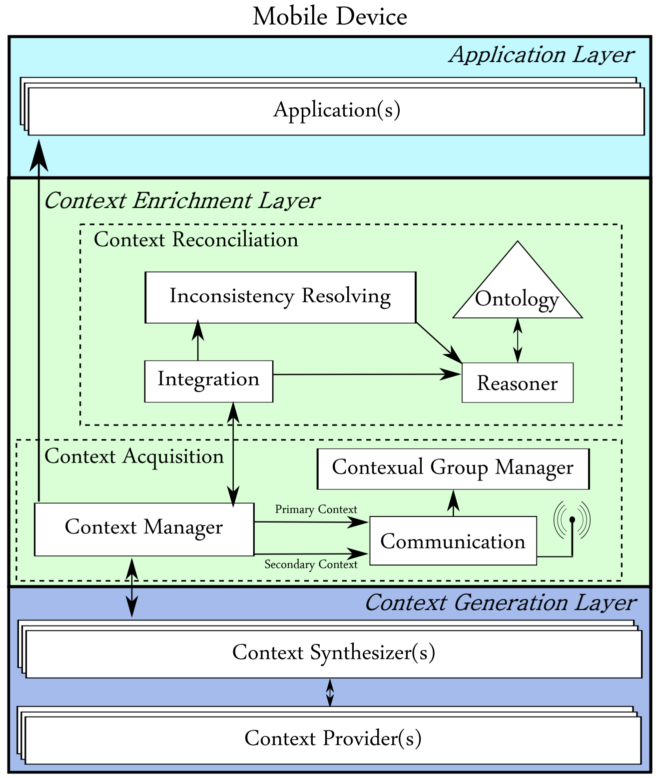

Fig. 2. High-level architecture of a node using the Triveni framework.

accuracy versus power trade off considerations. Triveni uses context synthesizers which can provide a measure of confidence for RDF relations, which gives the probability that the high-level context fact is true [2, 17].

Triveni leverages the context information provided by nearby devices to enrich the context model of the user. For this task, the system creates ad hoc wireless networks to communicate with other devices and to exchange context information. The system considers short to mid range communications to discover users nearby (e.g., the same room) because, in general, nearby users share the same location and activity. While same activity could be performed by more users located outside this range, this is out of the scope of this paper.

When connecting with other devices through wireless networks, Triveni must ensure that no eavesdroppers take part in the communication. We use Diffie-Hellman key exchange for sharing the secret key which can be further used in encrypting information shared between Triveni devices. This shared key can be used in symmetric encryption techniques such as Advanced Encryption Standard or Blowfish. Thus we utilize a decentralized key agreement protocol similar to [1]. Triveni then creates contextual groups among the devices connected enabling devices to exchange contextual information that is interesting for them and thus

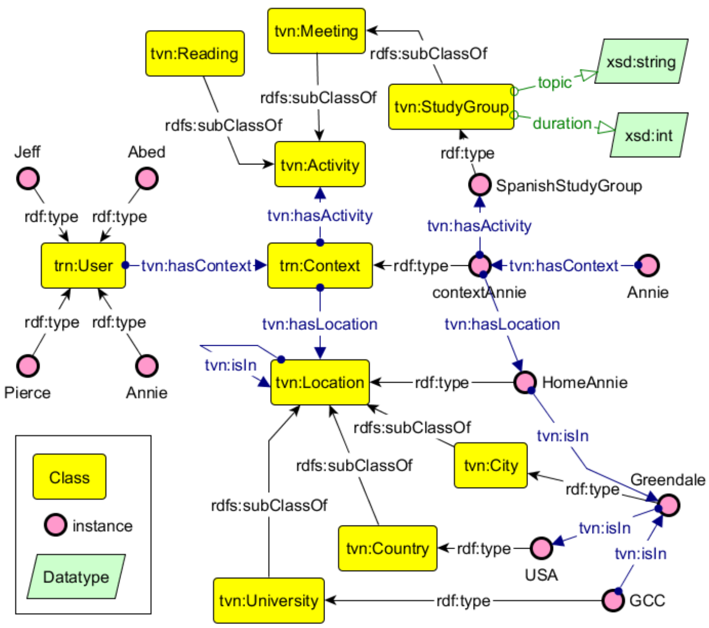

Fig. 3. An excerpt of the ontology used in our sample scenario.

(1) limit the amount of data to share, and (2) allow them to share additional details only with mobile devices of the group.

Once this network of mobile devices has been established the next step is to exchange context information among the members. For this purpose, a mobile device requests context information from other devices in the group. The device which receives the request replies with the appropriate information taking into account the user defined policies, based on various context options, to determine what pieces of context are shareable, under what contextual situations, and with whom. The idea of using context-aware policies for sharing of any private user data was explored in our previous work [12].

3.2 Context Reconciliation

When mobile devices using different context providers and synthesizers exchange context information, some of which is possibly imprecise, sometimes there can be divergent information. For example, consider Annie's tablet which receives information about the location such as *GCC* and *Study room F* while her own device thinks the location is *Home* (see Table 1). In this situation and for each piece of context, the system has to determine from all the possible values which ones are most likely.

Triveni uses the semantic reasoner and ontology to deduce if a given fact is supporting a different one. For example, Jeff's smartphone states that its

Identity	Location	Source	Confidence	‖Confidence‖
Annie	Home	Calendar	0.75	0.23
Abed	GCC	GPS and Geonames	0.8	0.25
Jeff	Study room F	Foursquare and GPS	0.7	0.22
Pierce	GCC	IP address	0.9	0.29

Table 1. Contextual information shared about location.

location is *Study room F* and so, Jeff's device is implicitly supporting that its location is *GCC*. Therefore, in the example of Table 1 three devices support that the location is GCC (Pierce's, Abed's, and Jeff's). The same situation can arise with activities, both Abed's and Annie's devices share that the activity performed is a *tvn:Study Group* and so, they support the *tvn:Meeting* activity shared by Jeff's device.

For each different context piece, cp_x, (e.g., cp_{loc} for location) we have to compute a global confidence on each of the different facts shared, f_i, (e.g., GCC, Home, and Study room F) taking into account that some of them can be supported by more than one device (e.g., GCC is supported by Pierce, Abed, and Jeff as mentioned before). Then, let T be the set of normalized confidence values related to a piece of context cp_i, and let S be the set of normalized confidence values that support a context value f_i (e.g., location facts from Abed's and Pierce's devices support location as GCC). We sum up the values in S and normalize it over T, to compute the global confidence gc_i, as follows:

$$gc_i = \frac{\sum_i nc_i}{\sum_j nc_j} \qquad \forall c_i \in S, nc_j \in T \tag{1}$$

After the context integration process, the system will finally obtain a list of candidate context pieces with their computed confidence, $GC(cp_x)$. In our previous example, the final possible locations computed for the users along with their confidences are: $GC(cp_{loc})$={*Home(0.23), GCC(0.77), Study room F(0.22)*}. Notice, that there is an inconsistency in this shared primary context information as there are two conflicting locations present, namely *Study Room F/GCC* and *Home*.

To detect semantic inconsistencies, constraints should be modeled in the ontology. For example, in the context ontology that we defined for our use case (see Figure 3) we stated that a user can only have one location (by defining the *tvn:hasLocation* property as functional), and that the activity class *tvn:Standing* is disjoint with the class *tvn:Running*. Triveni uses the context facts along with their confidence values for inconsistency detection and resolution. For each piece of context, cp_x, and the list of possible values, $GC(cp_x)$, the system reorders $GC(cp_x)$ according to the confidence computed for each element in descending order.

The list of possible locations in our example will be reordered to $GC(cp_{loc}) = \{$*GCC(0.77), Home(0.23), Study room F(0.22)*$\}$. Then, for each element of $GC(cp_x)$, it creates an axiom and materialize it in the local ontology and use the reasoner to check whether the ontology is still consistent. In the case of the reasoner inferring that the ontology is inconsistent, Triveni removes the last axiom materialized because its confidence will be lower than previous one(s).

In some scenarios it is possible that only a few devices share interesting and precise information and so the confidence computed for them will be low (e.g., in our previous example only Jeff shares that the location is Study room F and then the confidence computed is the lowest). However, it is also possible that this low confidence is caused by wrong information being shared. A variety of approaches could be followed to tackle this problem, from conservative solutions (only use the context with the highest confidence) to optimistic approaches (use all context pieces that are not inconsistent). Our system uses a semi-optimistic approach: use all context pieces that are not inconsistent and whose confidence is greater than a threshold value.

4 Related Work and Discussion

Context-aware computing is a very active field. A survey of the literature [2] shows that context extraction and user context generation from mobile sensors or other sources, has been well studied. The techniques proposed can be broadly classified into two categories: the first relies on machine learning models to learn about features from sensor data to predict the user context [14]; the second focuses on defining context using ontologies and rules and uses a reasoner to infer associations between sensor data and user context [3].

In our work we avoid low-level context extraction and focus on using peer-to-peer (P2P) networks of devices to share high level context information and in a context enrichment process. Wibisono et al. [19] also leverages P2P networks of devices in context-awareness computing. However, in their approach, devices in a specific location (e.g., a room) are used to detect the "situation" there (the situation concept they use is similar to the activity concept used in this paper). They integrate low-level sensor information and use machine learning techniques to reason the most probable situation from the previously defined list of situations for the room. In our approach, we consider high-level context information shared by the devices and base our integration on semantic techniques (ontologies and a semantic reasoner). In addition, we do not start with a set of possible situations for a location. Finally, the contextual groups enable us to limit the information that the devices in a P2P network share.

The idea of using user groups to share information, like in a meeting, discussion or party was explored in [10]. They used activity history and contact information to suggest relevant groups. In this paper we have explained the idea of a data-driven need-based contextual group formation. In another work done by Lane et al. [15] crowd-sourcing to correct context classification errors and label sensor-data was exploited. The idea of Community Similarity Networks (CSN) proposed in [15] is however, significantly different from contextual groups defined in this paper. While CSN uses similarity dimensions (based on lifestyle, physical differences and sensor-data) to cluster users, contextual groups are based on the primary pieces of context. Also CSN uses a centralized classification approach to context recognition thereby creating a single point of failure, while contextual groups are distributed in nature.

5 Conclusion and Future Work

A richer notion of context is required for providing relevant services to users and protect their privacy at the same time. We described a system for cross-device, semantic context management that enriches intra-device context. By gathering context pieces collaboratively and reasoning over them to discover and resolve inconsistencies, the system is resilient in the face of missing or erroneous sources of information.

The resulting shared context models have the potential to be more complete and accurate than any of the participating intra-device models. Using the Triveni system, mobile devices perform two key activities: 1) performing cross-device context discovery and integration to create a richer context model using the shared information, and 2) using secure communication and semantic policies to facilitate the exchange of context information within contextual groups.

As future work we plan to evaluate the effectiveness of our system through the development of a prototype. Also, we plan to incorporate transition of context in the groups as well as the collaboration among different groups for further enrichment. Finally, we plan to take into account the trade off between energy consumption and creation and maintenance of contextual groups [6].

Acknowledgments. This research work was supported by the CICYT projects TIN2010-21387-C02-02, TIN-2013-46238-C4-4-R and DGA-FSE and NSF grants 0910838 and 1228198.

References

1. Alvarez Bermejo, J., Lodroman, M., Lopez-Ramos, J.: A decentralized protocol for mobile control access. The J. of Supercomputing 70(2), 1–12 (2014)
2. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. Int. J. Ad Hoc Ubiquitous Comput. 2(4), 263–277 (2007)
3. Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A survey of context modelling and reasoning techniques. Pervasive and Mobile Computing 6(2), 161–180 (2010)
4. Chen, H., Perich, F., Finin, T., Joshi, A.: SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In: Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous) (2004)
5. Cisco: Cisco visual networking index: Global mobile data traffic forecast update, 2013−2018. http://bit.ly/CiScOWP (February 2013), white paper c11-520862

6. Das, P.K., Joshi, A., Finin, T.: Energy efficient sensing for managing context and privacy on smartphones. In: First Int. Workshop on Society, Privacy and the Semantic Web - Policy and Technology (PrivOn) (2013)
7. Dentler, K., Cornet, R., ten Teije, A., de Keizer, N.: Comparison of reasoners for large ontologies in the OWL 2 EL profile. Semantic Web 2(2), 71–87 (2011)
8. Ghosh, D., Joshi, A., Finin, T., Jagtap, P.: Privacy control in smart phones using semantically rich reasoning and context modeling. In: IEEE Workshop on Semantic Computing and Security (WSCS) (2012)
9. Gu, T., Wang, X.H., Pung, H.K., Zhang, D.Q.: An ontology-based context model in intelligent environments. In: Communication Networks and Distributed Systems Modeling and Simulation Conf. (CNDS) (2004)
10. Guo, B., He, H., Yu, Z., Zhang, D., Zhou, X.: GroupMe: Supporting group formation with mobile sensing and social graph mining. In: 10th Int. Conf. on Mobile and Ubiquitous Systems: Computing, Networking, and Services (Mobiquitous) (2013)
11. Gupta, S., Joshi, A., Finin, T.: A framework for secure knowledge management in pervasive computing. In: Workshop on Secure Knowledge Management (2008)
12. Jagtap, P., Joshi, A., Finin, T., Zavala, L.: Preserving privacy in context-aware systems. In: 5th IEEE Int. Conf. on Semantic Computing (ICSC) (2011)
13. Kagal, L., Finin, T., Joshi, A., Greenspan, S.: Security and Privacy Challenges in Open and Dynamic Environments . Computer 39(6), 89–91 (2006)
14. Lane, N.D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., Campbell, A.T.: A survey of mobile phone sensing. IEEE Communications Magazine pp. 140–150 (2010)
15. Lane, N.D., Xu, Y., Lu, H., Hu, S., Choudhury, T., Campbell, A.T., Zhao, F.: Enabling large-scale human activity inference on smartphones using community similarity networks (CSN). In: 13th Int. Conf. on Ubiquitous Computing (UbiComp) (2011)
16. Pappachan, P., Yus, R., Joshi, A., Finin, T.: Rafiki: A semantic and collaborative approach to community health-care in underserved areas. In: 10th IEEE Int. Conf. on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom) (2014)
17. Ranganathan, A., Al-Muhtadi, J., Campbell, R.H.: Reasoning about uncertain contexts in pervasive computing environments. IEEE Pervasive Computing 3(2), 62–70 (2004)
18. Shvaiko, P., Euzenat, J.: Ontology matching: State of the art and future challenges. IEEE Trans. on Knowledge and Data Engineering 25(1), 158–176 (2013)
19. Wibisono, W., Zaslavsky, A.B., Ling, S.: Situation-awareness and reasoning using uncertain context in mobile peer-to-peer environments. Int. J. Pervasive Computing and Communications 9(1), 52–71 (2013)
20. Yus, R., Bobed, C., Esteban, G., Bobillo, F., Mena, E.: Android goes semantic: DL reasoners on smartphones. In: 2nd Int. Workshop on OWL Reasoner Evaluation (ORE) (2013)
21. Yus, R., Mena, E., Ilarri, S., Illarramendi, A.: SHERLOCK: Semantic management of location-based services in wireless environments. Pervasive and Mobile Computing 15(0), 87–99 (2014)
22. Zavala, L., Dharurkar, R., Jagtap, P., Finin, T., Joshi, A.: Mobile, Collaborative, Context-Aware Systems. In: AAAI Workshop on Activity Context Representation: Techniques and Languages. AAAI, AAAI Press (August 2011)
23. Zavala, L., Murukannaiah, P.K., Poosamani, N., Finin, T., Joshi, A., Rhee, I., , Singh, M.P.: Platys: from position to place-oriented mobile computing. AI Magazine 35(4), 1–9 (2014)