A semantic web model for ad hoc context-aware communities : Application to the Smart Place Scenario

Pierre Maret, Frédérique Fla Laforest, Dimitri Lanquetin

A Semantic Web Model for Ad Hoc Context-Aware Virtual Communities
Application to the Smart Place Scenario

Pierre Maret, Frédérique Laforest and Dimitri Lanquetin
Université Jean Monnet, F-42000 Saint-Etienne, France
{*pierre.maret, frederique.laforest*}*@univ-st-etienne.fr, dimitri.lanquentin@gmail.com*

Keywords: Communities, Mobility, Services, User, Semantic Web, RDF, Ontology

Abstract: In this paper, we propose a model for an open framework that allows mobile users to create and to participate to context-aware virtual communities. The model we propose and implement is a generic data model fully compliant with the semantic web data model RDF. This model is suited to let mobile end-users use, create and customize virtual communities. We combine fundamentals for a decentralized semantic web social network with context-aware virtual communities and services. Smart cities scenarios are typically targeted with this approach. It can be implemented in places like metro stations, museums, squares, cinemas, etc. to provide ad hoc context-aware information services to mobile users.

1 INTRODUCTION

A virtual community (VC) is a computer supported space where information can be shared in-between stakeholders with common interest or purpose (Preece, 2001; Koch et al., 2002). Millions of VCs exist on the Internet and they form what is called the Social Web, where people can join, leave and participate in many VCs (Bouras et al., 2005). In this paper, we address the topic of context-aware virtual communities and more specifically we present a data model and its implementation suited to let mobile end-users use but also create and customize VCs.

Some community platforms take advantage of the mobile context of their users (GPS coordinates, IDs of close devices) to provide a specific service. Examples are numerous: Personal Smart Spaces (Gallacher et al., 2012), Group Me (Guo et al., 2012), Meeting Assistant (Zenker et al., 2012; Al Ridhawi et al., 2012). They use context information to provide guidance, profile matchmaking or recommendations. Each platform provides a specific service. Users willing to use (not even to combine) various services face the so called "silo" problem and they have to register to each platform (Yeung et al., 2009). Also, these platforms don't let users define the services they need in their communities. The platforms proposing user-defined communities require programming skills. Synthesizing all these remarks, nowadays community platforms are closed platforms that do not allow to take profit to the open world of open data, communities are hardly tunable by their creators without computing skills, the sharing of profile information or knowledge between communities and with the open world is hard or even impossible.

In smart cities, users are supposed to move and use various services. So, there is need for a model and an open architecture for users to create the communities they want, with the services they want, and at the location they want (or more generally in the context they want the services to be used).

In this paper, we propose a model and a platform architecture for ad hoc context-aware communities: communities can be created by naive end-users with a selection of pre-defined semantically described services. Users get access to these communities with a single sign-in. Our model makes use of Semantic Web technologies which allows it to provide self-describing services and to offer a open door to the open linked data inand out of the virtual communities. The semantic web approach promotes open data model, machine readable data and reasoning, which let the users be supported by personal software agents to manage their personal data and social relations in respect with the chosen policy. Our platforms also uses the multi-agent paradigm to support users' interactions. These interactions take advantage of context information to help services tune their offered functionalities. The multi-agent paradigm emphasizes autonomy and cooperation of communicating entities with goals. Our design typically addresses smart cities scenarios. It can be applied in places

like metro stations, museums, public squares, cinemas, park places, etc.

We first present a state of the art related to the main features for communities: contents, user profile and services. Then we make a tour of existing community platforms. In section 3 we present our conceptual framework along the three main features from the state of the art. In section 4 we provide a description of our implementation, and conclude in section 5.

2 CONTENTS, USER PROFILE AND SERVICES IN VCS

2.1 Contents

Contents are information pieces exchanged between users within a community. The underlying data model impacts the sharing of these contents within the community platform and out of the platform: inputs, queries, exports, imports of information can be envisioned differently depending on this model. The model can be either platform dependent or open.

Most social networking websites use their own data models, specifically designed for their architecture. Moreover, the data model is rarely publicly available. Platform-dependent data formats make it nearly impossible to import/export contents, profiles and social relations (Razmerita and Firantas, 2009). It leads to the well-known isolation of social networks.

Platform independent data models are public. Since community platforms run on the Internet and in a Web environment, norms or standards from these fields are the best candidates to share models. SIOC (Breslin et al., 2005; Bojars et al., 2008) has been proposed for the representation of social data. It is founded on the Semantic Web technologies. The Semantic Web approach (Berners-Lee et al., 2001) aims at representing data in terms of RDF[1] triples: (*term relation value*) or (*term1 relation term2*). It is a user- and machine-readable model. Each item of a triplet can be described anywhere on the Internet (URI, Uniform Resource Identifier). This builds the so-called linked data. Sesame (Broekstra et al., 2002) and Jena are Java frameworks used to store RDF triples. Reasoning processes can be launched to infer facts, especially in conjunction with formal description of domains (ontologies) which are also written with RDF (more specifically using RDF-based languages: OWL and RDFS). SIOC is written in the RDF/OWL language. Its use provides models and techniques for a platform independent data model.

[1] http://www.w3.org/RDF/

2.2 User Profile

Some platforms (Gallacher et al., 2012; Ganti et al., 2012; Guo et al., 2012) propose to enrich dynamically the user profile with context information: physiological parameters, GPS coordinates, nearby devices, etc. The management of several profiles is a complex task for end-users. Some works have proposed to federate different profiles (Mitchell-Wong et al., 2007). However, all characteristics cannot be federated and, for average users, keeping profiles safe and up to date quickly becomes an issue. This is crucial because user profiles contain personal information. A complex management of profiles raises privacy issues: users may not adequately control data sharing.

A unique profile used on several community platforms eases its management and allows the user to better control the access rights to personal data. The profile can be stored either on a remote system or on the user's side. Several proposals can be found in the literature (Seong et al., 2010; Koch and Worndl, 2001; Grzonkowski et al., 2009). The most advanced model for profile description is FOAF. It is described as an ontology, also written in the RDF/OWL language.

2.3 Services in VCs

A virtual community offers services to its participants to communicate *i.e.* create, modify, read, share information. It can take the form of forum, blog, chat, private messaging, walls... Services are key components of communities as they can strongly influence participation and adoption. Roles are used to define different rights of users: a forum can have a moderator, members, experts, etc. Other types of services can be found such as services related to the location, to information retrieval, to recommendation, etc.

The services offered on a platform are determined by the developers or the administrators. Also services are by definition not available out of the platform and they are not exportable: they are made for a given custom architecture and data model. Only some VC platforms allow users to create customized communities, but they require programming skills (see 2.4).

When the number of services increases, users should receive recommendations to select the most relevant ones regarding their preferences, social relations, and context (Schubert and Koch, 2003). In (Broens et al., 2004; Meyffret et al., 2010), ontologies for the descriptions of services, for the user's preferences and for context information are used to improve service discovery and selection. In (Sinner et al., 2004), a framework is used to match semantic user profiles with semantic descriptions of location-

based services for recommendation.

2.4 Contents, User Profile and Services in Existing VCs Systems

Popular social networking websites such as Facebook use proprietary models. User profiles are stored on the proprietary platform which is based on a centralized architecture. Different approaches have been proposed[2]. Diaspora is a software environment for distributing the components of the social network at the users' sides. Evri has platform-dependent profiles and services, however it uses ontologies for the semantic annotation of contents. dgFOAF (Schwagereit et al., 2010) proposes users to be registered in a unique community and to launch services from different platforms. Elgg and Liferay allow the development of community platforms. Services can be added as plug-ins (or portlets), making community creation and upgrading customizable. However, they require programming skills.

The multi-agent approach has been also used for VCs (Gunasekera et al., 2012; Aguero et al., 2012). Koch (Koch and Lacher, 2000) introduces agencies to separate user profile and community participation (*i.e.* use of community services). Guidelines are made available for people with programming skills wanting to develop a community platform, with no restriction on contents or services representation. (Fahad et al., 2012) is a community platform based on the multi-agent framework Cartago [3]. End-users can create and join communities, and they can delegate these tasks to autonomous agents. The communities are limited to four services: mailbox, forum, information dispatcher and personal box.

Properties of community platforms are summarized in table 1. None of these let end-users create and customize their VCs in selecting services and associating them to context information. Few platforms use open (semantic web) models.

3 A DATA MODEL FOR AD-HOC CONTEXT-AWARE VIRTUAL COMMUNITIES

In this section, we propose our data model to develop the framework for ad hoc context-aware communities. The model relies on RDF, and consists of three main elements: (i) *A user profile* based on the FOAF model and encompassing also user's context description, (ii) *A community model* that extends SIOC with communities-related properties, and (iii) *A service description model* that allows the use of on-the-shelf services.

Platforms	(1)	(2)	(3)
Facebook, Twitter	no	no	developer's choice
Diaspora	no	yes	developer's choice
Evri	yes	no	developer's choice
dgFOAF	yes	yes	developer's choice
Elgg, Liferay	no	no	developers' plug-ins
Fahad et al.	no	yes	predefined services
Koch et al.	no	yes	developer's choice

Table 1: Properties of community platforms.
(1)has an open semantic data model; (2)allows multi-platform; (3) implementation of services

3.1 User Profile

As we have seen in section 2.2, a unique profile stored at user's side is desirable. We use and extend the FOAF ontology. The model (Fig. 1(a)) has the following properties: *foaf:interest* (a topic about which the actor wants to share, obtain, create information), *foaf:friend* (another actor known by the user), *hasAvatar* (a delegate of the actor in the systemhosted by an electronic device of the user), *hasGoal* (goal to be achieved by the avatar, see 3.4). Properties foaf:friend and foaf:interest are taken from the FOAF ontology. The Actor class refers to both the foaf:Person and the UserAccount class in SIOC. The SIOC:Avatar class has no relationship with our Avatar class.

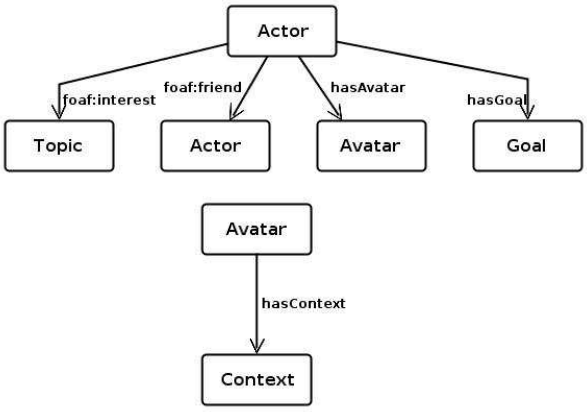

Figure 1: Actor model (a), Avatar model (b)

In addition, avatars have the property *hasContext*, which is the capability of the device hosting the avatar to deliver context data (Fig. 1(b)). We define context

[2]http://diasporaproject.org/, http://www.evri.com/, http://elgg.org/, http://liferay.com/

[3]http://cartago.sourceforge.net

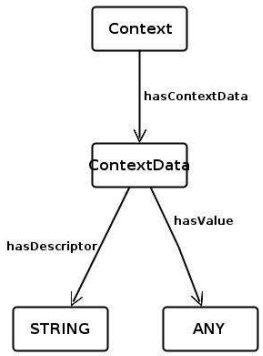

Figure 2: Context model

data as a simple set of pairs (descriptor, value). These pairs are used to represent current information sensed, calculated or known by the device hosting an avatar. Context data may relate to location but also to battery state, network connections, user's behavior (through an accelerometer), etc. The property *hasContextData* is the piece of information describing the state of the device hosting the avatar or a status of the actor (Fig. 2). It contains *hasDescriptor* (a string describing the information) and *hasValue* (context data value).

3.2 Content and Communities

Contents consist of pieces of information coded in RDF format with the properties *hasAuthor*, *hasTopic*, *type* and *value* which associates a value of a RDF class defined as a URI (Fig. 3(a)). The sharing of contents happens in the frame of communities. We define the Community ontology as an extension of the SIOC ontology (Fig. 3(b)). It is composed of *hasOwner*, *hasTopic* (contents used as subject of the community), *hasMember* (actors), *hasService* (see 3.3), *hasRole* (see 3.4) and *hasContent* (contents shared within the community). SIOC also provides the notion of topic with a dedicated property that usually refer to a semantic web resource.

3.3 Services

We define a service as a set of operations. For instance, a blackboard is a service, with the operations write, read and delete. Services in our framework are related to the management of contents. They are implemented to create and get contents from communities (*hasContent* property). As all communities are described using the data model defined above, any service can be used in communities. Services are described using the ontology given in Fig. 4. It contains the name of the service, its description and operations.

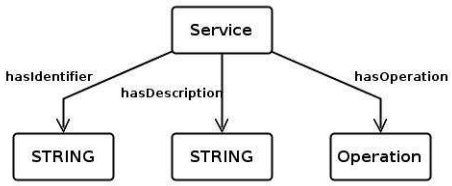

Figure 4: Service model

The Service class of SIOC has been adapted to define our Service class.

Properties of an operation are described in Fig. 5(a). Notice that the *hasRequiredContext* property introduces the context of the avatars (see 3.1). This is used to introduce the context-aware property of operations. The property *hasCommand* relates to the exchanges of information: Get, Put, Delete or Post; it uses *hasInputParameter* and *hasOutputParameter* A parameter (Fig. 5(b)) has a description, a type (RDF class) and constraints applied on its value (RDF triple). The *hasMultiProperty* is used to indicate multivalued properties.

3.4 Goals and Roles

Goals are defined by actors. They are part of their personal data and consist of sets of operations that avatars will execute (Fig. 6(a)). The goal model uses part of the operation's model defined above to ease service discovery and selection. Roles define the rights to execute operations in the frame of a community. A role is composed of a set of actors and a set of operations (Fig. 6(b)). The notion of role also exists in the SIOC ontology with the class Role that can be referenced by a UserAccount.

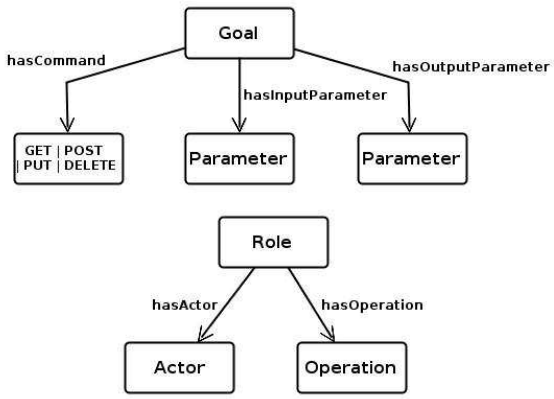

Figure 6: Goal model (a) and Role model (b)

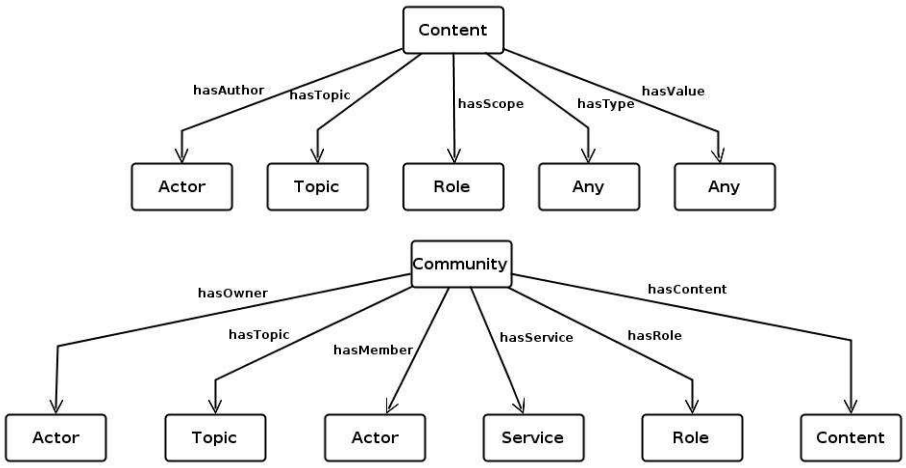

Figure 3: Content (a) and Community (b) models

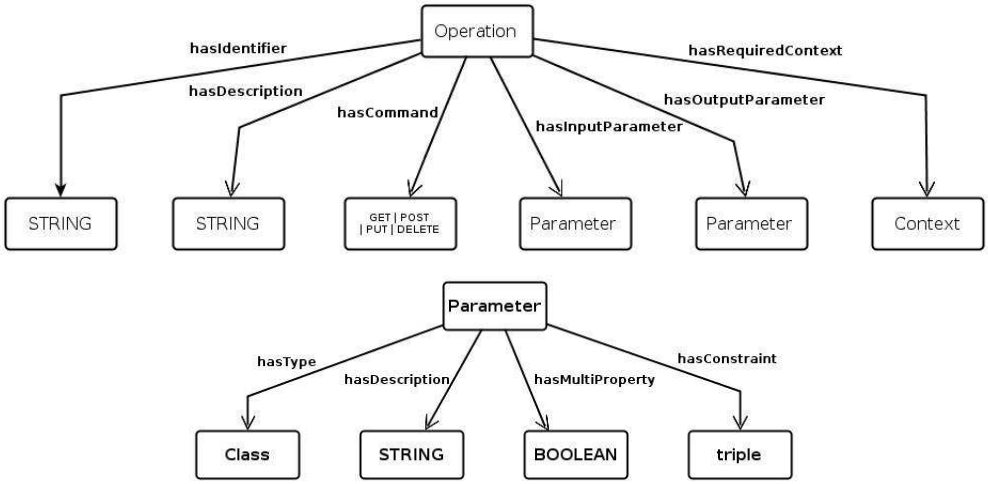

Figure 5: Operation model (a) and Operation parameter model (b)

4 FRAMEWORK IMPLEMENTATION

In the previous section, we have presented the data model of our framework. It allows users, communities and services modeling, and more specifically context-aware operations within communities. It is based on RDF principles and can be easily extended or adjusted. In this section, we present the implementation of this model and we show how it allows end-users to easily create their customized communities with different services and for dedicated contexts. The compliance with the RDF principles is kept with the framework implementation.

4.1 Main Implementation Choices

Our implementation combines the semantic web technologies with the multi-agent approach. Indeed, agents (users, avatars) as well as information and services are distributed in different concrete or virtual (*i.e.* computer supported) locations. Furthermore agents interact with others and share information in the frame of the communities to achieve user's goals.

We use the multi-agent platform Cartago to implement our system. In Cartago, agents interact with other agents but also with the environment through the *artifact* abstraction. We implement data store, avatars, communities and services with this abstraction. Through artifacts, agents can write/read data, get context values, create and participate to communities and execute operations. Cartago proposes also the

concept of *workspace* to group artifacts and agents. We use this concept to group communities based on context values (for instance the location). We call *Space of Communities* (SoC) a set of communities hosted in a same workspace. Distributed communities can also be realized since communication in-between workspaces is possible.

As previously mentioned, we use RDF triples to model contents within communities, users' profiles, and services. This RDF implementation conforms to open linked data models. We use Sesame repositories (Broekstra et al., 2002) within *artifacts* to store and access these triples. Services are built on top of Web services. Details of the framework architecture are described in the next subsections and summarized in figure 7.

4.2 Implementation of Users Profiles

The user profiles are implemented as described in the data model (see 3.1) under the form of RDF triples. Each user's avatar is composed of an artifact (Profile artifact) and 3 agents (Profilor, Space Detector, Participator). The *Profile artifact* is used for accessing a Sesame repository which contains triples about context information, information topics and friends as defined in 3.1. The *Profilor agent* lets users manage their profiles. The *Space Detector agent* is a contextual detector of SoCs. The *Participator agent* is created when a SoC is detected. It allows the user to join or create communities: the agent filters the information received from the SoCs, depending on the user's topics of interest, context and goals and notifies the user. This process is privacy-aware since no personal data is transmitted to the SoC. Notice that as the user profile is unique, avatars of a single user interact together to merge data and keep up to date.

4.3 Implementation of Communities and their Services

As stated before, Cartago artifacts are used to implement communities. Once a community is created, a community artifact allows the access to a Sesame repository. It stores the data of this community: shared contents, roles, services (see 3.2). A dedicated SoC artifact manages the list of communities and the list of services the SoC holds.

Services interfaces are also implemented as artifacts. They provide a set of methods for querying (using SPARQL), adding, deleting triples in the community repository. The availability of a given service in a community is decided by the community creator.

4.4 Smart City Scenario and Discussion

Our system can be used for the following scenarios. A city has implemented in the most populated quarters the ad hoc context-aware community platform and has settled several *spaces of communities* with different web services. A ride sharing service and a question/answer service are among these. Operations of the ride sharing service are: declare an offer, search for daily rides, and book for a ride. The question/answer service has two main operations: search a query and get answers; and read a query and give an answer. This second service is implemented at the train station.

As a committed citizen, Alice wants to develop daily ride sharing in her close area. She has learned the existence of the community platform and decides to register and to create her *avatar* (stored on her device). She creates a community with "ride sharing" as *topic*, the street name as the *context* for launching the services, and she selects *operations* from the ride sharing service. John is searching for "car ride" on the platform. Since he leaves close to Alice (same context), and thanks to the semantic analysis (same content), the platform proposes him to join Alice's community.

Bob and Colin have joined the platform. They often wait in the train station and have learnt that the train company has created a question/answer community. Bob loves winning points every day in answering questions asked by other participants. Colin loves asking questions about subjects he wants to learn about.

The scenarios illustrate the kind of implementations our data model can support for smart city applications. Users can manage their profiles on their mobile devices, their avatars can collect context data, and pieces of information can be gathered or delivered automatically following their goals. Spaces of Communities (with their artifacts, data stores and services) can be hosted by servers in specific places such as museums, cinemas, stores, campus, points of interest... Communities with ad hoc services can be created by users with no specific knowledge in computer science, thanks to the use of the semantic web technologies. Users can join communities and thus receive or deliver information.

Compared to other approaches, our approach proposes a user profile which is unique and which contains context values from the user's avatar. It can be used in all Spaces of Communities. The models for user profile, context and data are open and compliant with the Semantic Web technologies. Knowledge items are RDF triples contained into URIs. This

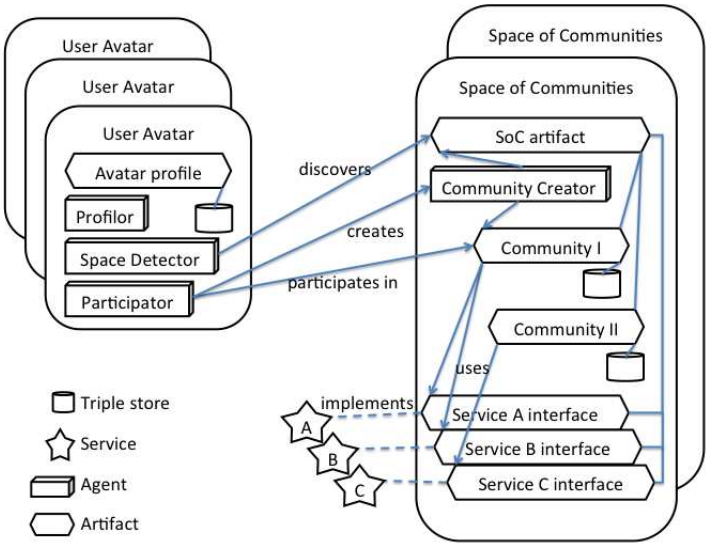

Figure 7: Framework architecture

opens our system to an unlimited amount of knowledge, as it follows the open linked data principles and technologies. Following such standards also allows to include other value added services ; for example reasoning on data can improve the user's experience with better recommendations. Finally, services can be added into Spaces of Communities and they can be easily included by end-users in the communities they create. No programming skill is required : the use of standards for the semantic description of services allows their inclusion with a high level of interaction with end users. Different types of communities can be created, using different services according to the communities needs. The semantic description of the avatar context also allows the use of such information by services ; they can adapt their behavior to the end-user context of usage.

5 CONCLUSION

The design of ad hoc context-aware communities typically addresses smart cities scenarios. We have proposed a model and an implementation which rely on the Semantic Web approach. Our approach can be useful in many different places and for many scenarios. Our data model can be easily extended to suit specific needs, and various services can be used by end-users. Cities, shops, malls, museum, cinemas should be interested by the approach. In the near future, our next steps will consist of building a more appropriate and illustrative user interface for end-user, and of developing new services which will illustrate different kinds of communities that could be initiated by users in different places.

REFERENCES

Aguero, J., Rebollo, M., Carrascosa, C., and Julin, V. (2012). Developing pervasive systems as service-oriented multi-agent systems. In Snac, P., Ott, M., and Seneviratne, A., editors, *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, volume 73 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 78–89. Springer.

Al Ridhawi, Y., Al Ridhawi, I., Bruno, L., and Karmouch, A. (2012). Policy-based personalized context dissemination for location-aware services. In Senac, P., Ott, M., and Seneviratne, A., editors, *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, volume 73 of *Lecture Notes of the Inst. for Comp. Sc., Social Informatics and Telecommunications Eng.*, pages 366–371. Springer.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5):34–43.

Bojars, U., Passant, A., Cyganiak, R., and Breslin, J. (2008). Weaving SIOC into the Web of Linked Data. In *Linked Data on the Web (LDOW) workshop, in conjunction with WWW conference*.

Bouras, C., Igglesis, V., Kapoulas, V., and Tsiatsos, T. (2005). A web-based virtual community. *Int. J. of Web Based Communities*, 1(2):127.

Breslin, J., Harth, A., Bojars, U., and Decker, S. (2005). To-

wards Semantically-Interlinked Online Communities. In *2nd Eur. Semantic Web Conference*.

Broekstra, J., Kampman, A., and Harmelen, F. V. (2002). Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In *International Semantic Web Conference - ISWC*, pages 54–68.

Broens, T., Pokraev, S., Sinderen, M. V., Koolwaaij, J., and P (2004). Context-aware, ontology-based service discovery. *Ambient Intelligence*, pages 72–83.

Fahad, M., Boissier, O., Maret, P., and Gravier, C. (2012). Smart places: multi-agent based virtual community management system. In *Proc. of the 4th Int. Workshop on Web Intelligence and Communities*, New York, USA. ACM.

Gallacher, S., Papadopoulou, E., Taylor, N., Williams, M., and Blackmun, F. (2012). Linking between personal smart spaces. In Senac, P., Ott, M., and Seneviratne, A., editors, *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, volume 73 of *Lecture Notes of the Inst. for Comp. Sc., Social Informatics and Telecommunications Eng.*, pages 401–408. Springer.

Ganti, R., Mohomed, I., Raghavendra, R., and Ranganathan, A. (2012). Analysis of data from a taxi cab participatory sensor network. In Puiatti, A. and Gu, T., editors, *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, volume 104 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 197–208. Springer.

Grzonkowski, S., Gzella, A., Kruk, S. R., Breslin, J. G., Woroniecki, T., and Dobrzanski, J. (2009). Sharing information across community portals with FOAF-Realm. *Int. J. of Web Based Communities*, 5(3):351.

Gunasekera, K., Krishnaswamy, S., Loke, S., and Zaslavsky, A. (2012). Adaptation support for agent based pervasive systems. In Snac, P., Ott, M., and Seneviratne, A., editors, *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, volume 73 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 90–101. Springer.

Guo, B., He, H., Yu, Z., Zhang, D., and Zhou, X. (2012). GroupMe : supporting group formation with mobile sensing and social graph mining. In *9th International Conference on Mobile and Ubiquitous Systems : Computing, Networking and Services*, Beijing, Chine.

Koch, M., Groh, G., and Hillebrand, C. (2002). Mobile Communities Extending Online Communities into the Real World. In *Wirtschaftsinformatik*, pages 7–17.

Koch, M. and Lacher, M. (2000). Integrating community services-a common infrastructure proposal. In *4th Int. Conf. on Knowledge-Based Intelligent Eng. Syst. and Allied Techn. Proceedings*, volume 1, pages 56–59. IEEE.

Koch, M. and Worndl, W. (2001). Community Support and Identity Management. In *European Conf. on Comp. Supported Cooperative Work*, pages 319–338.

Meyffret, S., Laforest, F., and Médini, L. (2010). Semantic, Social & Context-Aware Registry for Services. In *Ubimob*.

Mitchell-Wong, J., Kowalczyk, R., Roshelova, A., Joy, B., and Tsai, H. (2007). Opensocial: From social networks to social ecosystem. In *Digital EcoSystems and Technologies Conference, 2007. DEST '07. Inaugural IEEE-IES*, pages 361–366.

Preece, J. (2001). Sociability and usability in online communities: Determining and measuring success. *Behaviour & Information Technology*, 20(5):347–356.

Razmerita, L. and Firantas, R. (2009). Towards a New Generation of Social Networks: Merging Social Web with Semantic Web. In *Proceedings of I-KNOW 09 and I-SEMANTICS 09*, number September, pages 412–423.

Schubert, P. and Koch, M. (2003). Collaboration platforms for virtual student communities. In *36th Annual Hawaii Int. Conf. on Syst. Sc.*, page 10 pp. IEEE.

Schwagereit, F., Scherp, A., and Staab, S. (2010). Representing Distributed Groups with dgFOAF. In *European Semantic Web Symposium*, pages 181–195.

Seong, S.-W., Seo, J., Nasielski, M., Sengupta, D., Hangal, S., Teh, S. K., Chu, R., Dodson, B., and Lam, M. S. (2010). PrPl: A Decentralized Social Networking Infrastructure. In *Proc. 1st ACM Workshop on Mobile Cloud Comp. & Services Social Networks and Beyond - MCS '10*, pages 1–8, New York. ACM Press.

Sinner, A., Kleemann, T., and von Hessling, A. (2004). Semantic user profiles and their applications in a mobile environment. *Art. Intell. in Mobile Systems*.

Yeung, C.-m. A., Liccardi, I., Lu, K., Seneviratne, O., and Berners-lee, T. (2009). Decentralization: The future of online social networking. In *W3C Workshop on the Future of Social Networking Position Papers*.

Zenker, B., Pyatkovka, V., and Drabek, C. (2012). Towards meeting assistance: An integrated appointment assistance and group pedestrian navigation system. In Puiatti, A. and Gu, T., editors, *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, volume 104 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 338–349. Springer.