An Ontology Based System for Social Networking for Health Application Support

Ahmed Loubiri, Abdel Obaid[1], Fatiha Sadat
LATECE Lab., Department of Computer Science
University of Quebec at Montreal,
Montreal, Canada
[1]obaid.abdel@uqam.ca

Abstract—**This work was carried out in the context of information modeling of social networks information and its use in the area of medical applications. The ontology of a social network (namely *Facebook*) is determined. This modeling together with specific ontology in the health domain, allowed us to build an ontology-based system for context dependant application.**

Keywords-component: Context-dependant applications; ontology; social networks; Facebook; medical application.

I. Introduction

Context-dependent systems gather information about the user's context from various sources in order to adapt their services. Context-aware applications can use this information to improve the human-computer interaction by reducing human intervention. They also allow applications to exploit user's surrounding information and adapt their behaviour accordingly [1].

It is important to define and model the user's context to provide context information to a context-dependent system in a rigorous and formal manner. This requires developing models capable of representing the different components of the context and their relationships.

Various architectures have been proposed in the literature [2] [3] [4] [5] [6]. However, we can identify some of their shortcomings with respect to the modeling and the use of context data:

- The lack of formalism that allows rigorous understanding and processing of context information.
- Disuse the large amount of contextual information that already exists in environments of online social networks e.g. Facebook.
- The lack of mechanisms for the integration of knowledge representations that can be extracted from these social networks in the processing of context information
- Disregard the linkage between online information and useful applications such as medical applications. Several applications can benefit from patient information that already exists in his/her social network profile data.

In this paper, we describe the use of context modeling for an application in the medical field. It exploits user's biomedical data to characterize his/her medical context. We use the ontology to enrich the Facebook context model with data extracted from this environment.

II. The Architecture

The establishment of context-dependant systems led us to come up with an architecture and middleware that includes independent modules capable of acquiring information from different sources and process them to derive the context and to provide it to the appropriate services.

We present an infrastructure composed of three layers: sensor layer, middleware layer and service layer.

A. The sensor layer

This layer contains all data sources that can provide useful context information. We can distinguish three types of sensors [7]: a) physical sensors that generally provide raw data, b) virtual sensors that gather data primarily through web services, c) logical sensors that combine both types of sensors to derive information on the context using logic expressions.

In our project, we use sensors to measure physiological data such as body temperature, blood pressure and blood glucose level. These small sensors are attached components to different parts of the body and allow communication between the user and a remote system [8] via a mobile device (Fig. 1).

This configuration allowed us to form a network of body sensors to collect information in a centralized manner. The sensors are interconnected through wireless channels such as ZigBee, Bluetooth or WiFi depending on the desired channel quality. These channels are also used for data transmission to the mobile device.

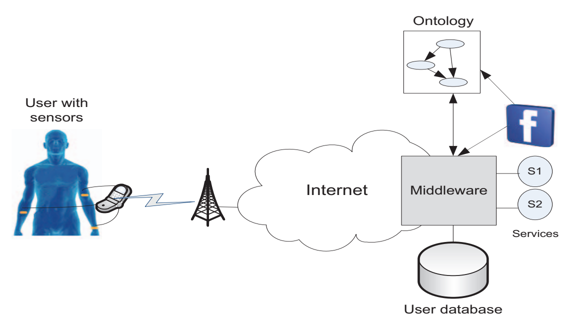

Fig. 1. Global system architecture

B. The middleware

Our middleware is structured in layers that divide the processing into sub-steps depending on the task at hand (Fig. 2). The layering allows organizing the process of information management and the generation of context.

1) The *collection* layer is responsible for retrieving data from sensors. It provides a set of interfaces and abstract methods that use the drivers for the physical sensors as well as their APIs and implements query processing capability to retrieve data. This layer is bound to a component named *Context Matcher* so that it can detect all sensors and collect appropriate data.

2) The *filter layer* is used to filter out data. Captured data may either contain measurement errors or not satisfy specific requirements (e.g. inapt to specific intervals). Sensors can provide erroneous or unnecessary for the services context. This layer is linked to a database that contains all the constraints to be applied to a given data type. These constraints can be predefined and may later be changed or updated manually or automatically.

3) The *data processing layer* is responsible of processing data retrieved from the previous layer. These received data are derived from various sources and are heterogeneous. They represent numerical values with various measuring units. They may not be appropriate or understandable for non-context-dependent services. This layer is responsible for processing the data captured and presenting it in a shareable format and can be used as an input for specific inference rules. An example of such format is the OWL[1] language expressions.

4) The *inference module* is responsible for inferring the context from data retrieved from the previous layer. It represents information in a higher level of abstraction. The low-level information derived from captured raw data is often insufficient, which we refer to as *low-level context*. It is combined with other captured data in order to infer more relevant and valuable context information to result in a *high-level context*. This layer is also used to query the context information database and to check the consistency of context information [9]. The inference layer encapsulates various modules to enable the execution of these tasks:

- The *Knowledge base* provides the tools of storing, querying and modifying knowledge about context. It contains ontology of a specific context domain and its instances.
- The *Reasoning Module* is responsible for the inference of context information from low-level context information to a higher level. This module uses a first order logic formalism [10].

5) The Context Matcher is responsible for locating and managing context sources. It uses a mechanism for mapping context sources with the context service needs.

[1] http://www.w3.org/TR/owl-features/

C. Context Services

These are the various services that are context dependant. These services need information from the context in order to adapt to the current situation of the user and his/her environment.

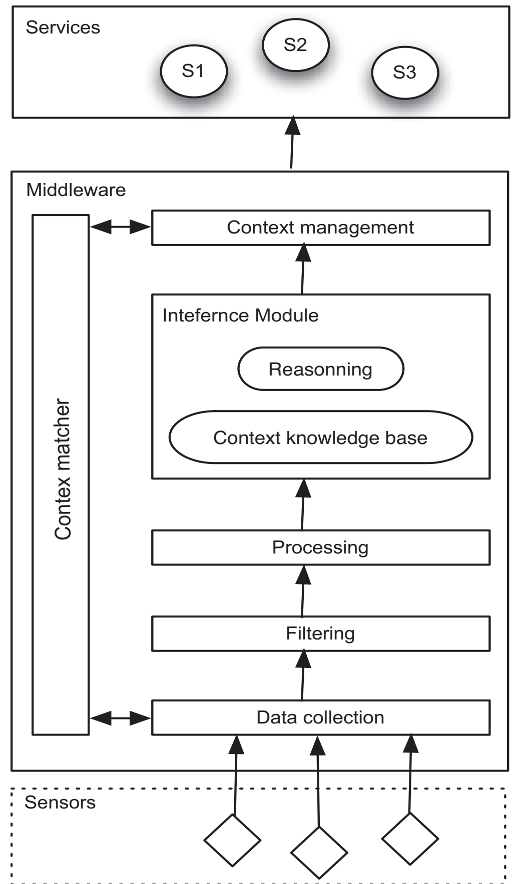

Fig. 2. Middleware architecture

III. Modeling The Facebook Ontology

We analyzed an online social network (namely Facebook) in order to extract context information from user's profile and personal data. This information is used to enrich our context model. The various features of Facebook offer a rich source of personal data and provide relevant context information such as links, events, location, online presence, user's personality and preferences, etc.

We represented the Facebook model as an ontology. We have defined a global model to represent different internal objects and their relationship. For editing and handling the ontology, we used the Protégé[2]system and OWL language.

We also used the *Facebook Graph* API[3] to develop the ontological model of this network. This API allowed us to define objects and relationships as well as the set of properties for each

[2] http://protege.stanford.edu
[3] https://developers.facebook.com/docs/reference/api

object. Generated objects, properties and relations are shown in Tables 1 and 2.

TABLE I. FACEBOOK OBJECTS.

Object	Description
User	Central element of *Facebook* structure.
Account	User's pages and applications.
Page	Follower's page.
Album	Photos posted by a user.
Application	Applications developed on *Facebook*.
Check-In	User current location on his/her wall.
Event	Programmed events.
Home	Main page through which the user may visualize its friend's activities.
Friendlists	List of user's friends.
Friendrequest	Friend requests sent by the user.
Group	Group of persons sharing a common interest.
Feed	User wall containing registrations, received requests, etc.
Inbox	Incoming message box.
Outbox	Outgoing message box.
Note	User personal articles.
Notification	Notifications received by a user regarding an activity in which he/she is involved.
Post	Inscriptions done by a user on his/her wall or a friend's wall. This could be a photo, a video, a text message or a link.
Photo	Photos posted by a user.
Video	Videos posted by a user.
Status	Text messages posted by the user.
Link	Links to a site or an external Facebook object posted by the user.

TABLE II. FACEBOOK OBJECT PROPERTIES.

Objet	Properties
User	id, description, links, location, name, privacy, type, category, biography, birthday, gender, user name, address email, education, city of birth, interests, citations, political orientation, religion, language.
Page	id, name, category, link, administrator, location, telephone.
Photo album	id, description, link, location, name, privacy, type.
Application	id, category, company, name, description.
Location marking	id, location, markings, creation date.
Event	id, name, description, location, creators, visibility, start time, end time.
List of friends	id, name, type.
Group	id, name, description, link, version, privacy.
Note	id, message, subject, creation date.
Photo	id, name, source, creation date, height, width, update time, tags.
Video	id, name, description, creation date, modification time, tags.
Status	id, message, creation date, markings.
Link	id, URL, creation date, message.

Analysis of objects, their properties and the relationships among them allowed us to model a Facebook ontology shown in Fig.3. This figure is a reproduction of the ontological model obtained by using Protégé. We do not set the properties of different classes to ease the diagram. We specified the entities and relationships between them to describe the context for our application. The context model covers the functional requirements of the system.

IV. CONTEXT MODELING AND REASONING

Context modeling is "the specification of all entities and relations between these entities which are needed to describe the context as a whole" [11]. For the modeling of the context, we used the CONON ontology [12] as a high-level ontology. We extended this ontology with concepts related to the medical field. We integrated the obtained ontology with the one we developed to model the elements of Facebook in order to have a comprehensive model capable of supporting the inference rules.

From the CONON high-level ontology (O1), we added concepts from the medical field and formed another ontology (O2). This ontology is coupled with the ontology of Facebook (O3) in order to have our context model (O4). This allowed the deduction of context by inference (Fig.4).

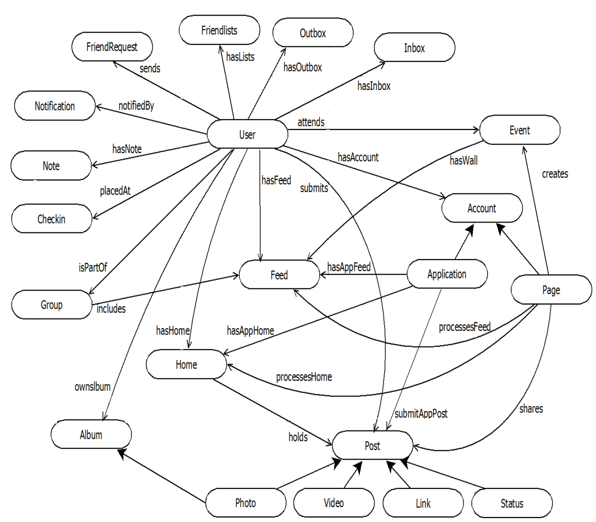

Fig. 3. Facebook ontology.

A. Step A: High Level Ontology

The CONON high-level ontology (Fig.5) is a generic model that can be adapted to a specific area. We used this model and simplified its structure. We opted for a hybrid method as a technique for designing ontology of context. In order to fulfill application requirements, we added a new generic concept named *Time* to represent the date and time. The *happensAt* relation relates this concept to the Activity concept.

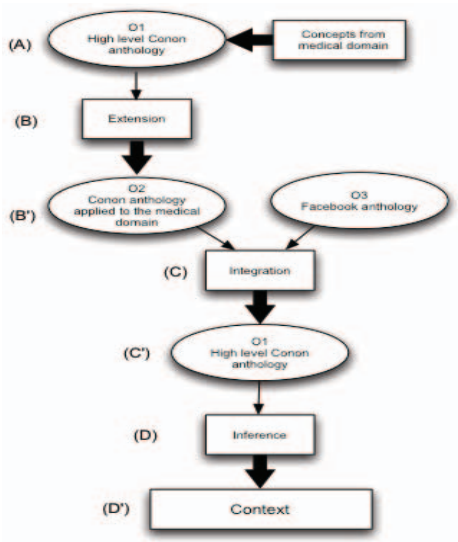

Fig. 4. Ontology construction approach.

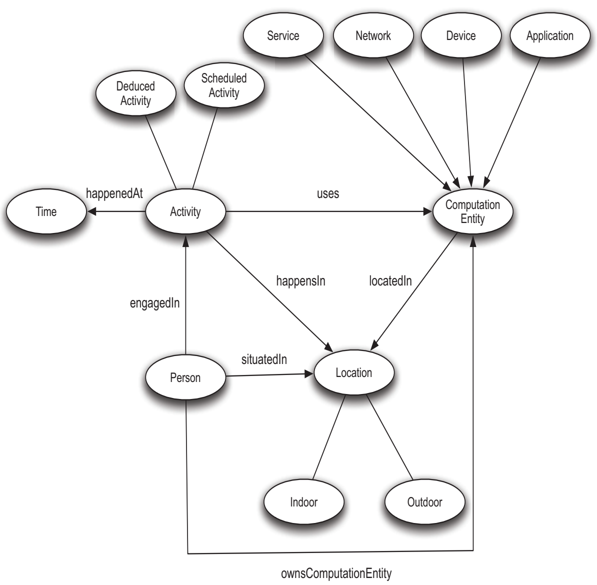

Fig. 5. CONON ontology.

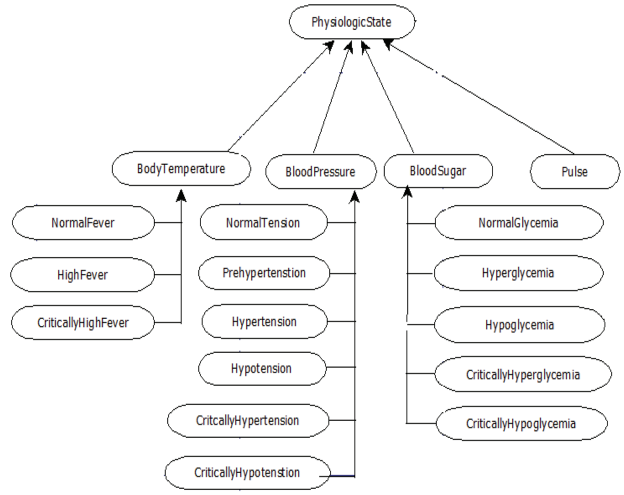

Fig. 6. A subset of medical ontology.

B. Step B: Extension of the high-level ontology with related concepts in the medical field

We developed our context model by extending this high-level ontology with concepts related to the medical field. We also used a thesaurus of medical vocabulary, namely the *Medical Subject Headings of the U.S. National Library of Medicine*. The data to be captured are body temperature, heart pulse, blood sugar level and blood pressure. We may be able to add other data in the future. We grouped them as a super-concept named *Physiological state,* which may contain other categories. We defined a ranking of each state under different measurement scenarios (Fig. 5).

The relationships among these concepts have been modeled. A medical device captures the *Physiological state* (i.e. using the *assessedBy* relation. A *person* has an *age* and *gender* categories as well as a *physiological state*.

Fig. 6 shows the ontology resulting from the adaptation of high-level ontology to the needs for our application.

C. Step C: Integration of the ontology of context and ontology Facebook

We coupled the ontology for Facebook with our ontology of the medical context. The coupling between two ontologies consists in finding their relationship so that they represent a single coherent ontology [13].

We did this integration manually. The resulting ontology (C') is shown in Fig.7.

We added a super-class that includes all the online profiles of a user, *OnlineSocialContext* that is connected to *Person*. The choice to add this entity is meant to facilitate any future expansion by other online networks.

We compared each concept in the two ontologies in order to find potential matches. For example, the *Event* class of Facebook ontology (O3) is a subclass of *ScheduledActivity*of the medical ontology (O2). It represents an event that is considered as a scheduled activity on Facebook. *Checkin* (O3) and *Location* (O2) are also related by the *ProceedsAt* relation.

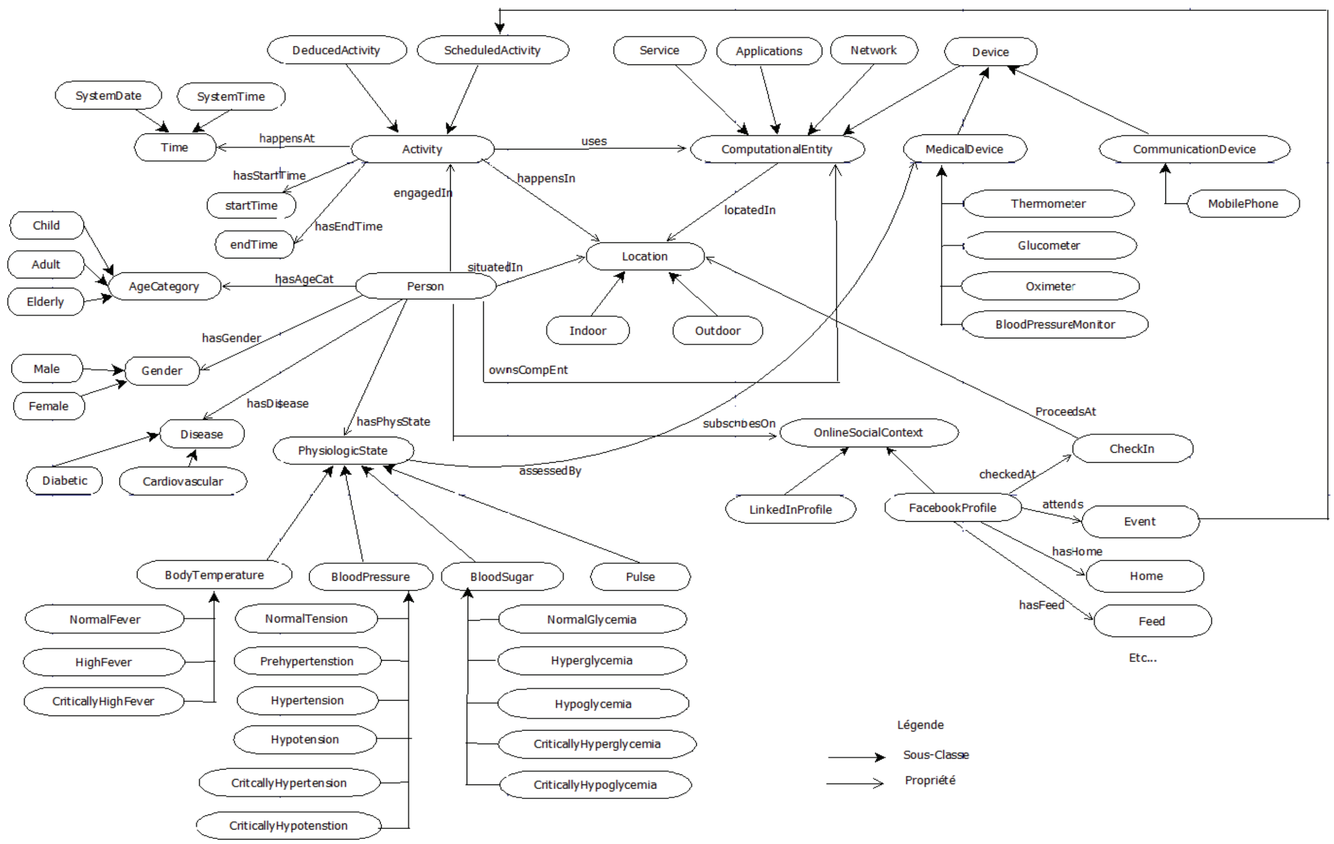

Fig. 7. Combined ontology.

D. Step D: Reasoning and Inference

The choice of the ontological approach in the context modeling was motivated by the possibility of applying the reasoning on the model. The reasoner can act as a classifier by inferring the class hierarchy. It is responsible for checking the consistency of the ontology. Using the reasoner permits to infer a high-level context from a set of collected raw data (low-level context). For instance, to infer the state of hyperglycaemia for a person, we need a set of conditions to be satisfied. These conditions are defined by a set of sensor data (age, gender, blood sugar, etc.).

We used the ontology implemented on Protegé v. 4.1.0, to define the inference rules. We also used the SWRL language and rule editor provided by Protégé and *RacerProTG*, a version of *RacerPro*.

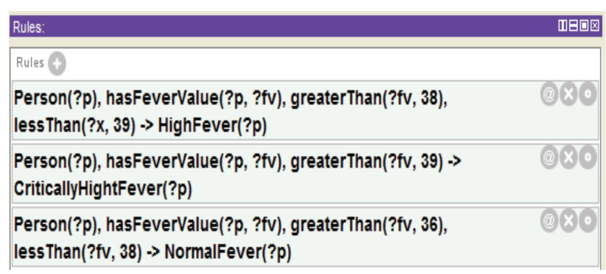

Fig. 8. Rules defining the body temperature levels on the user.

Normal values for physiological parameters vary from one category to another. We then grouped individuals (*Person*) by *age* category using *Child*, *Adult* and *Elderly*.

The example below shows the classification of body temperature. This temperature does not depend on *gender* or *age* categories of *person*. *Normal* body temperature (*NormalFever*), high (*HighFever*) or very high (*CriticallyHighFever*) is derived as follows:

- If a person has a body temperature higher than 39 degrees, then it is very high.
- If a person has a temperature between 38 and 39 degrees, then it is high.

- If a person has a body temperature lower than 38 degrees, then it is normal.

The screenshot in Fig. 8 shows the rules that can be inferred from the *physiological state* of the person to react to this context.

Unlike body temperature, blood sugar depends on the state health of the person (*normal person*, *person with diabetes, etc.*), *gender*, *age* and time of blood sampling (e.g.. after an overnight fast and the evening after a meal). Other inference rules were implemented according to these parameters.

V. SIMULATION

We performed simulations on *Protegé* to ensure the correct operation of the inference rules. We populated the ontology with individuals (*Individuals* category), defined the properties of these individuals and launched the inference engine.

We defined a person (*Ahmed*) with the following properties: *Age = 20 years*; *Gender = male*; *body temperature = 40*.

We then launched the inference engine on *RacerPro*. We could verify the classification of the individual properties as it was assigned. We conducted this verification by the new description of the individual after inference. Fig. 9 shows a screenshot of Protégé of the individual *Ahmed* after the inference. *Ahmed* is a field under *Adult*, *Male* and *CriticallyHightFever* (very high body temperature) categories.

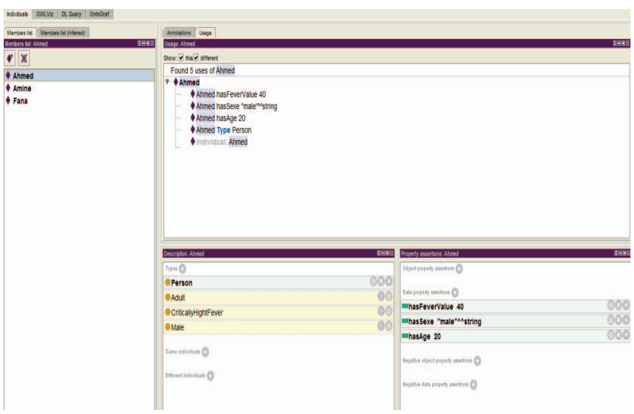

Fig. 9. Inference for an individual.

The simulations showed that the inference gives us correct results and that our model is consistent with properly defined and implemented inference rules.

Other rules were developed in the same manner in order to have a model that covers all possible cases for each physiological state.

However, in the case of a real application of our model, the population of the ontology is done automatically and the definition of the properties is acquired either through data extraction from Facebook or through received information from sensors.

VI. CONCLUSION

Social networking websites have become an incredible influential power since the advent of the Internet. We have proposed an ontology-based model that can mine social network data for context-dependant system such as a health application. The proposed ontology model enriches the Facebook context model with data extracted from this environment as well as from received information from sensors.

REFERENCES

[1] Z.Zainol and K. Nakat, "Generic context ontology modelling: A review and framework", in IEEE 2nd International Conference on Computer Technology and Development (ICCTD 2010), Piscataway, NJ, pp. 126-130. 2010.

[2] H. Chen, T. Finin, and A. Joshi, "An Ontology for Context-Aware Pervasive Computing Environments", The Knowledge Engineering Review, vol. 18, no. 3, pp.197-207, 2003.

[3] T. Gu, H.K. Pung, and D.Q. Zhang, "A service-oriented middleware for building context-aware services", Journal of Network and Computer Applications, vol. 28, no. 1, pp. 1-18, 2005.

[4] P. Fahy and S. Clarke, "CASS – a middleware for mobile context-aware applications", paper presented at Workshop on Context Awareness, MobiSys 2004, Boston, Massachusetts, June 6, 2004.

[5] T.Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann, and W. Retschitzegger, "Context-awareness on mobile devices - the hydrogen approach". In Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS '03), Washington, D.C.: IEEE Computer Society, vol. 9, no. 9, 2003.

[6] N.D. Rodriguez, "A Framework for Context-Aware Applications for Smart Spaces". In IEEE/IPSJ 11th International Symposium on Applications and the Internet (SAINT), Washington, D.C.: IEEE Computer Society, pp. 218-221, 2011.

[7] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context aware systems". Int. J. Ad Hoc Ubiquitous Computing. vol. 2, no. 4, June 2007, pp.263-277.

[8] A.D. Jurik and A.C. Weaver, "Body Sensors: Wireless Access to Physiological Data", IEEE Software, vol. 26, no. 1, pp. 71-73, 2009.

[9] P.T.T. Thuy, Y-K. Lee, and S.Y. Lee, "DTD2OWL: automatic transforming XML documents into OWL ontology", in Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human (ICIS '09). New York, NY, USA: ACM, pp. 125-131, 2009.

[10] S. Li, Z. Yang, and X. Lin, "RTCR: a soft real-time context reasoner", In Second International Conference on Embedded Software and Systems (ICESS'05), Washington, DC, USA: IEEE Computer Society, pp. 386-391, 2005.

[11] A. Devaraju and S. Hoh, "Ontology-based context modeling for user-centered Context-aware Services Platform", in International Symposium on Information Technology (ITSim 2008), Piscataway, NJ: IEEE, vol. 2, pp. 1-7, 2008.

[12] X.H. Wang, D.Q. Zhang, T. Gu, and H.K. Pung, "Ontology based context modeling and reasoning using OWL", Proceedings of the second Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, vol., 9., pp. 18- 22, March 2004.

[13] J. Euzenat and P. Valtchev. "An Integrative proximity measure for ontology alignment", in Proceedings of the 1st International Workshop on Semantic Integration (ISWC-2003), CEUR-WS.org, vol. 82, p. 33-38. 2003.