ISSN No. 0976-5697

Volume 3, No. 2, March-April 2012

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Multiple Ontology Reasoning over Location Services by Enabling Semantics

Ms. Akansha A. Totewar*
M Tech (CSE), Department of Computer Science and Engineering Government College of Engineering
Amravati, India
Akannsha.Totewar@gmail.com

Dr. Prashant N. Chatur
HOD (CSE), Department of Computer Science and Engineering Government College of Engineering
Amravati, India
prashant.chatur@yahoo.com

Mr. Amit Meena
M.E (CSE), Persistent System Limited
Nagpur, India
amitmeena.sgsits@gmail.com

Abstract: This paper aims at presenting a process of enabling semantics framework, determining how to implement intelligent system using ontology reasoning over location based services. There are several methods and tools that can be applied to contribute in the process of bringing up a module wherein the multiple ontology reasoning can be applied over the location based services. These ontologies may include the context, location, and services knowledge representation. Basically depending on the users context (date, time, longitude and latitude), the framework will infer overall useful information by considering contextual of user and inferring new knowledge on it.

Keywords: Ontology; Services Oriented Architecture; Jena reasoning; OWL and RDF; Protégé.

I. INTRODUCTION

All the field of knowledge engineering is becoming an increasingly important area of computer science. Initiatives such as the Semantic Web [1] "… in which information is given well-defined meaning, better enabling computers and people to work in cooperation" [2], will rely on ontologies to share data. Ontologies provide a shared conceptualization of a domain by defining the concepts in the domain and describing how those concepts are related to each other. However, most domains of discourse are not static, but evolve as the understanding of the domain grows. In order for ontologies to evolve successfully, there is a need for effective tool support. Representation standards for ontologies such as the W3C's Web Ontology Language (OWL) [3] and development tools like Protégé [4] are becoming prevalent, but the tools to support version control and difference comprehension are still lacking for ontology development. Of the ontology development tools currently available, the open source Protégé project developed by the medical informatics group at Stanford University is one of the most mature and best-adopted. The key feature that has contributed to Protégé's success is its open source plug-in architecture that allows it to be easily extended to better suit the needs of particular users.

This present paper establishes a contribution towards enabling Semantics over the Location Based Services using ontology reasoning. The ubiquitous computing develops an environment for human-computer interaction. The framework can be developed so that the user interaction can become easier.

Location based services only provide the attributes like latitude, longitude and altitude but cannot interpret the semantic of locations. Generally, there are two actions performed by the LBS system: one is positioning and the other is providing services based on the location of users. Through the analysis of user attributes, tag attributes and service attributes and the relationship among them, the system will use the rule of ontology reasoning to find the web services to meet user demand, enhancing services searching of precision and completion.

Through the analysis of user attributes, tag attributes and service attributes and the relationship among them, the system will use the rule of ontology reasoning to find the web services to meet user demand, enhancing services searching of precision and completion. The use of ontologies and explicit semantics enable performing logical reasoning to infer sufficient knowledge on the classification of processes that machines offer, and on how to execute and compose those processes to carry out manufacturing orchestration autonomously.

II. RELATED REVIEWS

Wherever in ubiquitous computing environment, it is an important issue for context-awareness, which is aware of context and reasons appropriate service according to the context. Particularly, it is expected that home network adopt ubiquitous computing first of all and provides variable context aware services. Among context-aware services, location-aware service is one of the key context-aware computing [5].

The framework can be used for representing a user-centric view of usage contexts [6]. DL-reasoners can then be used for organizing context definitions, merging domain knowledge into these definitions, and performing recognition of contexts from sensor inputs.

OWLDL ontologies [7] represent non-trivial aspects of context, and to prefer forms of off-line ontological reasoning, while resorting to on-line ontological reasoning only when strictly required.

The motivation having local ontology reasoners is twofold. On one side, we believe that each entity, other than accessing shared ontologies, may hold private ones and may use values in ontological reasoning that should remain private. On the other side, this choice enables what we call on-demand ontological reasoning, which is described in [8]. In particular cases contextual data can be derived through ontological reasoning only populating the ontology with information provided by different entities. In this case, reasoning must be performed on-demand at the time of the service request.

A database supported approach, based on our "meta mapping" approach [9], successfully developed in a former project for huge ontologies on servers. It has the capability to be efficiently scaled down to mobile devices.

Ontologies are seen as the key technology [10] used to describe the semantics of information at various sites, overcoming the problem of implicit and hidden knowledge and thus enabling exchange of semantic contents. As such, they have found applications in key growth areas, such as ecommerce, bio-informatics, Grid computing, and the Semantic Web. An ontology can be described as [11] a specific vocabulary referring to an abstract model of basic concepts of a problem domain. Ontologies are composed of classes describing basic concepts in a domain and relations between them, properties related to features and attributes of the concepts, restrictions of properties and individuals, i.e. instances of the predefined classes.

The Web Ontology Language (OWL) is a W3C recommendation standard that can be used for expressing ontologies which can be processed by software. OWL DL is a sub language of OWL, based on Description Logics and supports those users who need maximum expressiveness while retaining computational completeness which makes it ideal for the ebBP ontology [12].The semantics of an information source according to [13] may be described using an ontology defined as "an explicit specification of conceptualization". The task of integration using different ontologies is a classical problem in information science, and continues to be highly active research issue within many topics, including databases, interoperability, the semantic Web, knowledge representation, data warehousing, and geographical information integration.

III. RESERCH AND METHODOLOGIES

A. Ontologies and their Development:

Ontologies provide a formal specification of a domain of discourse and are becoming increasingly prevalent in the high tech world. There are many different definitions of an ontology and also some question of where an ontology ends and a knowledge base begins [14]; however, for our purposes, Gruber's short definition is suitable. "An ontology is an explicit specification of a conceptualization" [15]. The use of ontologies to construct knowledge base systems is growing rapidly. As already mentioned, they are widely used in the medical community and will provide the backbone of the Semantic Web. On the surface ontologies may appear to be like database schemas; however, ontologies are not a way of organizing a specific data set for efficient retrieval, but rather a reusable structure for data within a domain that is designed to capture all the inherent relationships and meta-data among the knowledge that will be stored in there. Ontologies are intended for both humans and computers to manipulate. In short, ontologies provide a common vocabulary for communication of knowledge within domains [16].

There are two primary methods that have been used to construct ontologies. Description Logic based systems and Frame based systems. The following is a description of the top level items of a frame-based knowledge mode:

- **a. Classes** are collections of objects that have similar properties. Classes are arranged into a subclass-superclass hierarchy using either single or multiple inheritance. Each class has slots (described next) attached to it. Slots can be inherited by the subclasses.
- **b. Slots** are named binary relations between a class and either another class or a primitive object (such as a string or a number). Slots attached to a class may be further constrained by facets.
- **c. Facets** are named ternary relations between a class, a slot and either another class or a primitive object. Facets may impose additional constraints on a slot attached to a class, such as the cardinality or value type of a slot.
- **d. Instances** are individual members of classes.

As ontologies become more complex their development becomes increasingly collaborative, requiring a group of domain experts and engineers to construct them [17]. This parallels the historical development of software systems where, as systems grew in size and complexity, more people were required to complete the project and *ad hoc* development procedures were not suitable. Formal models were required to define the software development process and workflow management tools were required to help engineers adhere to the model. Nowadays, a suite of tools is often used to support software development projects. Two key tools within such a suite are some sort of version control software to track the evolution of the system and a difference tool to compare versions of files. Likewise, ontology development is beginning to enter the stage where projects require a formal development process [17] and the support of tools to help engineers to adhere to a defined process. The set of tools is similar to those used in software engineering with ontology development sharing the need for versioning and difference detection tools.

B. Protégé and OWL:

The Ontology editor Protégé is a free, open source ontology editor and knowledge-base framework which is developed by Stanford University and Manchester University (version 4.0 and above). Protégé is based on Java, is extensible, and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development. It is a desirable tool for editing, browsing ontologies, and some reasoning operations such as incoherence detection can also be performed in it. Protégé 4.0 and the editions above have embedded Pellet and FaCT++ reasoners into ontology editor, which makes reasoning more convenient. Protégé is supported by a strong community of developers and academic, government and corporate users, who are using Protégé for knowledge solutions in areas as diverse as biomedicine, intelligence gathering, and corporate modeling.

C. The Jena Reasoning Agent:

Agents are aware of context of itself, reason context using that information, act of itself and communicate with other agents. And agents react to user action or context, and they have reasoning and learning ability as well as pro-activity for knowledge. Moreover, they perform autonomously action that user needs and have social ability for cooperation among multiple agents. Jena is a Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFs and OWL, SPARQL and includes a rule-based inference engine. Jena is open sources and grown out of work with the HP Semantics Web Programme HP Laps.

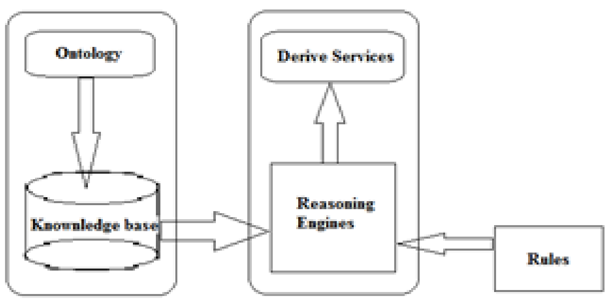

Figure 1: Architecture for agent

The Jena Framework includes:

a. A RDF API
b. Reading and writing RDF in RDF/XML, N3 and N Triples
c. An OWL API
d. In-memory and persistent storage
e. SPARQL query engine.

At its core, Jena stores information as RDF triples in directed graphs, and allows your code to add, remove, manipulate, store and publish that information. We tend to think of Jena as a number of major subsystems with clearly defined interfaces between them.

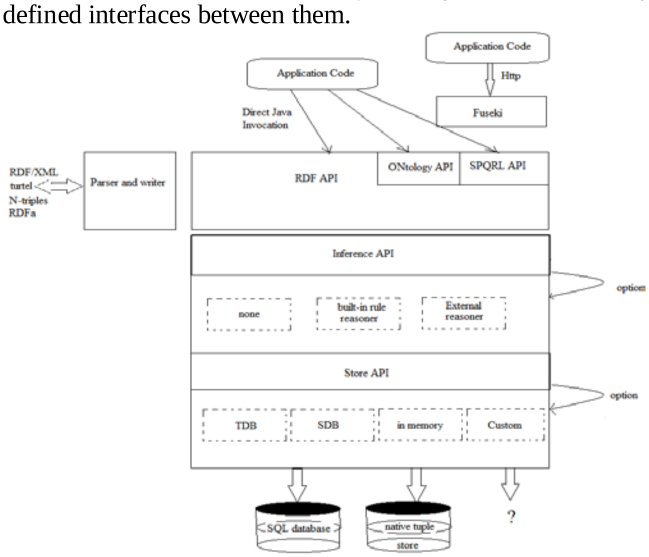

Figure: 2 Jena Architecture Overview

RDF triples and graphs, and their various components, are accessed through Jena's RDF API. Typical abstractions here are Resource representing an RDF resource (whether named with a URI or anonymous), Literal for data values (numbers, strings, dates, etc), Statement representing an RDF triple and Model representing the whole graph. The RDF API has basic facilities for adding and removing triples to graphs and finding triples that match particular patterns. Here you can also read in RDF from external sources, whether files or URL's, and serialize a graph in correctly-formatted text form. Both input and output support most of the commonly-used RDF syntaxes.

While the programming interface to Model is quite rich, internally, the RDF graph is stored in a much simpler abstraction named Graph. This allows Jena to use a variety of different storage strategies equivalently, as long as they conform to the Graph interface. Out-of-the box, Jena can store a graph as an in-memory store, in an SQL database, or as a persistent store using a custom disk-based tuple index. The graph interface is also a convenient extension point for connecting other stores to Jena, such as LDAP, by writing an adapter that allows the calls from the Graph API to work on that store.

A key feature of semantic web applications is that the semantic rules of RDF, RDFS and OWL can be used to *infer* information that is not explicitly stated in the graph. For example, if class C is a sub-class of class B, and B a sub-class of A, then by implication C is a sub-class of A. Jena's inference API provides the means to make these *entailed triples* appear in the store just as if they had been added explicitly. The inference API provides a number of rule engines to perform this job, either using the built-in rulesets for OWL and RDFS, or using application custom rules. Alternatively, the inference API can be connected up to an external reasoner, such as description logic (DL) engine, to perform the same job with different, specialised, reasoning algorithms.

The collection of standards that define semantic web technologies includes SPARQL - the query language for RDF. Jena conforms to all of the published standards, and tracks the revisions and updates in the under-development areas of the standard. Handling SPARQL, both for query and update, is the responsibility of the SPARQL API.

Ontologies are also key to many semantic web applications. Ontologies are formal logical descriptions, or *models*, of some aspect of the real-world that applications have to deal with. Ontologies can be shared with other developers and researchers, making it a good basis for building linked-data applications. There are two ontology languages for RDF: RDFS, which is rather weak, and OWL which is much more expressive. Both languages are supported in Jena though the Ontology API, which provides convenience methods that know about the richer representation forms available to applications through OWL and RDFS.

While the above capabilities are typically accessed by applications directly through the Java API, publishing data over the Internet is a common requirement in modern applications. Fuseki is a data publishing server, which can present, and update, RDF models over the web using SPARQL and HTTP.

IV. IMPLEMENTATION AND MECHANISM

Before implementation of semantic architecture for location services, we require to consider in according to the resources available in the current situation. For defining or designing the ontology reasoner we can use protégé as an editor while we can also use the agent like Jess or Jena reasoners which provide the environment related to the frames, also pellet is a new reasoner which provide the helpful approach towards finding a complete OWL-DL reasoner with extensive support for reasoning with individuals, user-defined data types, and debugging support for ontologies.

Mechanism which can be followed during the implementation of the semantics framework is more complicated as compared to normal framing in Java. As according to the above mentioned semantics architecture we can make a structure by the help of factory design pattern.

V. EXPERIMENTTAL RESULTS

The Fig 3 shows the hierarchy of the classes to be followed during the implementation of the RDF using Jambalaya Tab in protégé.

As we know that there are three types of ontology classification that is,

a. Single Ontology.
b. Multiple Ontology and
c. Hybrid Ontology

Here we can use multiple ontology as it requires ontology to be accessed from the different sections.

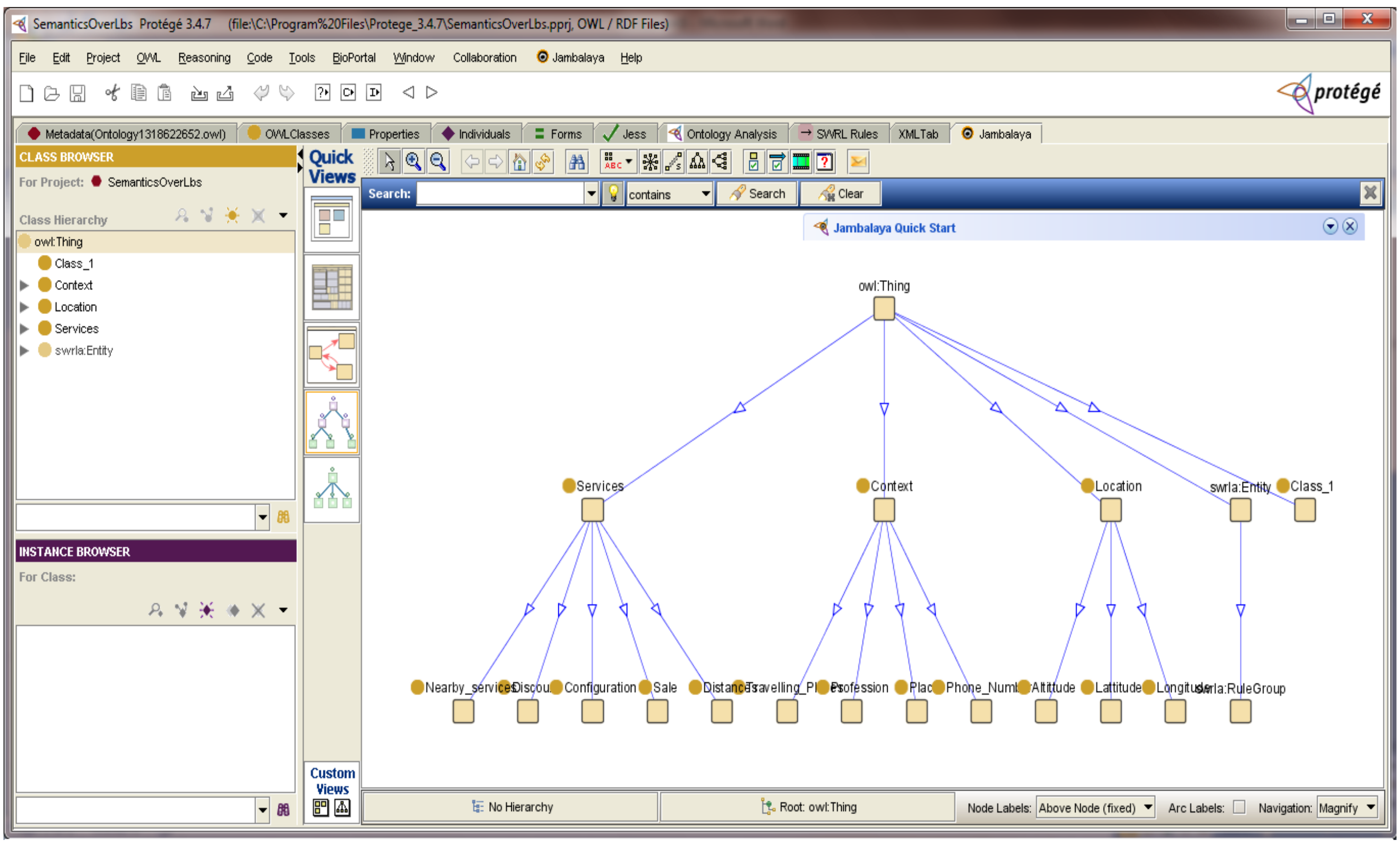

Figure 3: Nested hierarchy of OWL classes (jambalaya tab protégé)

As shown in the Fig 4, there is a user interaction process which is required to be followed. The two fundamental principles of the semantics-based approach are that: all descriptions of service-oriented concepts should be made in an ontology-based formalism; that all ontology-based descriptions should be capable of being connected via mediation.

In the implantation of this process the corresponding user of the semantics will have to bring in his user id and password to enter into the phase of location services mode. This user id provided to the user by the application bring in the more probable possibility for the user to get accurate results as it will provide the user exact position.

The ontology management plays a important role in development of any kind of knowledge representation as in Fig 4, all the ontology will be arranged in such a way that it can automatically approachable according to the situation. Then after making the decision of user proper position in the place the request composition will check for the availability of the user in the database of the ontology as well as the corresponding services data are also followed to make the appropriate output to the user.

There are different ontologies that are to be applied for different types of components in the hierarchy of the classes. Hence the multiple ontology approach is followed during the implementation of this kind of processes.

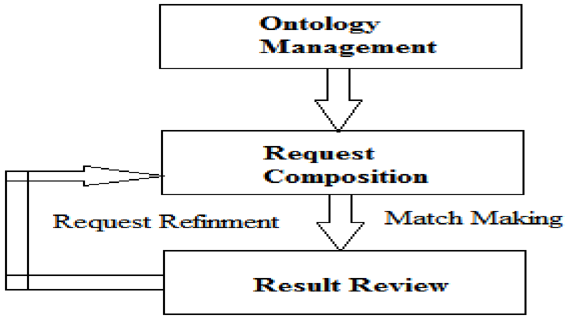

Figure 4: Work flow diagram for user interaction with clients

Sample protégé page Fig 5 elaborate the owl classes and there hierarchy that to be followed during the collaboration of the different ontology.

Here we will be requiring ontology for location of the user which involves the context for the user . Also after finding out the exact position we can also follow the ontology for the service provider as it would bring up different services for which the user is using the application, also as according to different possible and most frequently approached places and sites are made involved so that it becomes easier for the user for making the choices.

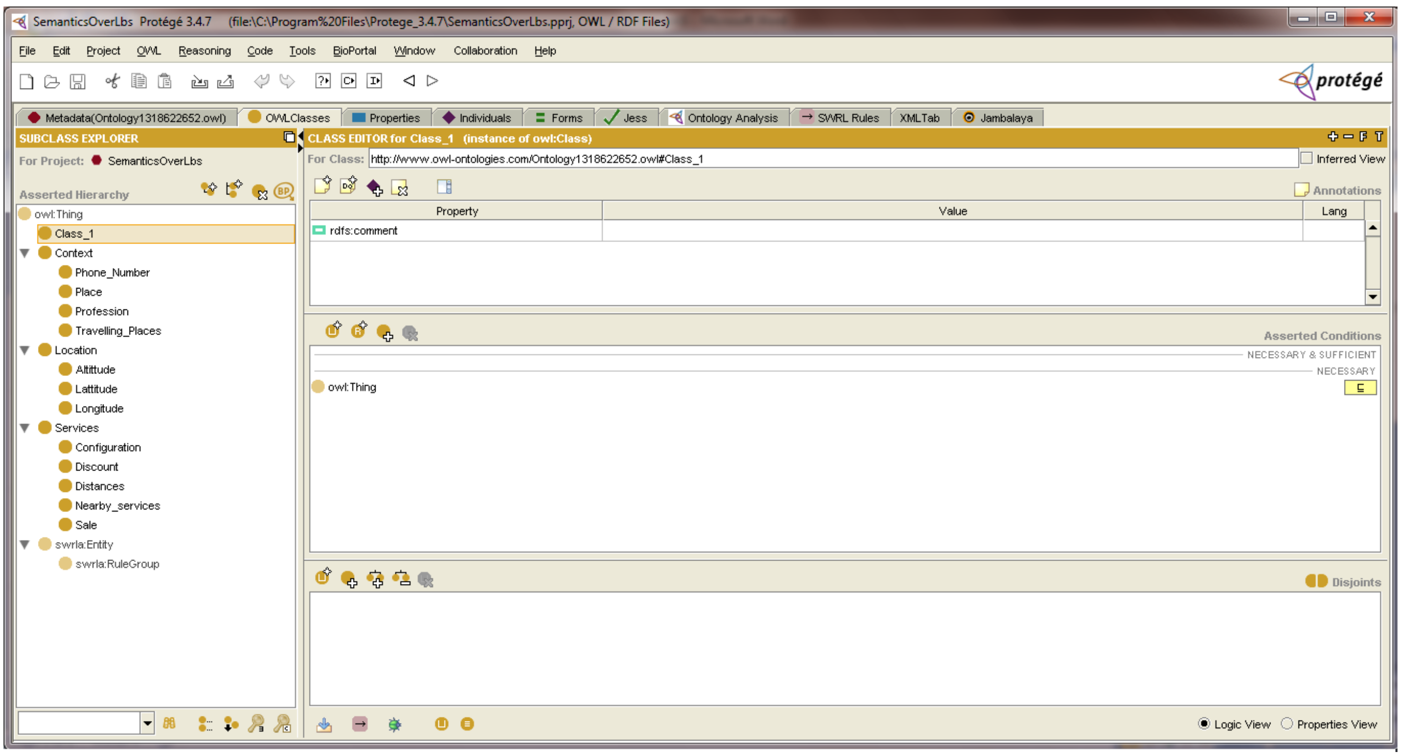

Figure 5: OWL hierarchy of classes and there instances

VI. CONCLUSION

Ontology reasoning can be used to enable the semantic framework. Using the location based services we can contribute to the web services provided by any of the stock markets as well as the user based attributes can used to find out the nearby available sources to the user helping it all the way. This technology is growing day by day and is helping to introduce advance services.

Location Based Services plays an important role in realizing enhancing the usability of the Semantics framework design, the improvement of customers' relations and improving the requirement of system performance and so on. Ontology reasoning provides the support for the Semantic Framework design, and other business making decision, etc.

VII. ACKNOWLEDGMENT

The Location Based Services has provided many reviews and approach towards the new beginning of advance technology which can be achieved by enabling semantics over it. Also, Protégé is supported by a strong community of developers and academic, government and corporate users, who are using Protégé for knowledge solutions in areas as diverse as biomedicine, intelligence gathering, and corporate modelling.

Hence, this presented topic of dissertation has been supported and sponsored by Persistent System Limited (PSL), Nagpur Maharashtra, India. Our special thanks to the experts who have contributed towards developing the project. PSL is known for its innovative approach towards the technical development and encouraging candidature to implement new ideas.

VIII. REFERENCES

[1] W3C Semantic Web: http://www.w3.org/2001/sw/

[2] T. Berners-Lee, J. Heldler, and O. Lassila, "The Semantic Web," in Scientific American, 2001, vol. 284, pp. 34-43.

[3] W3C Web Ontology (WebOnt) Working Group (OWL): http://www.w3.org/2001/sw/ WebOnt/

[4] The Protégé Project: http://protege.stanford.edu

[5] I. Kong, J. Kim, S. Lee, Jung-Tae Lee and Hwang W., "Design and Implementation of Ontology-based Context Reasoning Engine for Adaptive Multimedia Service Migration", IEEE digital library, 2005.

[6] O. Lassila and D. Khushraj, "Contextualizing Applications via Semantic Middleware", IEEE proceedings of the Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'05) , 2005.

[7] Alessandra Agostini, Claudio Bettini and Daniele Riboni, "Loosely Coupling Ontological Reasoning with an Efficient Middleware for Context-awareness", IEEE digital library, 2005.

[8] Ioanna Roussaki, Maria Strimpakou, Carsten Pils, Nikos Kalatzis, Miltos Anagnostou, "Hybrid context modeling: A location-based scheme using ontologies", Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06), 2006.

[9] Gunther Specht and Timo Weithoner, "Context-Aware Processing of Ontologies in Mobile Environments", Proceedings of the 7th IEEE International Conference on Mobile Data Management (MDM'06) , 2006.

[10] Dong Liu, Junliang Chen, and Jie Guo, "An Ontology-based LBS System", IEEE, 2008.

[11] Radoslaw Brendel, Henryk Krawczyk, "E-mail User Roles Identification Using OWL-based Ontology Approach", Proceedings of the 1st IEEE International Conference on Information Technology, IT 2008.

[12] B. R. Heravi, David Bell, Mark Lycett, "Towards an Ontology for Automating Collaborative Business Processes", 14th IEEE International Enterprise Distributed Object Computing Conference Workshops, 2010.

[13] Y. G. Cheong, Y. Kim, S. Y. Yoo, H. Lee, S. Lee, S. C. Chae and H. Choi, "An Ontology-Based Reasoning Approach Towards Energy-Aware Smart Homes", The 8th Annual IEEE Consumer Communications and Networking Conference - Emerging and Innovative Consumer Technologies and Applications, 2011.

[14] N. F. Noy, and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," Stanford University, Stanford, CA 2001.

[15] T. R. Gruber, "Toward Principles for the Design of Ontologies Used for Knowledge Sharing.," *International Journal of Human-Computer Studies*, 1995, vol. 43, pp. 907-928.

[16] A. Farquhar, R. Fikes, W. Pratt, and J. Rice, "Collaborative Ontology Construction for Information Integration," Stanford University, Stanford, CA, 1995.

[17] Y. Lu, "Roadmap for Tool Support for Collaborative Ontology Engineering," Computer Science, University of Victoria, Victoria, 2003.

