Modeling an Ontology for Managing Contexts in Smart Meeting Space

Mohammad Rezwanul Huq, Nguyen Thi Thanh Tuyen, Young-Koo Lee, Byeong-Soo Jeong and Sungyoung Lee
Department of Computer Engineering
Kyung Hee University
Yongin-Si, Gyeonggi-Do, Republic of Korea

Abstract *- In recent years, computing becomes more mobile and pervasive; these changes imply that applications and services must be aware of and adapt to their changing contexts in highly dynamic environments. To allow interoperability in a context-aware computing environment (e.g. smart meeting space), it is necessary that the context terminology will be commonly understood by all the connected entities in a particular space. In this paper, we propose a context ontology model for smart meeting space. The proposed context ontology model defines seven major concepts those can be reused as well as provides the basic infrastructure to build an ontology model for smart space environments. As each smart environment may have different kinds of users, devices etc., we design domain-specific ontology for smart meeting space and implement a prototype for this ontology model using Protégé 3.2[1] ontology editor and RacerPro 1.9 [2] reasoning engine. We execute queries on our ontology using SPARQL [3] query language; furthermore, we define rules using SWRL [4] and infer those rules by JESS [5] rule engine. The proposed context ontology model has been designed to improve knowledge sharing, knowledge reuse, context querying and reasoning in a smart meeting space.*

Keywords: Ontology, Context, Smart Meeting Space.

1 Introduction

Emerging ubiquitous computing technologies provide "anytime, anywhere" computing by decoupling users from devices and applications as entities that perform tasks on behalf of users [6]. With the help of these emerging ubiquitous technologies, people are aiming to create a new human workspace (e.g. Smart Meeting Space) in which human interaction remains focused on the interaction with other humans and on actual human activities. The meeting space must be aware of its contexts and automatically adapt to their changing contexts- known as context-awareness. By context, here we refer to any information that can be used to characterize the situation of an entity, where an entity can be a user, environment, services provided to users, location, platform, integrated devices and the rooms' activities.

Building context-aware system in an environment like smart meeting space is still a complex and time consuming task due to lack of an appropriate infrastructure or middleware-level support. In this paper, we propose a context ontology model for efficient knowledge sharing, knowledge reuse, context querying and reasoning in smart meeting space. It supports semantic context representation by defining the common ontology known as smart space ontology in smart environments. Though our primary concern is to build the context ontology model especially for smart meeting space, we identified the major concepts in a smart environment so that these concepts can be reused to build a large-scale context ontology model without starting from scratch.

The rest of the paper is organized as follows. Section 2 begins the discussion of related work. In section 3, we provide an example of smart meeting space. In the next section, we try to explain the uniqueness of our model. In section 5, we describe our proposed ontology followed by prototype implementation in section 6. Finally, we talk about some of our future works and draw conclusion in section 7 followed by acknowledgement in section 8.

2 Related Work

Much research has been initiated in the area of context-aware computing in the past few years. AT&T Laboratories at Cambridge built a dense network of location sensors to maintain a location model shared between users and computing entities [7]. Microsoft's EasyLiving [8] develops a prototype architecture and technologies for building intelligent environment which is aware of users' presence and adjusts environment settings to suit their needs; other relevant projects include Stanford University's iRoom [9] and Carnegie Mellon University's Aura [10] have significantly contributed to smart space research by taking advantage of different pervasive computing features. Some works focused more on constructing ontology for context in a specific domain to reach the goals of knowledge sharing across distributed systems. CONON [11] introduced extensible context ontology for pervasive computing environments. Eleni Christopoulou et al. focused on an ontology-based context modeling, management and reasoning process developed for composing context-aware UbiComp applications from AmI artifact [12]. COBRA [13] proposed by Chen et al. used an ontology to describe person, places and intentions.

The ontology oriented approach explores the potential capabilities of context reasoning based on semantic web technologies. We propose a context ontology model for smart meeting space. Our proposed model is divided into smart space and domain specific ontology so that the general ontology can be reused for other smart spaces like hospital, departmental store etc and the domain specific ontologies can be extended with the passage of time for a particular domain. CONON, COBRA is the general purpose ontology model. On the contrary, our ontology model exclusively focuses on smart meeting space and also identifies the major concepts for smart environments.

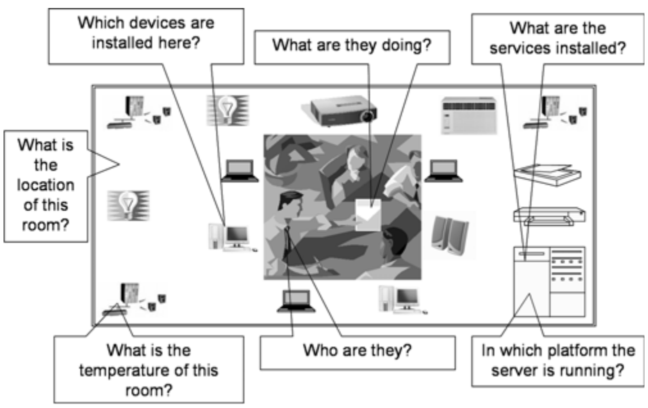

Figure 1: Smart Meeting Space: An Example

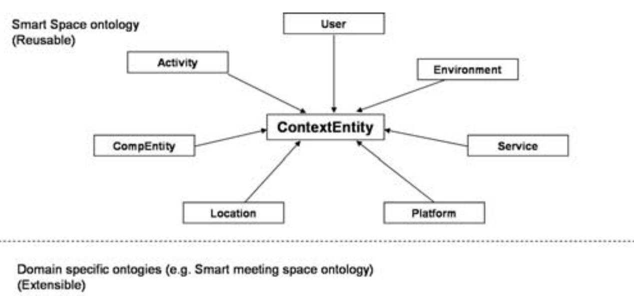

Figure 2: General Context Ontology

3 Smart Meeting Space: An Example

Figure 1 shows a typical example of a smart meeting space. By observing the scenario, there are some questions arise into our mind. The questions are already mentioned in the figure. Based on these questions, we tried to identify the major concepts those can be shared among all smart environments. Identifying the major concepts (figure 2) can help us to build ontology model for other smart spaces without starting from the scratch.

Considering the fast evolution in the hardware and software industry, it is important that decisions made today regarding our context specification are adaptable and extensible. We therefore opted to define generic context ontology model. The ontology model is structured around a set of concepts, each describing a physical or conceptual object including *User*, *Environment*, *Service*, *Platform*, *Location*, *CompEntity* (Computational Entities) and *Activity*.

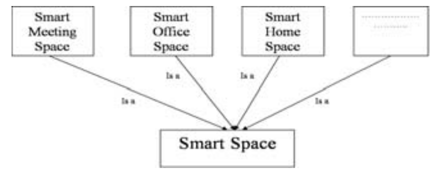

Figure 3: Different kinds of Smart Spaces

4 Uniqueness of Our Model

In our model, we have introduced two levels of ontology. Smart space ontology or general ontology and Smart meeting space ontology or domain-specific ontology. In the next part we will describe the reason for introducing general ontology and domain-specific ontology. Moreover, we will try to differentiate our model from CONON and COBRA.

4.1 Why Smart Space or General Ontology?

There may be a number of smart space environments like smart office, smart home etc. (figure 3). Though our primary concern is to build ontology for smart meeting space, we start to build our ontology model by considering the common traits of a smart space. So that, whenever we will build ontology for smart office then we can reuse some classes or major concepts. Moreover, for building a complete ontology for any smart organization, the major concepts should be shared among all the domain-specific ontologies like smart meeting space ontology or smart home ontology.

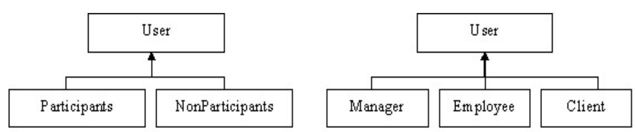

Figure 4: Different classification of User concept depending on domain

4.2 Why Smart Meeting Space or Domain-specific Ontology?

The figure 4 explains the User concept for two different domains. The domain-specific ontology is extensible in the sense that any new concept or property to any concept can be added without modifying the set of major concepts. The smart space ontology contains the basic concepts and common properties related with these concepts for all kinds of smart spaces. The domain-specific ontology will be built by reusing these concepts and properties.

4.3 Why Environment, Service and Platform concepts?

CONON ontology model has four concepts such as Computational Entities, Location, Person and Activity where as COBRA has Place, Person and Intentions concepts. We introduce *Environment*, *Service* and *Platform* as major concepts; those are not considered in CONON and COBRA. In the next portion we will try to explain why these concepts have been included in our model.

Environment concept holds information on environmental condition. Each location has some different environmental conditions. As for example, temperature is environmental information. If the meeting space temperature is high then the air-conditioner will adjust the temperature in a comfortable state.

Service concept identifies the services provided to the meeting participants inside any smart meeting space. During the meeting, a meeting participant may use one or more services provided by the smart meeting space. It is very important to know that which user uses which service currently as each service may have some maximum limit of users.

Platform provides software and hardware. Each platform may use different application software, operating system, middleware or Java VM. According to our requirements, we can choose any platform that provides our required software and hardware.

4.4 Comparative Study

We will try to compare our ontology with CONON and COBRA. These two ontology models are more similar to our ontology model than any other models.

4.4.1 CONON vs. Our Ontology

CONON defines four basic concepts. They don't define the environment, service and platform concept. Therefore, their model is unable to satisfy the queries on environment, service and platform. This is a major drawback of their ontology model. On the contrary, as the contexts on environment, service and platform will be stored in our ontology, it is very much possible to satisfy queries on these concepts.

4.4.2 COBRA vs. Our Ontology

COBRA does not have any general or upper level ontology model. For building a complete ontology model for a smart organization it will be difficult to ensure knowledge reuse and knowledge sharing. It defines only three concepts Person, Place and Intentions (User role). Therefore, any queries except these concepts cannot be satisfied using COBRA ontology model; and obviously our ontology model focuses on with smart meeting space where as COBRA and CONON are general purpose ontology model.

5 Proposed Ontology for Smart Meeting Space

In this paper, we divide our context ontology model into general ontology that can be reused for any application domain and domain specific ontology that is capable of supporting extensibility for the domain of interest. The general ontology is a high-level ontology which captures general features of basic contextual entities. Specific ontology is a collection of ontology set which defines the details of general concepts and their features in each sub-domain.

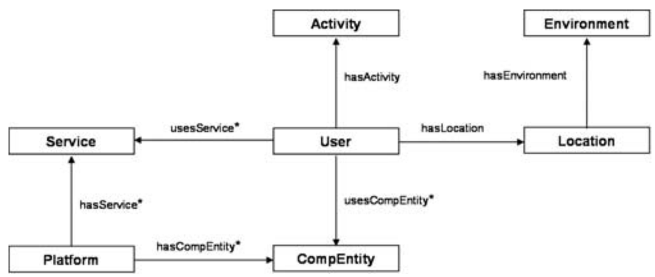

Figure 5: Context Ontology Overview

5.1 Overview of Proposed Ontology Concepts

We determined seven major concepts around which we built our ontology. These are based around the most important aspects in context information.

i) User: The user plays a vital role in smart environments. The applications and services within its environment should adapt to the user, and not vice versa.

ii) Environment: The environment of smart space in which the user interacts is an important aspect of the context specification.

iii) Service: Services provide specific functionality to the user. User uses service and these services will be provided by platform.

iv) Platform: This concept is dedicated to the hardware and software description of a smart environment. This includes specification of operating system, middleware and virtual machine etc.

v) Location: This concept provides the location of the smart space. Location has been classified into two different sub classes such as *SmartMeetingLocation* and *OtherLocation*.

vi) CompEntity: CompEntity stands for computational entities. It includes not only the computational devices but also the resources.

vii) Activity: A smart environment may have different kinds of activities like deduced and scheduled activities.

The interaction among aforementioned ontology concepts proposed in this paper is given in figure 5.

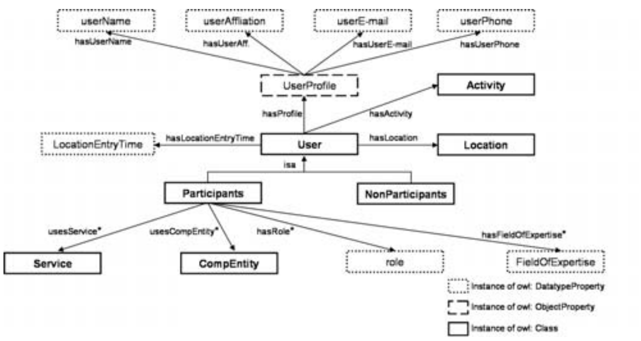

Figure 6: User Ontology Concept

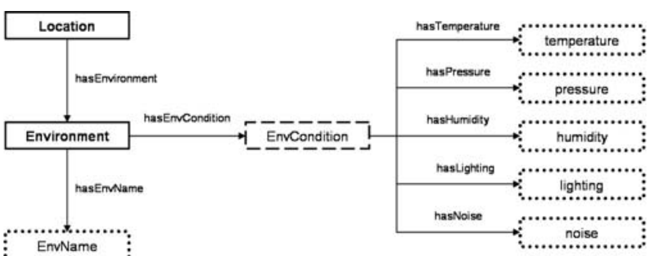

Figure 7: Environment Ontology Concept

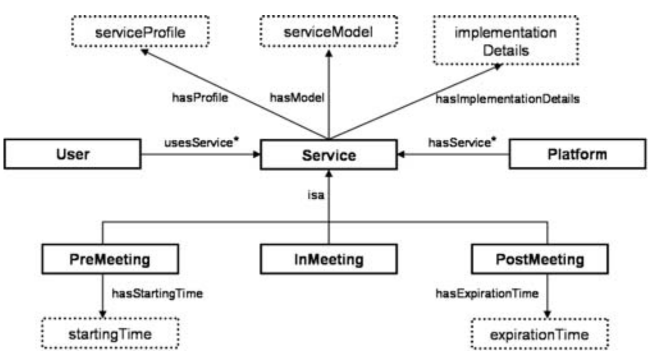

Figure 8: Service Ontology Concept

5.2 User Ontology Concept

Context information is only relevant if it influences a user's task. For this reason, the user should take a central place in smart environments. In our proposed model, User class is classified into two sub classes such as *Participants* and *NonParticipants* class. Figure 6 depicts user concept, its attributes and relationship with other concepts. The legend used here is also applicable from figure 7-12.

5.3 Environment Ontology Concept

Environment consists of environmental condition information. Environmental conditional information includes temperature, pressure, humidity, lighting and noise. This environmental condition information is very important to control devices to give proper condition to hold any kind of activity in smart meeting room. Figure 7 shows the underlying relationship in environment ontology concepts.

5.4 Service Ontology Concept

In several computer science domains the concept of service refers to a functionality offered by any smart environment. In figure 8, we give an overview of the service concept. Typically, a user wants to use a service to meet his/her demands. Each platform can host several services and/or employ several remote services by using different CompEntity connected to the smart environment. Services can be broadly classified into three types such as *PreMeeting*, *InMeeting* and *PostMeeting* service.

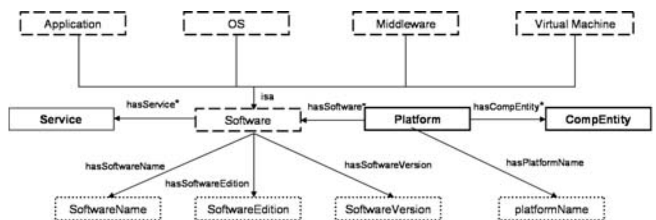

Figure 9: Platform Ontology Concept

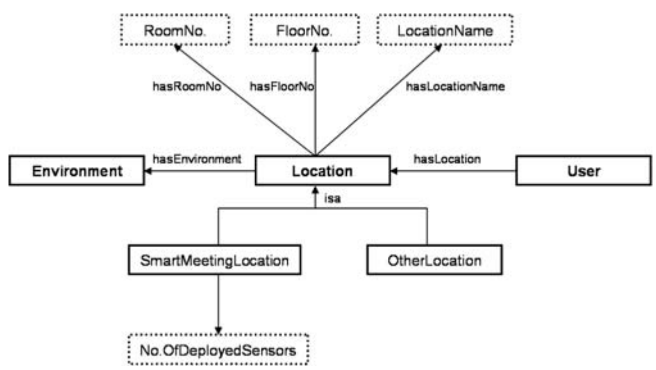

Figure 10: Location Ontology Concept

5.5 Platform Ontology Concept

The platform section of the ontology provides a description of (i) the software that is available on the device for the user or other services to interact with, and (ii) the computational entities-CompEntity which defines the devices and system resources. An overview of this part of the context specification is shown in figure 9.

5.6 Location Ontology Concepts

Location concept has two sub classes such as *SmartMeetingLocation* and *OtherLocation*. Figure 10 gives us an overview of the relationships of location concept. It shows that each user has a specific location and each location has its own environment. The figure also depicts the properties associated with location concept.

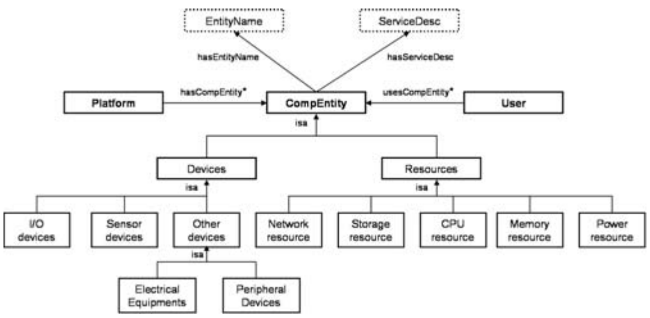

Figure 11: CompEntity Ontology Concept

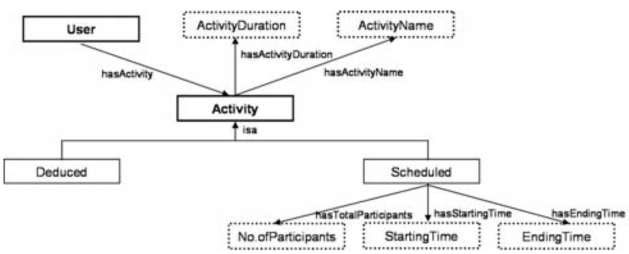

Figure 12: Activity Ontology Concept

5.7 CompEntity Ontology Concepts

Computational entities include different devices as well as system resources. Devices can be broadly classified into three categories such as *I/O devices*, S*ensor devices* and *Other devices*. System resources can be roughly divided into the following categories such as *power* resources, *memory* resources, *CPU* resources, *storage* resources and *network* resources. Figure 11 shows the graphical view of CompEntity ontology concept.

5.8 Activity Ontology Concepts

An overview of activity ontology concept of the context specification is shown in figure 12. Each smart environment has some activities to perform. These activities can be grouped into S*cheduled* activities and *Deduced* activities. Detection of current activities is an important part of context reasoning.

6 Prototype Implementation

In this section, we present the implementation of our proposed context ontology for smart meeting space. We used Protégé 3.2 as a tool to build our proposed ontology. Protégé is a free, open source ontology editor and knowledge-base framework which was developed by Stanford Medical Informatics at the Stanford University School of Medicine.

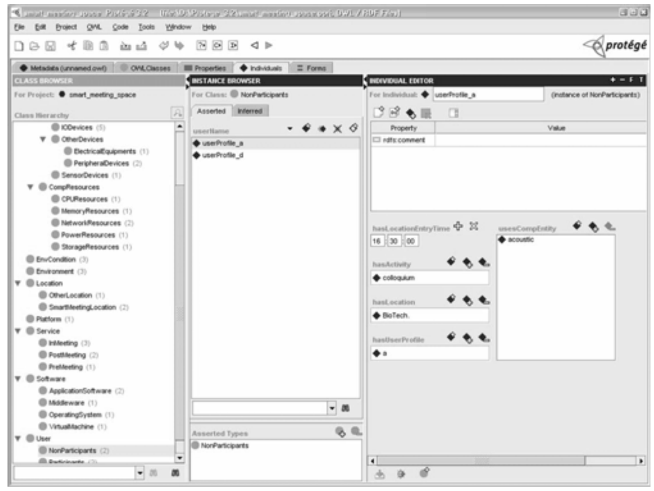
Figure 13: Classes and Instances for Smart Meeting Space


Figure 14: Inferred Taxonomy generated by RacerPro 1.9

6.1 Classes and Instances

We entered all the concepts involved with our context ontology model as an instance of *owl:class*. For the simplicity of our work, we introduced some more *owl:class* (e.g. *UserProfile* to keep users information). We have also defined the *owl:DatatypeProperty* and *owl:ObjectProperty* for each *owl:class* and their cardinality according to our context ontology model. There are some owl:ObjectProperty which has been related with another one by means of owl:inverseOf functionality. As for example, *hasLocation* and *hasUser* is two inverse owl:ObjectProperty. Figure 13 shows the classes and instances declared for the development of our context ontology model for smart meeting space. We have also used RacerPro 1.9 reasoning engine to identify the inconsistent classes, inferred class hierarchy as well as inferred instances. Figure 14 shows the inferred taxonomy of our ontology model.

6.2 Queries

Using some synthetic instances of owl:class we execute a reasonable number of queries which are very important and useful for our smart meeting space. We used SPARQL for executing queries on out ontology model.

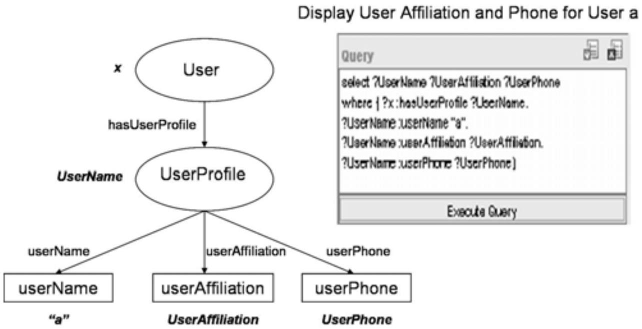

Figure 15: Execution mechanism for SPARQL queries

Figure 15 depicts an example of SPARQL query. We want to find the user affiliation and phone no. for user "a". In SPARQL, we can declare some binding variables which will be bounded to particular owl:class. Here *x* is bounded to the owl:class *User*. *UserName* is bounded to the owl:class *UserProfile* using the property *hasUserProfile*. *UserProfile* class has three properties *userName*, *userAffiliation* and *userPhone* and these properties has been bounded to variable "*a*", *UserAffiliation* and *UserPhone*. Finally, we are displaying the values of the binding variables "*a*", *UserAffiliation* and *UserPhone*.

We have divided the queries into the following classifications. Later, we include examples for each type of queries.

i) Queries on single individual.
ii) Queries on multiple individuals.
iii) Combined form of the first two types.
iv) String matching and testing values queries.

Figure 16: Display Affiliation and Phone for user "a"

Figure 17: Display Type of the user "b"

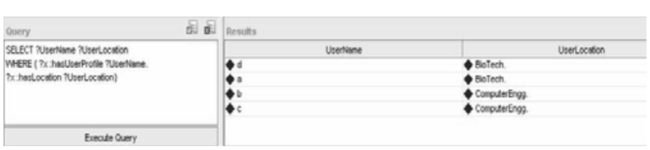

Figure 18: Display user Name and their Location

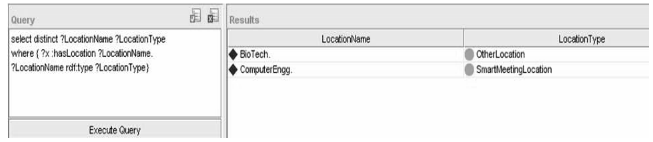

Figure 19: Display location Name and location Type

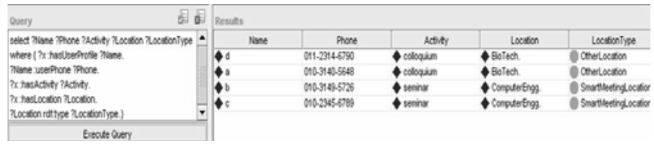

Figure 20: Display user Name, Phone, Activity, Location

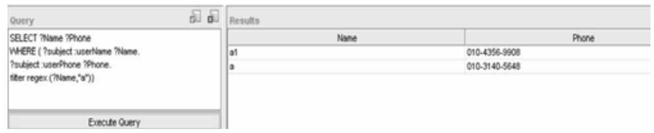

Figure 21: Display user Name and Phone no. whose name starts with "a"

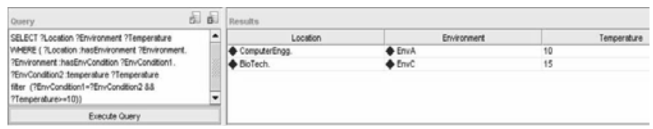

Figure 22: Display Location, Environment and Temperature if the location temperature >= 10

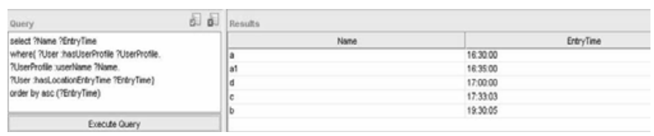

Figure 23: Display user Name and location Entry Time in ascending order of Entry Time

Figure 24: Display only 3 records of Name and Phone no.

Moreover, by using the keyword *group by*, *distinct*, *limit*, we can order and slice result set according to our necessity (figure 23 and 24).

6.3 Rules

In our prototype implementation, we try to define rules for the smart meeting space environment. We define rules using SWRL (Semantic Web Rule Language) and infer those rules using JESS (Java Expert Shell System). All rules are expressed in terms of OWL concepts (classes, properties, individuals, literals etc.). Examples of SWRL rules has been given below.

i) *General Case*: User(?x)∧hasLocation(?x, ?y)∧ SmartMeetingLocation(?y) → Participants(?x)

ii) *SWRL Rule with Named Individuals*:
User(User_c) ∧ hasActivity(User_c, ?y) ∧ Activity(?y) → hasStatus(User_c, "busy")

iii) *SWRL Rule with Literals and Built-ins*:
hasEnvCondition(?x, ?y) ∧
hasTemperature(?y, ?z) ∧
swrlb:greaterThan(?z, 12) →
hasEnvComments(?x, "Hot")

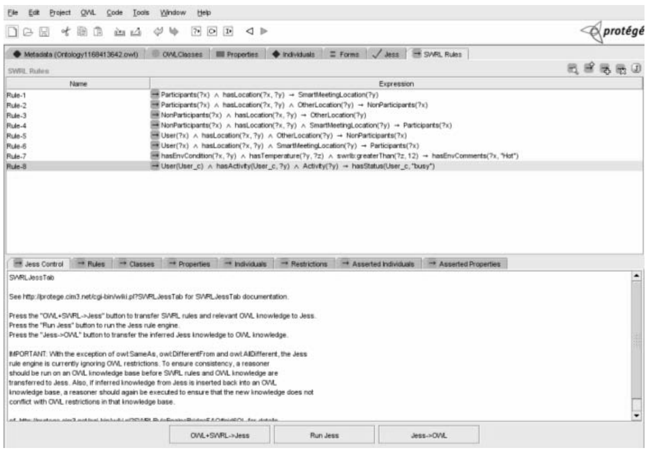

Figure 25: Interface for defining SWRL rules and inferring rules using JESS

Next, we use JESS rule engine to infer those rules. Figure 25 shows the interface for defining SWRL rules and inferring those rules using JESS.

7 Conclusion and Future Work

The necessity of ontologies for the establishment of context-aware pervasive computing systems is broadly acknowledged. In this paper, we have presented a context ontology model for smart meeting space. The major concepts can be reused for other smart environments too. Moreover, we can add more properties to each concept exist in our proposed model depending on the domain-specific needs.

Based on the gained experience from prototype implementation, our context ontology model will be further refined. Moreover, we want to build ontology models for other smart environments too by using the general ontology defined in this paper. Further attention will be paid to identify frequently occurring queries as well as to define more rules generating high level context from low level context to facilitate context reasoning thus appear to build an established ontology model for smart environments.

8 Acknowledgement

This work is supported by Ubiquitous Computing and Network (UCN) grants Z1500-0603-0004, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.

9 References

[1] Protégé (http://protege.stanford.edu), Stanford Medical Informatics.

[2] RacerPro (http://www.racer-systems.com), Racer Systems.

[3] SPARQL query language for RDF (http://www.w3.org/TR/rdf-sparqlquery/), World Wide Web Consortium.

[4] SWRL: A Semantic Web Rule Language Combining OWL and RuleML, World Wide Web Consortium.

[5] JESS: The Rule Engine for Java Platform (http://herzberg.ca.sandia.gov/jess/), Sandia National Laboratories.

[6] K. Henricksen, J. Indulska, and A. Rakotonirainy, "Infrastructure for pervasive computing: Challenges"; Workshop on Pervasive Computing INFORMATIK 01, Vienna, Austria, September 2001.

[7] A. Harter and et al., "The anatomy of a context-aware application"; Wireless Networks, vol. 8, no. 2-3, pp. 187-197, 2002.

[8] B. Brumitt and et al., "Easyliving: Technologies for intelligent environments"; 2nd International Symposium on Handheld and Ubiquitous Computing (HUC 2000), LNCS 1927, Springer-Verlag, 2000, pp. 12-19.

[9] B. Johanson, A. Fox, and T.Winograd, "The interactive workspaces project: Experiences with ubiquitous computing rooms"; IEEE Pervasive Computing, vol. 1, no. 2, pp. 67-74, 2002.

[10] D. Garlan and et al., "Project aura: Towards distraction-free pervasive computing"; IEEE Pervasive Computing, vol. 1, no. 2, pp. 22–31, 2002.

[11] X. Wang and et al., "Ontology based context modelling and reasoning using owl"; Workshop on Context Modelling and Reasoning at IEEE Int'l Conference on Pervasive Computing and Communication, USA, 2004.

[12] E. Christopoulou, C. Goumopoulos, and A. Kameas, "An ontology-based context management and reasoning process for ubicomp applications"; Joint Conference on Smart Objects and Ambient Intelligence, Grenoble, France, 2005, pp. 265-270.

[13] H. Chen, T. Finin, and A. Joshi, "An ontology for context-aware pervasive computing environments"; Special Issue on Ontologies for Distributed Systems, 2003, knowledge Engineering Review.