December 2005

A Framework for Ontology-based Context Base Management System

Sungjin Kim
Pohang University of Science and Technology

Euiho Suh
Pohang University of Science and Technology

Keedong Yoo
Pohang University of Science and Technology

Yunmi Lee
Pohang University of Science and Technology

A Framework for Ontology-based Context Base Management System

Sungjin Kim	Euiho Suh	Keedong Yoo	Yunmi Lee
Pohang University of Science and Technology	Pohang University of Science and Technology	Pohang University of Science and Technology	Pohang University of Science and Technology
sciv@postech.ac.kr	ehsuh@postech.ac.kr	kdy@postech.ac.kr	ymlee82@postech.ac.kr

Abstract

Although many context-aware systems have been developed, little attention has been paid to building a uniform framework of a Context Base and Context Base Management Systems (CBMS). Consequently, each context-aware system has to be built from the ground up. It is important that context-aware systems share context data across different domains and with all devices or services. Therefore, context data in the context base is represented as ontology which represents a new paradigm for modeling and representing the context data in this research. In this paper, we describe a CBMS framework and a model of an ontology-based Context Base. CBMS consists of 3 modules which are a browse module, a logical module and a physical module. A business trip request system has been implemented as a prototype. Traditional databases and DBMSs have limitations adapting to a ubiquitous computing environment. As context-aware services develop further, the necessity of a uniform framework of Context Base and CBMS increases. It will support many services which use context data and give guidelines to developers who seek methods to handle context data.

Keywords: context, context base, CBMS, ontology

1. Introduction

Many researchers have paid attention to handling context data since ubiquitous computing (sometimes called 'pervasive computing') was proposed (Gu, 2005). Context data, which provide information about the present status of people, places, things and devices in an environment, play an important role in ubiquitous computing (Khedr and Karmouch, 2005). Similarly, Prekop and Burnett (2003) defined context data as information that can be used to characterize the situation of an entity defined as the person, place, or object, that is considered relevant to the interaction between a user and an application. Context provides a ubiquitous computing environment with the ability to adjust the services or information it provides by implicitly deriving the user's needs from the context (Prekop and Burnett, 2003).

Context-aware computing is an application that uses the context of an entity to modify its behavior to best meet the context of the user (Dey and Abowd, 1999). Barkhuus and Dey defined three levels of services for a context aware application: personalization, passive context-awareness and active context-awareness (Barkhuss and Dey, 2003). It is essential for context-aware computing to store context data and use it properly for any level of services.

However, there are some problems related with context-aware systems. First, there is no uniform support for representing and storing contexts. Because of this lack of generality problem, each application has to be built from the ground up (Dey et al. 1999). Furthermore, context data must be shared by devices and services. However, this is difficult to accomplish because of differences in programming languages.

In order to solve these problems and to support handling context, we define a Context Base and a Context Base Management System (CBMS). To perform context-aware services, it is essential to store context data and to use them properly. A context Base, which stores contexts, is similar to a database. It stores various contexts while the CBMS updates, modifies and deletes context data from context base.

In this research, context data in Context Base are represented as an ontology. An ontology is a formal explicit description of the terms in a certain domain and the relations among them (Gruber, 1993). The ontology defines a common vocabulary for researchers who need to share information in a domain. Ontology has been widely used in many areas such as AI (Artificial Intelligence) and Semantic Web (Berners-Lee et al., 2001).

There are some reasons why context is represented as ontology. Khedr and Karmouch (2005) described that ontology represents a new paradigm for modeling and representing context data. They concluded that, due to the variety of context data, it is unrealistic to formalize completely all context data. Instead, using ontology, they modeled context data in the form of expressiveness levels. It is important to share context data in the same domain or across different domains (Gu et al., 2005). By applying ontology to ubiquitous computing environments, it enables context sharing by explicitly defining contexts. Likewise, context must be shared by all devices or services. Context which is represented by ontology allows describing contexts semantically in a way which is independent of programming languages.
Hence, this paper describes how context data represented as ontology are stored in Context Base and are handled by CBMS. Section 2 discusses related works. In Section 3 we describe our modeling concepts, followed by the prototype implementation in Section 4. We present our conclusions in Section 5.

2. Related Research

2.1. Previous Context Base and CBMS

There are few related studies about Context Base and CBMS. Several ubiquitous computing projects such as 'Active Badge' (Want et al., 1992) and 'Cooltown' (Kindberg and Barton, 1999) were launched. In these projects, context data play an important role in building the so-called context aware system. Many researchers regarded context-awareness as important. Consequently, in handling context data, context-aware system can grasp the context of people or the environment. Therefore, there is no doubt that something similar to a database stores context data and by using it context-awareness can be achieved. However, these systems failed to provide not only a standard context data format but also a Context Base as a repository for context data.

Hong and Landay (2001) introduced an infrastructure approach to context-aware computing. They described the advantages of their approach such as independence from hardware, operating system, and programming language, improved capabilities for maintenance, and evolution and sharing of sensors, processing power, data, and services. In order to build a context-aware infrastructure, they defined standard data formats of contexts and protocols which need to be rich enough to cover the diverse range of sensors and assorted types of context. However, their data format and protocols are too complex to implement. The aforementioned limitations draw to the idea the building of a common model.

2.2. Ontology-based Context Base

Gandon and Sadeh (2004) described Semantic e-Wallet represented as an ontology. They introduced a Semantic Web architecture aimed at supporting automated discovery and access to personal resources in support of a variety of context-aware computing. In this architecture, every source of context data is represented as an ontology. However, this research emphasized not context data storage and management, but services with context data. In other words, this research focused on how context-aware applications or services were operated and accessed context data.

Kwon et al. (2005) proposed a ubiquitous decision support system (UbiDSS). In a UbiDSS, context data from various context trackers such as RFID readers, weather data acquisition sensors, etc. are stored in a Context Base. A UbiDSS has the context subsystem which consists of events acquisition, context inference, and action request module. Using this context data, the UbiDSS can support a user's decision. Nevertheless, a Context Base in UbiDSS is conceptual, and there is no description of how to store context data in a Context Base or on how to handle context data.

Some researchers have looked at context data represented as an ontology. Gu et al. (2005) have proposed the Service Oriented Context-Aware Middleware (SOCAM). In this research, context data were represented as an ontology. However this study focused not on the information system environment but rather on services in a home environment. Ranganathan and Campbell (2003) introduced a middleware which is a part of Gaia, for context-aware agents in ubiquitous computing environments. They used ontologies that define the structure of context data. Ontologies made it easier to specify the behavior of context-aware applications.

Sugumaran and Storey (2002) proposed an ontology management system. The Ontology management system is similar to the CBMS in a sense. As a matter of fact, the ontology management system of this research does not consider context data. This research proposes a methodology for creation and adoption of domain ontologies for database design. The methodology is heuristic-based and focuses on identifying and defining the properties, relationships and constraints needed to model an application domain.

3. Context Base Modeling

3.1. Context Base and ontology

Context can include all information that describes objects, users, location etc. For example, in an information system, there is general context ontology which represents general concepts of contexts. At the same time, it is necessary to separate specific domains such as the business trip request domain. The separation of domains significantly reduces the scale of context, and thus releases the burden of context processing for pervasive devices in each domain (Gu et al., 2005). Figure 1 shows diagram of the definition of ontology.

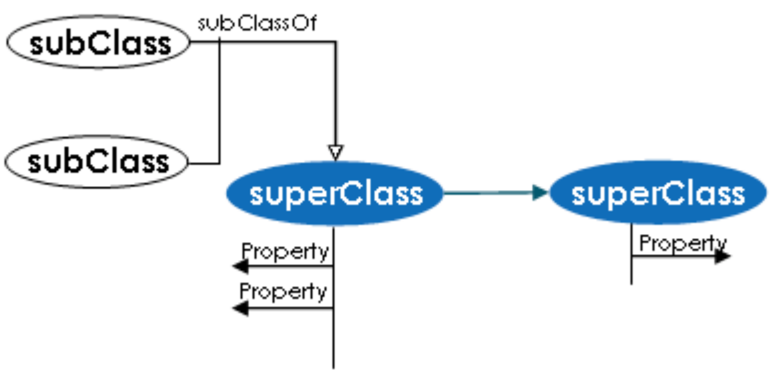


Figure 1. Diagram of definition of ontology

In Figure 1, the circles represent a superclass (upperclass) and a subclass. The white arrow represents the class hierarchy and the black arrow represents property of the class.

Context data can be represented by this definition. For example, 'Student', 'Faculty', and 'Staff' are subclasses of 'User' class which has properties such as age, SSN, etc, and 'Location' class has properties such as purpose, organization etc. Therefore all context data can be stored in appropriate classes and properties in the Context Base and the CBMS can modify it.

3.2. Ontology written in OWL

There are several ontology languages such as RDFS, DAML+OIL, OWL etc. In this research contexts are represented by ontologies written in OWL (Web Ontology Language). OWL was primarily designed to represent information about categories of objects and how objects are interrelated (Horrocks et al., 2003). OWL is more expressive than other ontology languages. In other words, OWL adds more vocabulary for describing properties and classes of RDF (resource description framework) resources, with semantics for generalization-hierarchies of such properties and classes. OWL is designed for use by applications that need to process the content of information instead of just presenting information to humans (McGuinness and Harmelen, 2004). There are more than 90 tools available for helping the use of ontologies from both non-commercial organizations and commercial software vendors. One of the most popular tools is Protégé which allows the user to construct a domain ontology, to customize data entry forms and to enter data etc. Using Protégé, Jiang et al. (2003) described a support system of ontology building in a clinical domain that integrated the linguistic knowledge information with the domain-specific context knowledge information. Zhou et al. (2004) used Protégé for ontology development of concepts and relationships that represent the domain and that will permit storage of knowledge. Protégé, which has the extensible components architecture and defines a flexible metaclass architecture and supports many formats such as RDF/RDFS, DAML+OIL, and OWL etc (Zhou, et al., 2004), integrates ontology editors. Therefore in this research, Protégé is deployed to build the Context Base.

3.3. CBMS

A framework of CBMS is shown in figure 2.

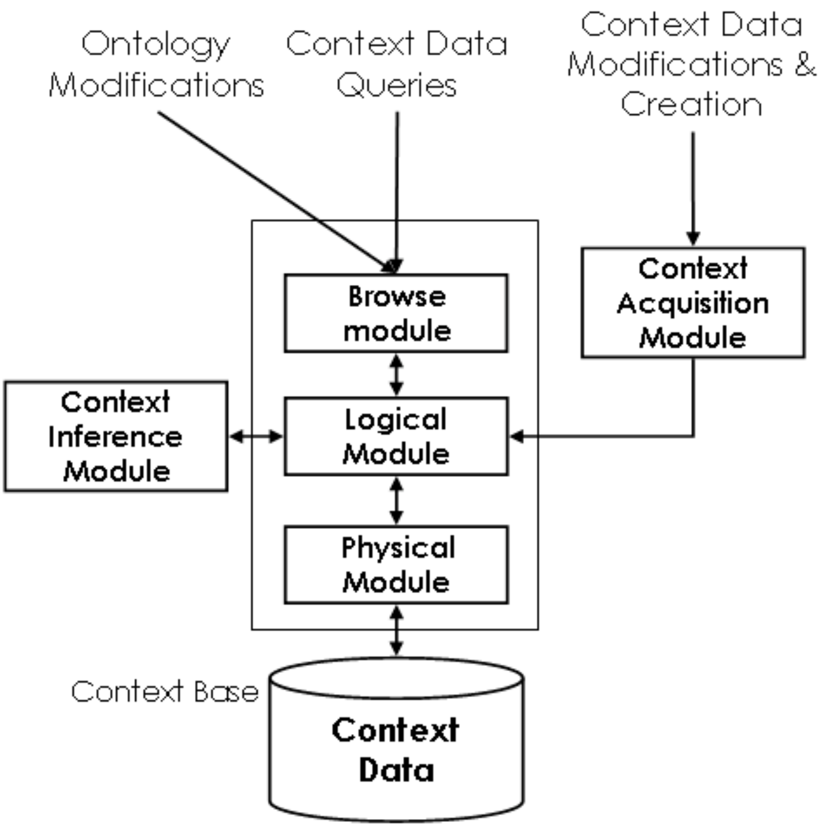


Figure 2. A CBMS Framework

3.3.1. Browse Module

Browse module provides systems, services, or information system administrators with the ability to view and access the context data. Through the browse module, ontology modifications or context data queries by services can enter the CBMS.

3.3.2. Logical Module

The logical module assists the CBMS in modifying, creating, and deleting context data. The logical module which receives context data from context acquisition module serves as an enabler for consistent ontology creation and management of existing ontologies. Furthermore, the logical module checks for consistency between ontologies.

3.3.3. Physical Module

The physical module provides context data for storage in the Context Base. If context data is created by sensors or systems, it is obtained by the context acquisition module and mapped to OWL automatically by physical module.

3.3.4. Context Acquisition and Inference Module

In the strict sense, the context acquisition module and context inference module are not included in the CBMS. However, these two modules are also important in providing context-aware services. Context acquisition module automatically obtains context data from federated sensors or systems. In other words, federated sensors such as RFID tags can acquire various context data and then, the context acquisition module obtains that data and sends it to the logical module. The context inference module provides reasoning of raw context data and retrieving new inferred context data which are used for context-aware services.

There are some systems such as DBMS and KBMS that are similar to CBMS. However, these 3 systems have different territories in which they are used. A KBMS is a development environment for knowledge-based systems that automates most of the processes used in building a knowledge-based application (Hicks, 2003). A KBMS provides full life-cycle support from knowledge acquisition to delivered code, in other words, the KBMS generates code for the knowledge-based system. Hicks (2003) described the KBMS architecture consisting of several parts. A DBMS is a collection of components that supports the creation, use, and maintenance of databases (Mannino, 2001). The DBMS provides efficient storage and retrieval of data.

However these systems are not suitable for handling context data in a ubiquitous computing environment. In ubiquitous computing environments, systems need real world information. Context data is not a set of data but information about the real world and how it operates. Databases and DBMSs can not handle real world information. Furthermore, context data is complex mainly because the context information is dynamic and acquired from multiple distributed sources (Kwon, 2004). As an example, we draw a classroom where there are several people one of whom, a student, is speaking in front of the classroom. Several sensors in the classroom can capture this context data, and a system needs to store and handle this context data. A DB and DBMS can only store and manage this information ostensibly. In order to store this information in a database, the DBMS has to create many relations therefore it is very inefficient to describe context data.

After the context data is stored, the system needs to query this context data. Both

DBMSs and CBMSs have a function which queries data. The DBMS only retrieves the same data as the one that was stored previously. However, the CBMS, which has a context inference module, can infer or reason on raw context data and retrieve new inferred context data. OWL has constructs which are not contained in traditional relational databases, such as relations, cardinality, equality, characteristics of properties, and enumerated classes (McGuinness and Harmelen, 2004). For example, using OWL's transitive property, if a user goes to Los Angeles on a business trip, the ontology-based CBMS recognizes the intentions of the user to go to the United States because Los Angeles is located in the United States. Therefore using OWL's characteristics, context data can be reasoned easily. The context inference module in a CBMS uses rule-based reasoning that uses predefined rules.

4. Prototype Implementation

We implemented a simple information system as a prototype. Example scenario is described in Section 4.1.

4.1. System Scenario

A suggested scenario for the prototype system is as follows.

An information system has a business trip request subsystem. If a person intends to go on a business trip, he/she must login to the information system and input data such as duration of stay, purpose, etc. The information system includes a Context Base which stores context data. This prototype contains only a business trip ontology domain, however in a real information system; more domains such as resources of an organization, marketing or financing etc. can exist. After the user inputs information about his/her business trip, the request system automatically calculates the expense of the trip and at the same time, if necessary, it sends reservation and entry fees to the air carrier, hotel, and conference organizers. Then, the system remits the rest of the expense money to the user.

4.2. Context Base Modeling

The screenshot of our prototype business trip request information system is shown in Figure 3. The information demonstrated in figure 3 is needed to build the Context Base and these contexts are represented by an ontology.

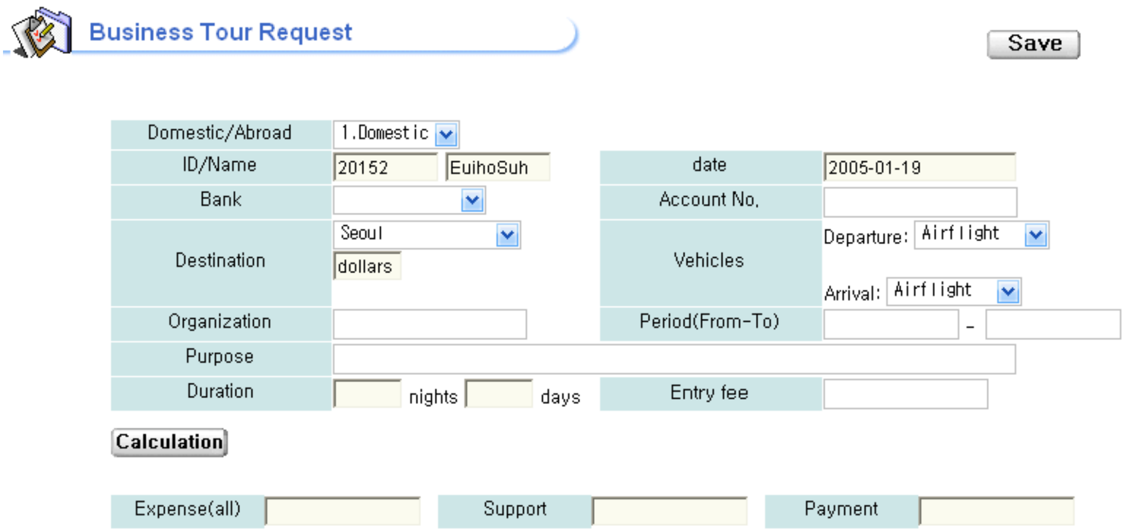
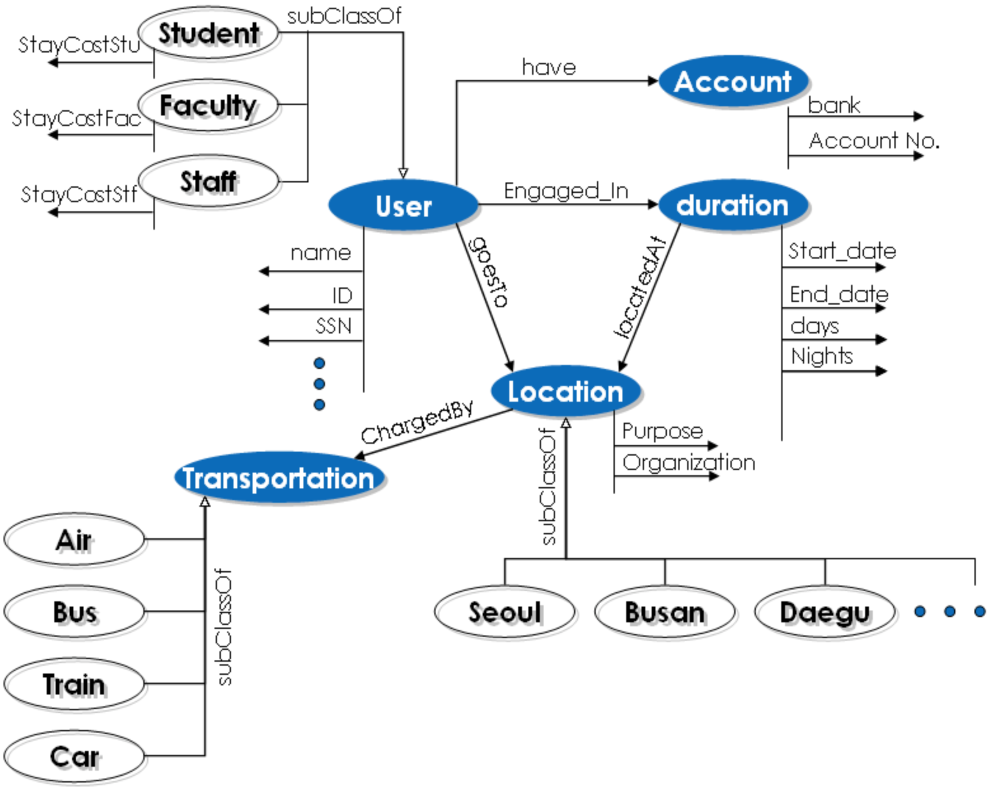


Figure 3. Screenshot of prototype business tour request system

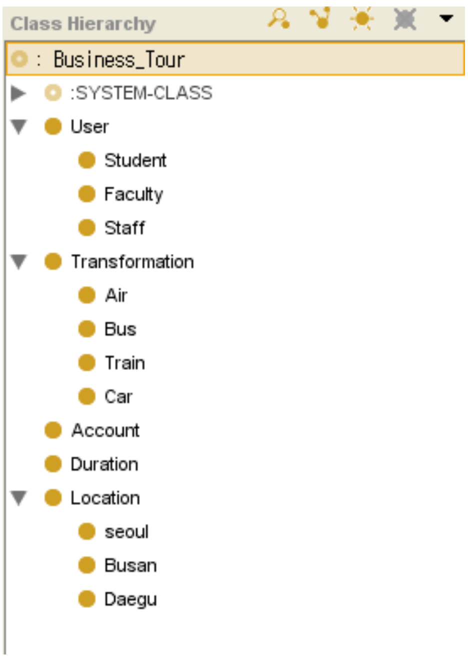
This prototype defines the ontology of the business trip as shown in figure 4(a), and the context ontology of the business trip domain is constructed using Protégé (figure 4(b)). There are several superclasses and subclasses and their properties. All classes are the subclasses of a 'Business tour' class (not shown in 4(a)). The 'User' class, which has 'Student', 'Faculty', and 'Staff' classes, and the 'Location' class, which has several 'City' (Seoul, Busan, etc.) classes, are the superclasses. We separated 'Student', 'Faculty' and 'Staff' classes because lodging costs are different according to status of user. The 'User' class has properties such as name, ID, SSN (social security number), etc. and the user has its own account that receives money for a business request. Therefore, we defined the entire ontology for the business trip as shown in figure 4(a).

We define some rules for constructing the ontology. Particularly, we should avoid cycles in the class hierarchy. If a class 'Car' has a subclass 'Bus' and at the same time, 'Bus' is a superclass of 'Car', there is a cycle. Maybe, some may think that the subclass of the location class is not a subclass of location, but instances of location. However, there are many business trips to one place at a time. For example user A goes to Seoul while users B and C also go to Seoul but they use different transportation. Then the 'Location' class has many similar instances which make ontology complex. Therefore we consider those instances as the subclasses of the 'Location' class.

Based upon the aforementioned description, context can be a converted ontology written in OWL. The CBMS automatically generates OWL code of the business trip context data. Figure 5 shows a partial context ontology written in OWL.



(a)



(b)

Figure 4. Definition of an ontology for business tour and class hierarchy

```
<rdf:RDF  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns="http://www.owl-ontologies.com/unnamed.owl#"
        xml:base="http://www.owl-ontologies.com/unnamed.owl">
…
<owl:Class  rdf:ID="Faculty">
<rdfs:subClassOf  rdf:resource="#User"/>
<owl:disjointWith  rdfs:resource="#Student"/>
<owl:disjointWith  rdfs:resource="#Staff"/>
</owl:Class>
…
<Faculty  rdf:ID="Euiho  Suh">
      <ID  rdf:datatype="http://www.w3.org/2001/XMLSchema#int">20152</ID>
</Faculty>
…
```

Figure 5. A partial context ontology written in OWL

Context acquisition module and context inference module are not focused on in this prototype. The prototype works as follows. A user that intends to go on a business trip has context data about the business trip. This context data can be entered into the logical module in the CBMS. Then the physical module generates OWL codes and stores them in the Context Base. In order to calculate the expense for the business trip, the request subsystem queries context data such as the user's position (faculty or student, etc.) or duration. The browse module plays a role in transferring context data from business trip request subsystem to the CBMS.

5. Conclusion

Building a Context Base is a significant area that plays an important role in ubiquitous computing environments. As context data is represented by ontology, context data can be used successfully for services or devices.

We developed a prototype system that uses a Context Base, however, our framework is not fully implemented in the prototype. It still lacks a context acquisition module and context inference module. However, this prototype represents the current core concept of a context base and a CBMS, there has been no uniform framework of a Context Base and CBMS developed. As context aware systems are introduced more and more, the size of context data will increase. Hence, a stable and well-structured context base will be needed.

This research can be applied not only to an information system, but also to context-aware services. As context-aware services develop, the necessity of a uniform framework of a Context Base and CBMS will grow. Performance experiments on a Context Base and CBMS are not performed in this research since it is not focused on the service area. Nevertheless, the Context Base and CBMS will be more efficient than traditional databases and DBMSs in ubiquitous computing environments.

References

Barkhuus, L. and Dey, A. K. (2003). Is context-aware computing taking control way from the user? Three levels of interactivity examined. *Proceeding of the 5th international conference on ubiquitous computing, Seattle, USA*

Berners-Lee, T., Hendler, J., Lassila and O. (2001). The semantic web, *Scientific American*

Dey, A. K. and Abowd, G. D. (1999). Towards a better understanding of context and context-awareness. *First International Symposium on Handheld and Ubiquitous Computing (HUC'99)*

Dey, A. K., Abowd, G. D. and Salber, D. (1999). The Context Toolkit: Aiding the Development of Context-Enabled Applications. *Proceedings of the 1999 Conference on Human Factors in Computing Systems (CHI '99)*, (pp. 434-441)

Dey, A. K., Abowd., G. D. and Salber, D. (1999). A Context-Based Infrastructure for Smart Environments. *Proceedings of the 1st International Workshop on Managing Interactions in Smart Environments (MANSE '99)*, (pp. 114-128)

Gandon, F. L. and Sadeh, N. M. (2004). Semantic web technologies to reconcile privacy and context awareness. *Web Semantics: Science, Services and Agents on the World Wide Web, 1(3),* 241-260

Gu, T., Pung, H. K. and Zhang, D. Q. (2005) A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications, 28,* 1-18

Gruber, T. R. (1993). A translating approach to portable ontology specification, *knowledge acquisition* 5, 199~220

Hicks, R. C. (2003). Knowledge base management systems-tools for creating verified intelligent systems. *Knowledge-Based Systems 16,* 165-171

Hong, J.I. and Landay, J. A. (2001). An Infrastructure Approach to Context-Aware Computing. *Human-Computer Interaction (HCI)*, 2001. 16, xxx-xxx

Horrocks, I., Patel-Schneider, P. F. and Harmelen, F. V. (2003). From SHIQ and RDF to OWL: the making of a Web Ontology Language. *Web Semantics: Science, Services and Agents on the World Wide Web, 1,* 7-26

Jiang, G., Ogasawara, K., Endoh, A. and Sakurai, T. (2003). Context-based ontology building support in clinical domains using formal concept analysis. *International Journal of Medical Informatics, 71*, 71-81

Khedr, M. and Karmouch, A. (2005). ACAI: agent-based context-aware infrastructure for spontaneous applications. *Journal of Network and Computer Applications, 28,* 19-44

Kindberg, T., Barton, J. (2001). A web-based nomadic computing system. *Computer Networks: The International Journal of Computer and Telecommunications Networking.* 35(4), 443-456

Kwon, O. (2004). Modeling and generating context-aware agent-based applications with amended colored Petri nets. *Expert Systems with Applications*, *27*, 609-621

Kwon, O., Yoo, K. and Suh, E. (2005). UbiDSS: a proactive intelligent decision support system as an expert system deploying ubiquitous computing technologies. *Expert Systems with Applications 28*, 149-161

Mannino, M. V. (2001), Database design, application development and administration, 2nd edition. *The McGraw-Hill Companies*

Mcguinness, D.L., Fikes, R., Hendler, J. and Stein, L. A. (2002). DAML+OIL: an ontology language for the Semantic Web*. IEEE Intelligent Systems*, 17, 72-80

McGuinness, D. L. and Harmelen, F. V. (2004) OWL Web Ontology Language Overview. *http://www.w3.org/TR/owl-features/*

Prekop , P. and Burnett, M. (2003) Activities, context and ubiquitous computing. *Computer communications*, 26 1168-1176

Ranganathan, A. and Campbell, R. H. (2003). A Middleware for Context-Aware Agents in Ubiquitous Computing Environments. *ACM/IFIP/USENIX International Middleware Conference*

Sugumaran, V. and Storey, V. C. (2002). Ontologies for conceptual modeling: their creation, use, and management. *Data & Knowledge Engineering 42*, 251-271

Want, R., Hopper, A., Falcao, V. and Gibbons, J. (1992). *The active badge location system. ACM Transactions on Information System, 10(1)*, 91-102

Zhou, X., Wu, Z., Yin, A., Wu, L., Fan, W. and Zhang, R. (2004). Ontology development for unified traditional Chinese medical language system, *Artificial Intelligence in Medicine 32,* 15-27