Ontology-based Context Modeling for User-Centered Context-Aware Services Platform

Anusuriya Devaraju
Faculty of Information and Communication Technology,
University Technical Malaysia Melaka
anusuriya@utem.edu.my

Simon Hoh
British Telecommunications Plc
(Malaysian Research Centre)
simon.hoh@bt.com

Abstract

Pervasive computing world composed of plethora of devices using different technologies, resources, connectivity capabilities; and these devices or associated applications meet difficulties in interpreting and integrating knowledge or context information coming from different sources. To tackle this problem, we have developed a middleware architecture called Context-aware Service Platform or CASP, to relieve the context-aware services from the burden of integrating, managing and querying context information from various sources. In this paper, we briefly introduce our context-aware middleware framework; this is followed by detailed descriptions on the context modeling approach based on ontology and the integration of the ontology-based information model into our platform. Following this, we present how the captured raw data are populated on our context model, and relevant contexts can be queried via the Context Query API.

1. Introduction

Pervasive Computing aims at creating an environment where the connectivity of devices is embedded in such a way that the connectivity is unobtrusive and always available [6]. Context-awareness is a fundamental aspect of the pervasive computing paradigm because it provides the ability for solutions to be aware to the situation a user is in, thus providing the ability for such solutions to react around the users every need. In order to achieve such goals, many issues that surrounds how context information can be gathered, represented, processed and consumed appropriately by the solution needs to be looked at.

Context-Aware Service Platform or CASP is an ongoing research project, aimed at constructing a middleware-based infrastructure to simplify the development of context-aware services by providing a common set of functionalities; functionalities that enable raw sensory information can be acquired and translated into knowledge so that appropriate effectors, hardware and software can be employed.

This paper mainly focuses on the modeling of contextual information based on the ontology using OWL to resolve the issues of semantic representation on our context middleware. The rest of the paper is structured as follows. Section 2 gives a short overview of context modeling approaches known from research literature; this followed by basic definitions/concepts related to ontology-based information modeling. Section 3 discusses the three layered of ontological framework developed in this study. Section 4 introduces the basic building blocks of CASP and integration of ontology model into the middleware. This includes descriptions on how the captured raw data are populated on our ontology model, and relevant contexts can be queried via the Context Query API. Finally, we conclude the paper with a short summary and give an outlook towards future work.

2. Basic Concepts and Definitions

This section provides definitions of terms related to context modeling using ontologies. *Context modeling* is the specification of all entities and relations between these entities which are needed to describe the context as a whole, for example, information on location, time, the user and its current or planned activity, and computational entities, while, *Context reasoning* means to automatically deduce further, previously implicit facts from explicitly given context information [3].

2.1. Existing Context Modeling Approaches

A significant body of related research has already been carried out in evaluating approaches for modeling contexts. [20] have evaluated six context modeling approaches, against the requirements of pervasive services, which are *distributed composition (dc), partial validation (pv), richness and quality of information(qua), incompleteness and ambiguity(inc), level of formality(for)* and *applicability to existing environments(app)* (refer Table 1).

Table 1. Context modeling schemes against the pervasive computing requirements

Approach - Requirem.	dc	pv	qua	inc	for	app
Key-Value Models	-	-	–	–	–	+
Markup Scheme Mod.	+	++	-	-	+	++
Graphical Models	–	-	+	-	+	+
Object Oriented Mod.	++	+	+	+	+	+
Logic Based Models	++	-	-	-	++	–
Ontology Based Mod.	++	++	+	+	++	+

Based on the evaluation conducted, [20] concluded that the most promising assets for context modeling for ubiquitous computing environments with respect to the requirements identified can be found in the 'ontology' approach. Similar studies have been performed by [13][14], who classified modeling techniques for context information into the above six different categories and benchmarked according to their support with regard to three levels for information interoperability requirements. [13] agrees that ontology-based approaches offer more capabilities for satisfying the information requirements at schema and service levels. In this study, we use ontologies to specify context information and the relationships among contexts because ontologies offer rich expressiveness and support evolutionary aspect of context modeling.

A widely accepted definition of ontologies was established by T.R. Gruber (1993) - *'Ontology is a formal specification of a conceptualization'*. An ontology specification is a formally described, machine-readable collection of terms and their relationship expressed with a language in a document file. A conceptualization refers to an abstract model of a domain that identifies a concept [7].

There are different languages which are used to define ontologies, e.g. Ontolingua [19], LOOM [11] and OWL Web Ontology Language [8, 12]. OWL is a family of knowledge representation languages for authoring ontologies, and is endorsed by the World Wide Web Consortium. OWL ontology consists of a set of axioms describing classes, properties, and the relationships between them. RDF/XML is used for marking up conforming instance data [7]. OWL is divided into three increasingly expressive sub languages OWL-Lite, OWL-DL and OWL-Full. OWL-DL is most often used because it provides maximum expressiveness (in opposite to OWL-Lite) without losing computational completeness and decidability (in opposite to OWL-Full) [3].

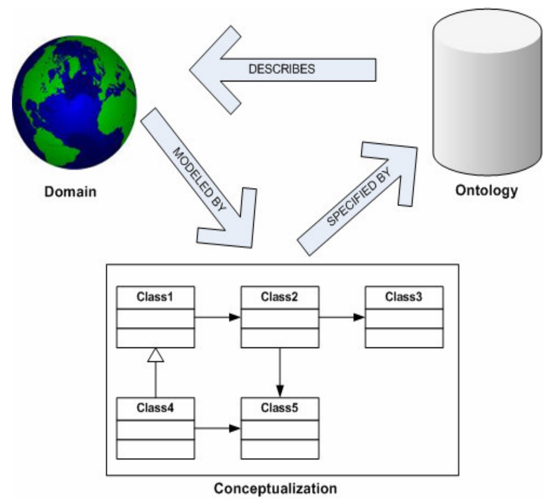

Figure 1. Gruber ontology definition [16]

3. Layered Ontological Framework

We propose three layered ontological framework for modeling contextual information. The ontological framework comprises of upper level, mid-level, and lower level ontology (refer Figure 2). We use Protégé [10] - a free, open source ontology editor and knowledge-base framework, to work on context modeling with OWL.

In our study, we have adopted *Suggested Upper Merged Ontology (SUMO)* [17] as the upper ontology that provides definitions for general-purpose terms and acts as a foundation for more specific mid-level and lower or domain ontologies. SUMO provides a structure and a set of general concepts upon which domain ontologies could be constructed. SUMO has been selected as upper ontology because it is the working paper of an IEEE-sponsored open-source standards effort. This means that users of the ontology can be more confident that the ontology will eventually be embraced by a large class of users. Further, SUMO was constructed with reference to very pragmatic principles. Any distinctions of strictly philosophical interest have been removed from the ontology [9].

The stated upper ontology has been expanded to include our *Mid-Level Ontology* or MILO which holds context entities representation. MILO acts as a ‘bridge’ between the abstract content of SUMO and the rich detail of the various subordinate domain-specific ontologies. Some of the concepts derived from SUMO into MILO are *region, process, number, and physical quantity*. Our strategy for developing MILO is to borrow terms from existing ontologies but not to import them directly. The design of MILO is inspired from existing context ontologies such as CONON [21] and SOUPA [4]. The Composite Capability/Preference Profiles (CC/PP)[1], a specification for defining capabilities and preferences of user agents has been implemented to extend *‘Resource’* entity in MILO ontology.

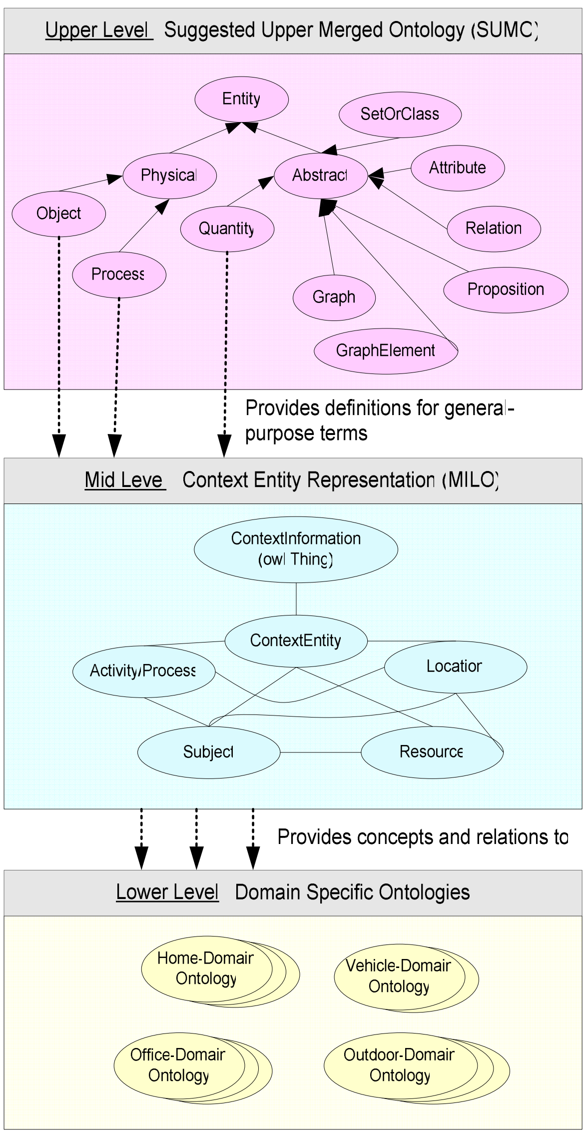

Figure 2. Three-layered ontology architecture

The lower-level contains collection of ontologies that provides the domain specific knowledge to context-aware applications such as weather service, smart home service and so on. Currently, we have designed 4 domain specific ontologies, which are the home, office, vehicle and outdoor ontologies.

The Figure 4 is a graphical representation of simplified MILO ontology. Parts of the constituting context entities, concepts and their mutual relationships are presented.

3.1. Related Ontologies

During initial study, we have surveyed the existing ontology-based models with the aim of identifying models that could be used as a basis in our context ontology. In the following section, we describe the key features of the related ontologies.

a. *Suggested Upper Merged Ontology (SUMO)[17]*
SUMO was developed within the IEEE Standard Upper Ontology Working Group. The goal of this Working Group is to develop a standard ontology that will promote data interoperability, information search and retrieval, automated inferencing, and natural language processing. The following Figure 3 presents SUMO’s highest level concepts. The root node of the SUMO is, as in much ontology, *‘Entity’*, and this concept subsumes ‘*Physical’* and *‘Abstract’*.

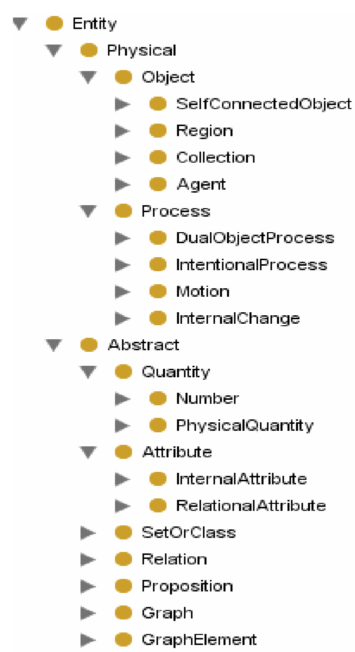

Figure 3. SUMO’s high level concepts

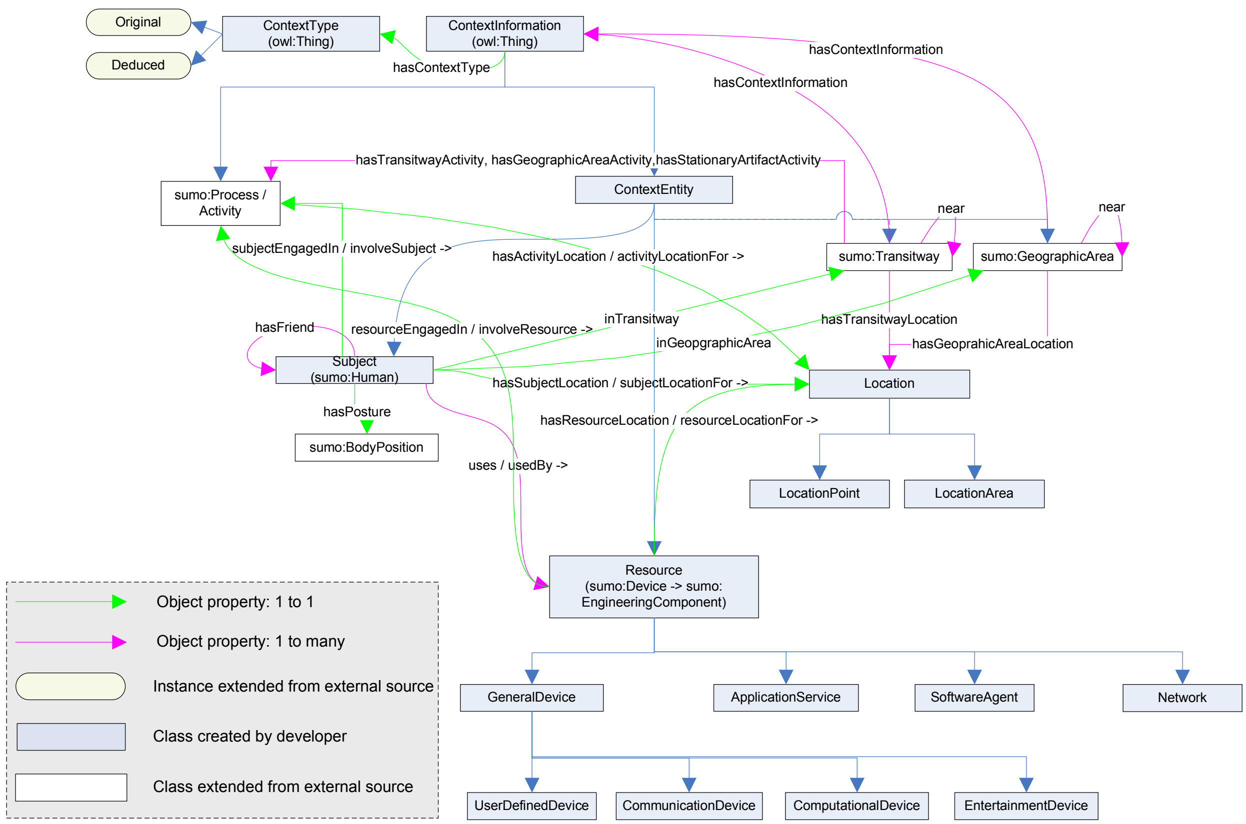

Figure 4. Mid-level ontology (MILO)

b. Ontology based Context Modeling and Reasoning using OWL (CONON)[21]

CONON is OWL encoded context ontology for modeling context in pervasive computing environments, and for supporting logic-based context reasoning. CONON provides an upper context ontology that captures general concepts about basic context, and also provides extensibility for adding domain-specific ontology in a hierarchical manner.

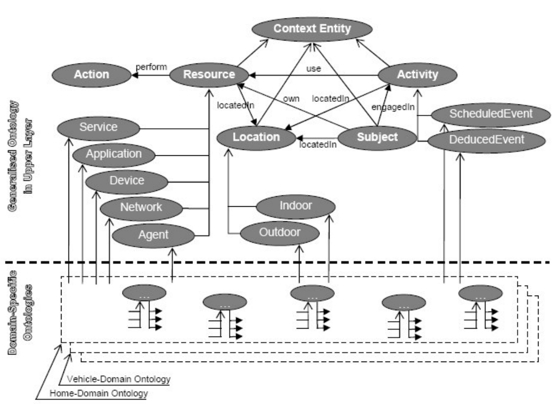

Figure 5. CONON

c. Standard Ontology for Ubiquitous and Pervasive Applications (SOUPA)[4]

SOUPA ontology is expressed using the OWL and provides a shared ontology that combines many useful vocabularies from different consensus ontologies. Ontologies referenced by SOUPA can be found at [4]. SOUPA consists of two distinctive but related set of ontologies: SOUPA Core and SOUPA Extension. The set of the SOUPA Core ontologies attempts to define generic vocabularies, while the set of SOUPA Extension ontologies define additional vocabularies for supporting specific types of applications and provide examples for defining new ontology extensions.

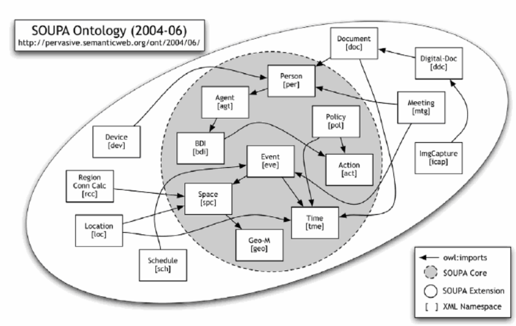

Figure 6. SOUPA

4. Context-Aware Services Architecture (CASP)

The architecture of CASP is modeled after design consideration of three subsystems called *Sensing, Thinking* and *Acting*.

The *Sensing* subsystem abstracts sensor types that exist in the physical or non-physical world, to provide a consistent interface method for raw data to be programmatically fed into the CASP via its Sensor Data Acquisition API.

The *Thinking* subsystem is made up of an ontology engine that attempts to map sensory information in the real world to object instances in a virtual world that computer systems can interact with. Here, a Context Query API has been developed to enable services or context consumers to acquire the appropriate knowledge that is relevant to the service being executed.

The *Acting* subsystem consists of the Service Execution Engine, its Service Creation API and the services itself. Each service conforms to a service wrapper framework requirement that requires it to define the context which would consequently trigger a response.

Figure 7 shows how the subsystems that can be abstracted into our infrastructure. Further information on the architecture can be found in our earlier papers Devaraju, A. (2007), Simon, H. (2008).

4.1 Integrating Context Ontology into CASP

When platform reboots, the Ontology Manager (OM) will create a 'virtual' ontology model based on local OWL file. Requests will be sent to all registered sensors in order to update their state - 'online' or 'offline'. On receiving a sensor registration request, the OM registers the sensor profile in the local data store. When the sensor becomes 'active', it reports data to the platform. An XML-based sensor data messaging protocol has been developed for sensor registration and management, and raw data reporting. More information on the sensor data acquisition component is available here [2].

The ontology engine maps the raw data (captured via Sensor Acquisition API) to object instances in the 'virtual' model. The object instances will be updated as new sample data received. For 'active' sensor, when the OM does not receive sensor data within a certain time frame, it considers the sensor to be 'offline'. It updates the sensor state in the sensor database, and then triggers an event and dispatches it to the Context Query (CQ) component to inform about the status.

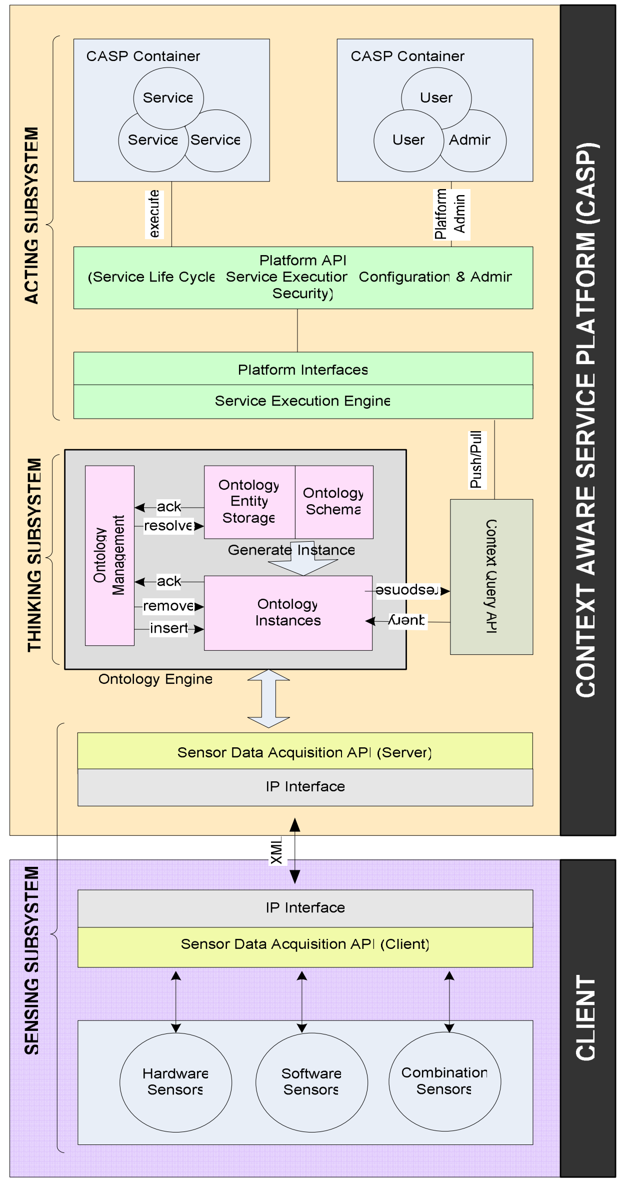

Figure 7. CASP architecture

4.2. Context Query Support

We use Jena Semantic Web Framework [5] as a Java API to access the OWL ontologies. Jena provides a programmatic environment for RDF, RDFS and OWL, and a rule-based inference engine. Jena also includes a SPARQL [16] - a data-oriented query language that searches the model and returns relevant information as a graph or a set of variables. Most forms of SPARQL query contain a set of triple patterns called a basic graph pattern. Triple patterns are like RDF triples except that each of the *subject, predicate* and *object* may be a variable [5].

The ontological framework provides mechanism for context-aware services to both retrieve context data either on periodic basis (push mode) or when queried (pull mode). The CQ supports 'trigger operation' to handle 'push' requests for context information from the subscribed services. An XML-based messaging protocol has been developed to enable context retrieval between Service Execution Engine (SEE) and Context Query (CQ) component.

The context query is performed as follows. Firstly, an ontology-based context model is created via the ModelFactory; SEE translates the request parameters into an XML request format (refer Figure 8). The context query format includes parameters such as *primary entity* which specifies the entity used as origin of query, and *interested entity* which determines the expected context information. When the CQ received XML-based request, it parses the requests, and then translates the elements into appropriate *triple pattern* to search for the entity and its relevant contexts from the model. Finally, the result model will be composed as XML response format and then forwarded to the service via SEE.

```
<?xml version = "1.0"?>
<casp>
<type>...</type>
<serviceid>....</serviceid>
<PrimaryEntity name ="SUBJECT">
    <param name="N" value="Annie"/>
    <param name="FN" value="Chan"/>
</PrimaryEntity>
<InterestedEntities>
    <entity name ="..." mode="partial">
        <params name="param1">Value1</params>
        <params name="param2">Value2</params>
    </entity>
    <entity name ="..." mode="partial">
        <params name="paramA">valueA</params>
        <params name="paramB">valueB</params>
        <params name="paramC">valueC</params>
    </entity>
  </InterestedEntities>
</casp>
```

Figure 8. Context query XML

5. Future Work and Conclusions

In this paper, we have presented a context middleware to support integration, management and querying of context information from various sources. Our three-layered ontology model is semantically rich and extensible to accommodate context information that is useful in building pervasive applications. The main components of the platform (sensing, thinking and acting subsystems) are 'loosely-coupled' to promote extensibility of the framework. The context information can be queried via flexible and expandable query format that can be changed at run time, which is simple enough to be easily utilized by the developers.

Our future works mainly focus on the following research areas. The current context model will be extended to represent a range of characteristics/quality context information and uncertain, partial or outdated context information. The Ontology Engine will be enhanced to implement complex OWL reasoning rules. We believe that this extension of work will provide the inference capabilities of context information from other sensory sources when the solution is not able to locate its primary source of context.

6. References

[1] CC/PP- Composite Capabilities/Preferences Profile, 2006, URL: www.w3.org/Mobile/CCPP/

[2] Devaraju A., Hoh S., and Hartley M., "A Context Gathering Framework for Context-Aware Mobile Solutions", International Conference on Mobile Technology, Applications and Systems, Singapore, 2007.

[3] Feruzan Ay, "Context Modeling and Reasoning using Ontologies", University of Technology Berlin, July 2007.

[4] H. Chen, F. Perich, T. Finin, and A. Joshi, "SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications", International Conference on Mobile and Ubiquitous Systems: Networking and Services, Boston, MA, 2004.

[5] Jena Semantic Web Framework, URL: jena.sourceforge.net/

[6] Kolos-Mazuryk, L. and van Eck, P.A.T. and Wieringa, R.J., "A Survey of Requirements Engineering Methods for Pervasive Services", Freeband A-MUSE deliverable D5.7a TI/RS/2006/018 Freeband, Enschede, 2006.

[7] Lee W.Lacy, "*OWL: Representing Information Using the Web Ontology Language",* Trafford Publishing, January 1, 2005.

[8] M. K. Smith, C. Welthy, and D. L. McGuinness: "OWL Web Ontology Language Guide", URL: http://www.w3.org/TR/owl-guide, 2004.

[9] Niles, I., and Pease, A. "Towards a Standard Upper Ontology", In Proceedings of the 2nd International Conference on Formal Ontology in Information Systems, Maine, USA, 2001.

[10] Protégé – Ontology Editor and Knowledge Acquisition System, URL: http://protege.stanford.edu/

[11] R. M. MacGregor: "Inside the LOOM Description Classifier", in ACM SIGART Bulletin, 1991.

[12] S. Bechhofer, F. v. Harmelen, J. Hendler, I. Horrocks, et al., "OWL Web Ontology Language Reference", URL: www.w3.org/TR/owl-ref/

[13] Serrano, J. M. Serrat, J. van der Meer, S. Foghlu, M. O., "Ontology-Based Management for Context Integration in Pervasive Services Operations", First International Conference on Autonomous Infrastructure, Management and Security, Oslo, Norway, June 21-22, 2007.

[14] Serrano, J.M., Serrat, J., "Context Modelling and Handling In Context-Aware Multimedia Applications", IEEE Wireless Magazine, Multimedia in Wireless/Mobile Ad-hoc Networks, 2006.

[15] Simon H, Devaraju A, Chin Chin W, "A Context Aware Framework for User Centered Services", 21st International Symposium on Human Factors in Telecommunication, Kuala Lumpur, 2008.

[16] SPARQL Query Language for RDF, URL: www.w3.org/TR/rdf-sparql-query/

[17] SUMO- Suggested Upper Merged Ontology, URL: http://www.ontologyportal.org/

[18] T.R. Gruber, "A Translation Approach to Portable Ontology Specifications", *Special issue: Current issues in knowledge modeling*, Volume 5, Issue 2, Academic Press Ltd, June 1993.

[19] T. R. Gruber: "Ontolingua: A Mechanism to Support Portable Ontologies", Stanford University, 1992.

[20] Thomas Strang, Claudia Linnhoff-Popien, "A Context Modeling Survey", Workshop on Advanced Context Modelling, Reasoning and Management as part of UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England, September 2004.

[21] X. H. Wang, T. Gu, D. Q. Zhang, and H. K. Pung, "Ontology Based Context Modeling and Reasoning using OWL", Pervasive Computing and Communications Workshops, 2004.