A Unified Context Model: Bringing Probabilistic Models to Context Ontology*

Binh An Truong, YoungKoo Lee**, and Sung Young Lee**

Department of Computer Engineering, KyungHee University,
Giheung-Eup, Yongin-Si, Gyeonggi-Do, 449-701, Korea
tabinh@oslab.khu.ac.kr, {yklee, sylee}@khu.ac.kr

Abstract. Ontology is a promising tool to model and reason about context information in pervasive computing environment. However, ontology does not support representation and reasoning about uncertainty. Besides, the underlying rule-based reasoning mechanism of current context-aware systems obviously can not reason about ambiguity and vagueness in context information. In this paper, we present an ongoing research on context modeling which follows the ontology-based approach while supports representation and reasoning about uncertain context. This unified context model then is used as a framework in our implementation of the context management and reasoning module of our context-aware middleware for ubiquitous systems.

1 Introduction

Most of the current proposed pervasive context-aware systems use sensors as the major source for providing data to applications. However, the data sensed, or raw data, is always imperfect and incomplete due to the sensing technologies. This result in the inaccurateness of the high-level information deduced from raw data. For example, it is difficult to infer that the user is sleeping based on the sensing data such as his location (in-bed), the room light (dark) and the sound (quiet). Furthermore the underlying logical, rule-based reasoning mechanism of current systems obviously does not support reason about uncertainty. Hence, dealing with uncertainty is the most challenge in context-aware computing research community.

Since its appearance, probabilistic model or Bayesian networks technique has showed to be a very powerful tool for the representation and reasoning about the uncertainty. In particular, a Bayesian network represents a full joint distribution over a set of random variables. It can answer queries about any of its variables given any evidence. Besides, Bayesian network provides different forms of reasoning including: prediction (reasoning from cause to result), abduction (inferring cause from result) and finally, explaining away (the evidence of one cause reduces the possibility of another cause given the evidence of their common results) which is especially difficult to model in rule-based systems [4]. Nevertheless, a fundamental limitation of

* This work is partially supported by Korea Science and Engineering Foundation (KOSEF).
** Corresponding authors.

T. Enokido et al. (Eds.): EUC Workshops 2005, LNCS 3823, pp. 566 – 575, 2005.

using Bayesian network for knowledge representation is that it can not represent the structural and relational information. Also, the applicability of a Bayesian network is largely limited to the situation which is encoded, in advance, using a set of fixed variables. Thus, it is not suitable for representation of contextual data which is highly interrelated and dynamic in pervasive computing environment [5].

In this paper, we present a unified context model which inherits the advantages from both the probabilistic model and ontology. It can be considered as the glue which connects and integrates these two techniques. Given the unified context model, we can build a unified context ontology which captures both structural and probabilistic knowledge of a domain. Given the unified context ontology, Bayesian networks are constructed autonomously and used for reasoning about uncertainty.

The rest of paper is organized as follows. Section 2 discusses on the related work. Section 3 presents a scenario which will be used through this paper for illustrating of our approach. In section 4, we present our unified context model in detail. The unified context-ontology paradigm is introduced in section 5. Section 6 introduces three types of reasoning supported given the context ontology defined based on our context model. Finally, the paper ends with discussions and conclusions in section 7.

2 Related Work

A lot of research has addressed the issue of uncertainty in context aware computing. First efforts tried to modeling the uncertainty of context information using various terms such as "imperfectness" [8], "confidence" [9], "accuracy" [10], etc. Nevertheless, those approaches lack of expressiveness to capture rich types of context information and they do not support the reasoning mechanism.

Recently, some research approaches have used Bayesian networks to model and reason about the uncertain context. Firstly, Ranganathan et al. [2] used Microsoft's Belief Network (MSBN) [12] software to create the Bayesian networks structure. The Bayesian network is defined by knowledge experts and is mapped to predicates in the ontology by developers. Each predicate is attached with a confidence value for representing its value's uncertainty. Secondly, Tao Gu et. al. [11] mapped each context predicate into a node in the Bayesian network. Then, an arc is drawn between two nodes if and only if a dependency relation exists between two context predicates. Thus, a RDF graph with dependency markup is translated into a Bayesian network.

These two approaches have solved the problem of dealing with uncertainty. However, their support of uncertain reasoning is application-specific. In both approaches, developing a new application needs to redefine a new Bayesian network even if the domains are similar. Also, the mapping between the Bayesian network and the ontology is done manually by developers in both approaches. Even when the probabilistic data in form of Bayesian networks is integrated into the context ontology in [11], it is still unable to be reused for a similar domain. The reason is that it is defined over the instances data. In summary, both approaches provide no systematical method to support the uncertain reasoning mechanism.

3 A Smart-Home Scenario

In this section, we describe a smart home scenario which will be used for illustrating our proposed context model. Our sample scenario is a smart automated home that can proactively control the environmental conditions to reduce resource consumption. The windows and blinds can be controlled automatically according to the situations to provide optimal cooling or heating process or to create a fresh air breeze. For instance, on a day when the temperature is shifted from cool to warm, the home might determine that the optimal warming strategy is to open the windows and blinds so that the warm air can go inside. This scenario seems to be very simple but it is practically more complicated in the real situation. For example: the outside is so noisy while there are people reading inside the room; someone does not want the blinds open because he/she is sleeping; the air outside is polluted by dust and smoke, and so on. The decision of open the windows and blinds is depended on many elements in the situation. Such dependencies are also changed from situation to situation.

In the next section, we will show how to model this smart-home domain based upon our proposed unified context model

4 The Unified Context Model

Our context model is influenced mainly by the Probabilistic Relational Model developed by Friedman et al. in [8] and the Probabilistic Frame-based systems of Koller et al. in [9]. We made some modifications in compare with the original PRM to make the model simple and suitable to our requirements.

The unified context model consists of two parts:

- *The relational schema* which represents the structural and organizational information in forms of class, binary relations, relation chains and properties.
- *The probabilistic models* which annotate the conditional probabilistic dependency relationship between properties of classes.

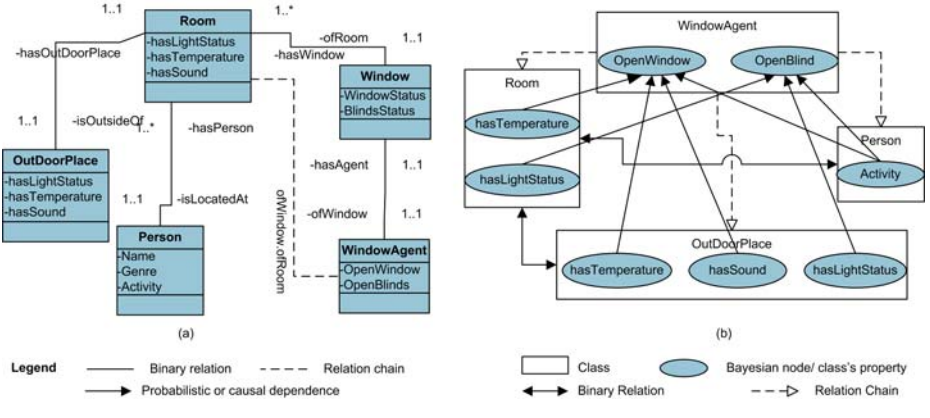

Fig. 1. (a) The relational schema and (b) the probabilistic model for the scenario

4.1 The Relational Schema

The basic unit of our context model is a class X. A class may be a sub-class of another (its super class). A class includes a set of relations $R_1, \ldots, R_n$ and a set of properties $P_1, \ldots, P_n$ with associated restrictions (or facet). A relation R_i specifies a binary relation between two classes X and Y. All relations are typed appropriately. The binary relation X.R(Y) can be considered as an object-property of class X which has the value-type of class Y. In Figure 1, the binary relation hasWindow defines the relationship between the class Room and the class Window.

A relation-chain is a sequence of binary relations separated by period. It creates an implicit relational link between two classes in a relational domain. A relation-chain $X.R_1.R_2 \ldots R_n$ refers to the final class which is the type of the final relation in the chain. Each relation R_i in the chain must be correctly typed. In Figure 1, the chain ofWindow.ofRoom creates an indirect relation between class WindowAgent and class Room. Similarly, a property-chain is formed by appending a relation-chain with a property of the referenced class. It specifies a reference from a class to a property, which can be its property or other class property.

4.2 A Modified Probabilistic Relational Model

We use probability for representing the uncertainty within a domain. A class which consists of probabilistic information is annotated with the local probabilistic model. This type of class is called p-class. A p-class, similarly to the normal class, has properties, relations and restrictions.

We call the property which contains probabilistic information a p-property. A p-property is either simple or complex. A p-class may also have other properties that do not participate in the probabilistic model, whose type is neither of the above. This feature allows existing knowledge bases to be annotated with probabilistic information without requiring a complete redesign of the ontology.

A simple p-property corresponds with a root node in Bayesian network. A simple p-property has two restrictions: hasValue and hasPD. The restriction hasValue is an explicitly enumerated list of possible values for the p-property. The restriction hasPD specifies the probability distribution over the values listed in the hasValue restriction. For example, the p-property hasTemperature of the class Room may have the restriction hasValue as *{Hot, Warm, Cool, Cold}* and the restriction hasPD as *{0.3, 0.25, 0.25, 0.3}*. The sum of all probability values listed in a restriction hasPD must be equal to 1 to satisfy the probability axioms.

A complex p-property corresponds with a Bayesian network's node which has a set of parent nodes. Beside the two restrictions, hasValue and hasPD, a complex p-property has two other restrictions, hasParents and hasCPT, which specify the conditional probabilistic dependencies on other p-properties. The hasParents restriction of the complex p-property P specifies a list of property-chains on which the value of this property depends. Each property-chain refers to one property of other class. For example, in the WindowAgent class, the parent of the p-property OpenWindow may be the property-chain ofWindow.ofRoom.hasTemperature. The hasCPT restriction specifies the conditional probability distribution over the values of

the property given values of its parents, which are listed by the `hasParents` restriction. The conditional probability distribution is represented by using a conditional probability table (CPT) as in Bayesian networks. For each combination of values of its parents, the CPT provides a probability distribution over values of the property given its parents. For simplicity, we assume that the CPTs are represented as fully specified functions of parent values.

5 The Unified Context Ontology

Based on the proposed unified context model, we build a unified context ontology to capture the knowledge of the smart home domain as described in section 2. The p-class can be used just like any other normal class. We can create instances of class, which inherit all of its template properties and restrictions. In particular, the probability distribution over values of properties of the instance will be described in the p-class. Similarly, the inheritance mechanism can be used to make one p-class a subclass of another. A subclass can extend the definition of the super-class as well as overwrites parts of it. It can redefine the probabilistic model of one or more p-property.

The unified context ontology should be able to captures all the characteristics of context information. We classify the pervasive computing domain into a collection of sub-domains such as smart-home domain, smart-office domain, smart-university domain, etc. It would be easy to specify the context in one domain in which a specific range of context is of interest. The separation of domains can also reduce the burden of context processing.

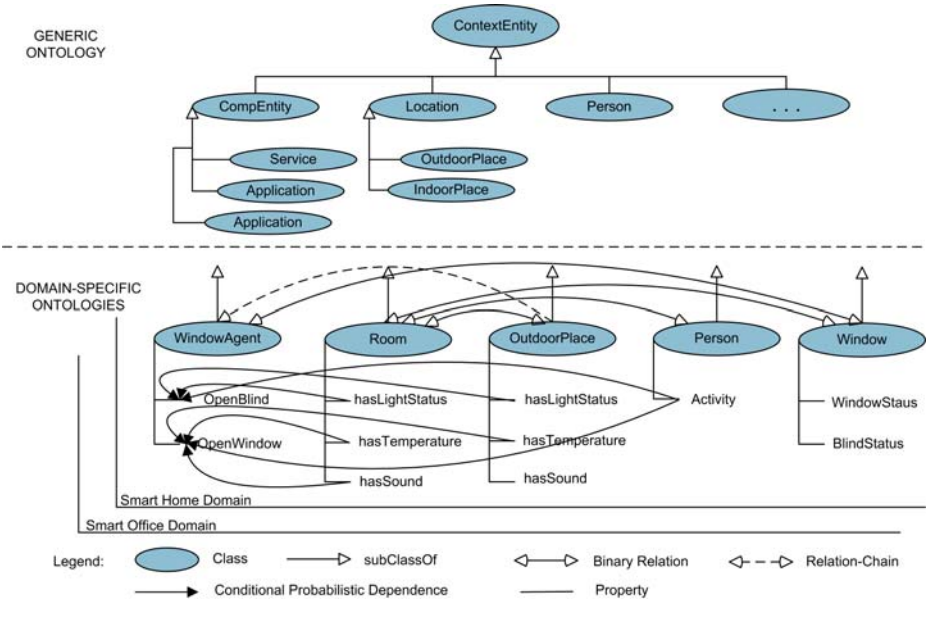

Fig. 2. A two-layer context ontology for relational and probabilistic knowledge

Our unified context ontology is divided into two layers including a generic ontology layer and a domain-specific ontologies layer as follows:

- *The generic ontology* is a high level ontology which captures general context knowledge about physical world in pervasive computing environment.
- *The domain-specific ontologies* are a collection of low-level ontologies which defines the details of concepts and properties in each sub-domain. A low-level ontology of a sub-domain consists of two parts: (1) relational schema which specifies relations and relation-chains of the sub-domain; and (2) probabilistic models which represent conditional probabilistic dependencies between properties in that sub-domain.

In Figure 2, the generic ontology defines basic concepts of CompEntity, Location, Person, Activity, etc. The details of each generic concept, such as relations, relation-chains, conditional probabilistic dependences, are redefined in domain-specific ontologies which may vary from one domain to another.

```
<owl:Restriction>
  <owl:onProperty>
   <owl:ObjectProperty rdf:resource="#OpenBlind"/>
  </owl:onProperty>
  <rdf:hasParents>
   <rdf:List>
    <rdf:first rdf:resources="#PC-ofWindow.ofRoom.hasLightStatus">/>
    <rdf:rest> <rdf:List>
      <rdf:first rdf:resources="#PC-ofWindow.ofRoom.-
        hasOutdoorPlace.hasLightStatus"/>
      <rdf:rest rdf:resource="&rdf;nil" />
    </rdf:List> </rdf:rest>
   </rdf:List>
  </rdf:hasParents>
  <rdf:hasCPT>
   <rdf:List>
    <rdf:first rdf:datatype="&xsd;integer">0.3</rdf:first><rdf:rest>
     <rdf:List>
      <rdf:first rdf:datatype="&xsd;integer">0.9</rdf:first><rdf:rest>
       <rdf:List>
        <rdf:first rdf:datatype="&xsd;integer">0.6</rdf:first>
         <rdf:rest>
          <rdf:List>
           <rdf:first rdf:datatype="&xsd;integer">0.4</rdf:first>
           <rdf:rest rdf:resource="&rdf;nil" />
          </rdf:List>
         </rdf:rest>
       </rdf:List> </rdf:rest>
     </rdf:List> </rdf:rest>
   </rdf:List>
```

Fig. 3. Example of the OWL-based probabilistic dependency relation

We also use the Web Ontology Language (OWL) [13] for representing context. However, we augmented new language elements to model new concepts such as relation-chain, property-chain and probabilistic dependency. We called this language

PROWL (Probabilistic annotated OWL). Figure 3 is an example of the PROWL-based ontology in which we use new markup elements.

6 Reasoning About Context

Based on the context ontology supports representation of both ontological and probabilistic knowledge, we could construct a knowledge base for a new application-domain. We support three reasoning types: rule-based reasoning, ontological reasoning and Bayesian reasoning.

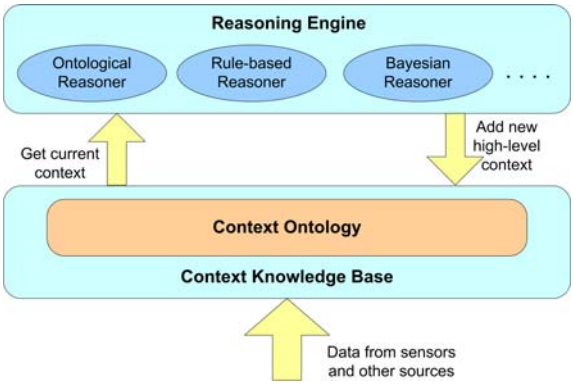

Fig. 4. Three supported reasoning mechanisms

6.1 Rule-Based and Ontological Reasoner

The rule-based reasoner is support by default given context ontology. However, there are differences when applying rules to a property of a p-class. The rule should update both the value of the property and the probability distribution over all values to satisfy the probability axioms.

For example, in our scenario, the context `WindowAgent.OpenBlind` can be deduced from the sensed, primary context `{OutdoorPlace.hasLightStatus, Room.hasLightStatus}` as follows:

```
Prob(hasLightStatus(OutdoorPlace, Bright), 1.0) ^
Prob(hasLightStatus(Room, Dim), 1.0) ^
Prob(Activity(Binh, Sleeping) , 1.0)
⇒ Prob(OpenBlind(WindowAgent, Open), 1.0)
```

The ontological reasoner can be described as an instance of the rule-based reasoner. However, it has a rule-base which consists of predefined rules to implement the language PROWL. In particular, the ontological reasoner can reason about OWL vocabularies and new concepts like relation-chain, property-chain. For example, if the class `WindowAgent` has a relation chain `ofWindow.ofRoom` which has type of the

class `Room`, the relation chain `ofWindow.ofRoom` of the instance `A` of class `WindowAgent` will refer to an instance `B` of class `Room` so that the relation-chain `ofWindow.ofRoom` satisfies.

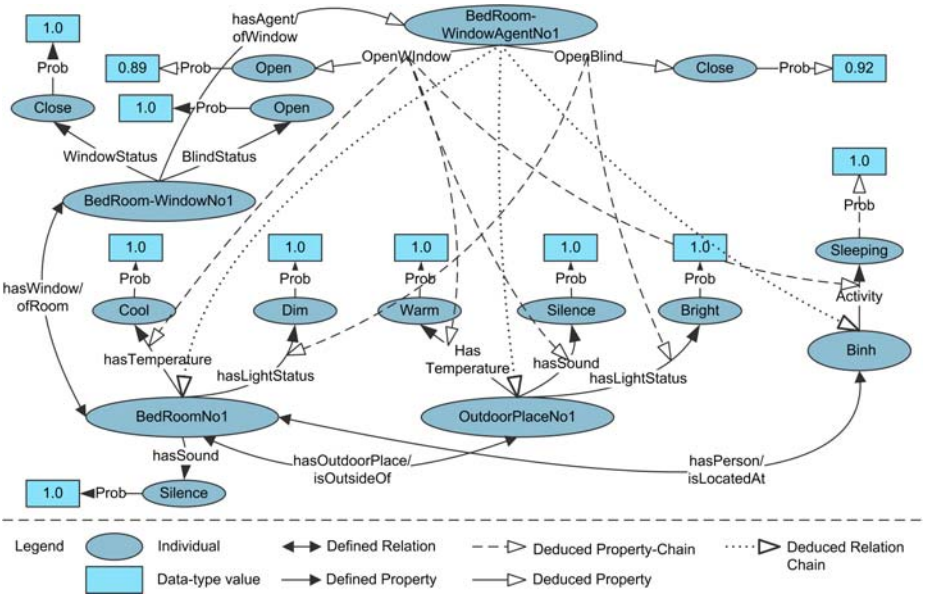

Fig. 5. An example of the context ontology for the scenario

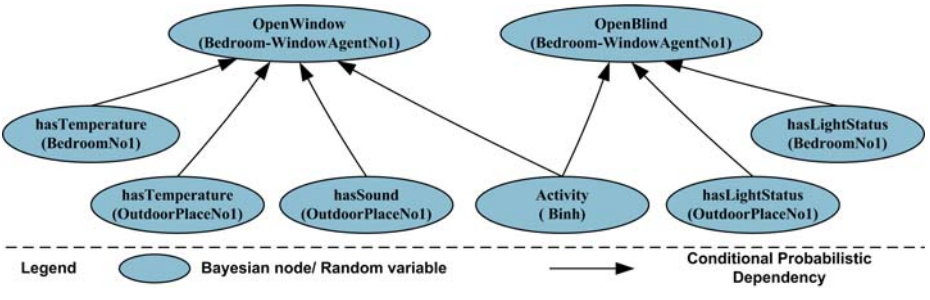

Fig. 6. A derived Bayesian network given the unified context ontology

6.2 Bayesian Reasoning

Before the standard Bayesian network inference can be used to answer queries about values of properties of instances, a Bayesian network is constructed from the context ontology. Depending on the domain, there may be more than one derived Bayesian network corresponding to each probabilistic relational model in the context ontology.

The algorithm **Construct-BN** [8] for deriving a Bayesian network is described as follows. Each node in the Bayesian net B has the form I.P where I is an instance of a p-class and P is a property. The algorithm maintains a list L of nodes to be processed. Initially, L contains only the simple properties of named instances. In each iteration, the algorithm removes a node from L and processes it. The removed node I.P is processed as follows. For each parent $I.RC.P_i$, which refers to a property of another instance, an edge is added form $I.RC.P_i$ to I.P; If $I.RC.P_i$ is not already in B, we add $I.RC.P_i$ into B and L; when all parents of I.P have been added, the CPT is constructed from the has-CPT restriction of I.P.

Since the Bayesian network is available, the standard Bayesian reasoner can use that network to infer about the probabilities of all nodes. Then, the probability of each node is updated directly to the property of instances the ontology.

We implemented the Bayesian reasoner based on the API of Microsoft Belief Network software [12]. The ontological and rule-based reasoners are developed based on the JenaAPI [14]. The mapping module for deriving Bayesian networks from the context ontology (implementing the Construct-BN algorithm) and updating new probability values to the ontology is also implemented based on Jena.

7 Discussions and Conclusions

The major characteristic in our approach is that we define the probabilistic information at the level of concepts. We not only specify the uncertainty of concept's value (property's value) but also specify the probabilistic or uncertain relationships between concepts. Since ontology mainly deals with concepts within a domain, our context model can easily extend the current ontology-based modeling approach. Based on our unified context model, we can easily define a unified, domain-oriented context ontology which captures both logical or relational and probabilistic knowledge. Given that unified context ontology, we can build several knowledge bases for similar applications. For example, we can model a smart-home domain and build the smart-home ontology. For every new smart-home applications, we only need to specify the instances given that predefined smart-home ontology without redefine or construct a new one. Besides, we can add probabilistic information into an existing ontology by adding relations, relation chains and restrictions without construct a new one from the scratch. Thus, our work in context modeling supports scalability and knowledge reusability. Since the mapping-relations between nodes in Bayesian networks and properties of classes are implicitly defined in the ontology, the mapping process can be programmed to run automatically. This feature reduces much burden on knowledge experts and developers in comparison with previous works [2], [11]. Finally, since probabilistic reasoning is supported, we can easily extend from reasoning to learning about uncertain context, which is simply learning about the parameters of Bayesian networks. The learning makes the Bayesian reasoning more robust and adaptive in highly dynamic and variable environments.

This paper describes our approach of representing and reasoning about uncertain context. Our study in this paper shows that the proposed context model is feasible and necessary for supporting context modeling and reasoning in pervasive computing. Our work is part of an ongoing research on Context Aware Middleware for Ubiquitous

System (CAMUS), which attempts to provide an easy, reusable infrastructure to develop ubiquitous context-aware applications. We are exploring methods to integrate multiple reasoning methods from AI area and their supported representation mechanism into the context reasoning and management layer.

References

[1] Satyanarayanan, M, "Coping with uncertainty", IEEE Pervasive Computing, page 2, Volume 2, Issue 3, July-Sept. 2003

[2] Anand Ranganathan, Jalal Al-Muhtadi, Roy H. Campbell, "Reasoning about Uncertain Contexts in Pervasive Computing Environments", IEEE Pervasive Computing, pp 62-70 (Vol.3, No 2) , Apr-June 2004.

[3] Abdelsalam, W.; Ebrahim, Y., " Managing uncertainty: modeling users in location-tracking applications", IEEE Pervasive Computing, pages 60-65, Volume 3, Issue 3, July-Sept. 2004

[4] J. Pearl, "Belief Networks Revisited". In Artificial intelligence in perspective, pages 49-56, 1994

[5] Henricksen, Karen and Indulska, Jadwiga and Rakotonirainy, Andry,. "Modeling Context Information in Pervasive Computing Systems". First International Conference on Pervasive Computing, Pervasive'2002, LNCS(2414), pages 167-180, Zurich, August 2002.

[6] Nir Friedman , Lise Getoor , Daphne Koller and Avi Pfeffer, "Learning Probabilistic Relational Models", Proceedings of the 16th International Joint Conference on Artificial Intelligence (pp. 1300-1307), Stockholm, Sweden, August 1999.

[7] Daphne Koller andAvi Pfeffer, "Probabilistic frame-based systems", Proceeding of the 15th National Conference on Artificial Intelligence (pp. 580-587), Madison, Wilconsin, July 1998.

[8] Gregory D. Abowd and Anind K. Dey, "Towards a Better Understanding of Context and Context-Awareness", Workshop on the what, who, where, when and how of context-awareness at CHI 2000, April 2000.

[9] Hui Lei, Daby M. Sow, John S. Davis, II, Guruduth Banavar and Maria R. Ebling, "The design and applications of a context service", ACM SIGMOBILE Mobile Computing and Communications Review, vol 6, no. 4, pp 44-55, 2002.

[10] Gray, P., Salber, D. "Modeling and using sensed context in the design of interactive applications", In Proceedings of 8th IFIP Conference on Engineering for Human-Computer Interaction, Toronto, 2001.

[11] Tao Gu1, Hung Keng Pung and Da Qing Zhang, "A Bayesian approach for dealing with uncertain contexts", Proceedings of the Second International Conference on Pervasive Computing (Pervasive 2004), Vienna, Austria, April 2004.

[12] Microsoft Belief Network software, http://research.microsoft.com/adapt/MSBNx/

[13] W3C, "Web Ontology Language (OWL)", http://www.w3.org/2004/OWL/

[14] Jena, "A Semantic Web Framework for Java", http://jena.sourceforge.net/