

Modeling Uncertainty in Context-Aware Computing

Binh An Truong, Young-Koo Lee and Sung-Young Lee
Computer Engineering Department, KyungHee University
Giheung-Eup, Yongin-Si, Gyeonggi-Do, 449-701, Korea
tabinh@oslab.khu.ac.kr, {yklee, sylee}@khu.ac.kr

Abstract

Uncertainty always exists as an unavoidable factor in any pervasive context-aware applications. This is mostly caused by the imperfectness and incompleteness of data. In this paper, we propose a novel approach to model the uncertain context. Our context model is a combination of two modeling methods: probabilistic models for capturing the uncertain information and ontology for facilitating knowledge reuse and sharing. Such combination of probabilistic models and ontology facilitates the sharing and reuse over similar domains of not only the logical knowledge but also the uncertain knowledge. Besides, we also support the uncertain reasoning in context-aware applications in a flexible and adaptive manner.

1. Introduction

Context-awareness takes an important role in pervasive computing [1]. As several computers are embedded in the user's surrounding, context-awareness is the key to enable computational system operating autonomously and adaptively without much concern from users. Context might be considered as any information which is relevant to the interaction between the user and the application. Time and date, location, temperature, light, noise and activity can be considered as contextual information or context.

Uncertainty is an unavoidable factor in any context-aware application [2] [3] [4] [5]. This is mostly caused by the imperfectness and incompleteness of data. It also caused by the limitation of underlying rule-based reasoning mechanism of current context-aware systems. As a result, the high-level context information deduced from raw sensing data may not be correct. For example, it is difficult to deduce accurately the user's activities if we only have the information about the user's location and his surrounding's lighting. Since a context-aware system relies mainly on sensors to

capture the context, it must have the capability of modeling and reasoning about the uncertain context.

Using ontology for modeling the context is currently a de-facto standard [4] [6] [7]. The main motivation is that ontology facilitates the knowledge sharing and reuse between devices and applications. Ontology is defined as "a formal, explicit specification of a shared conceptualization". Given a domain, the ontology specifies entities, relations, functions, axioms and instances. By ontological commitment, devices and services share a common understanding about the domain while communicating with each other. By explicitly ontological definitions, inconsistent knowledge can be detected and resolved and high-level context can be deduced from low-level, raw context. By reusing well-define context ontologies, we can build a large-scale context ontology without starting from the scratch. However, there has not been research on representing uncertainty in ontology to support uncertain reasoning.

In this paper, we propose an integrated context model to deal with uncertainty in context-aware computing environment. Our model is a combination of probabilistic models or Bayesian network and ontology. We associate the context ontology with the domain-specific probabilistic information and conditional probabilistic dependencies. Given such context ontology, particular Bayesian networks are derived for particular application for uncertain reasoning. We believe that such integration will provide more flexibility to the uncertain reasoning in context-aware systems.

The rest of paper is organized as follows. In section 2, we discuss about the related work. Section 3 describes a smart home scenario for illustrating our concepts in this paper. In section 4, we present our context model which is used to model both logical and uncertain knowledge. We will show how to model the context information corresponding to the scenario described in section 3. Section 5 describes a context ontology which is built based on our context model.

Section 6 describes briefly our context-aware middleware architecture CAMUS for constructing context-aware applications. Finally, we end the paper with conclusions and directions for further research in section 7.

2. Related work

There has been research in addressing the issue of uncertainty in pervasive computing environment. First efforts tried to modeling the uncertainty of context information using various terms such as "certainty measure" [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, using Bayesian networks to model and reason about the uncertain contexts was received much attention by context-aware research community. Tao Gu et. al. [7] represented a particular Bayesian network in the OWL-based ontology and then translates into a Bayesian network for reasoning. Ranganathan et al. [4] proposes to use Microsoft's Belief Network (MSBN) software to create the Bayesian networks structure. Before it can be used, the Bayesian networks needs to be trained by developers.

These two approaches solved the major problem of dealing with uncertainty by providing the efficient probabilistic reasoning mechanism. However, none of them consider the issue of reusing the uncertain knowledge captured by Bayesian networks even though ontology was used. Apparently, the knowledge captured by a particular Bayesian network is fixed and distinctive for a particular application so that it is unable to share and reuse between applications. Additionally, in similar domain-specific applications, reusing and sharing of Bayesian networks will make the ontology more compact and unique. For example, we can model a general Bayesian network for a general smart-home domain and then reuse that Bayesian network for all real, specific smart-home applications.

3. A Smart Home Scenario

In this section, we describe a smart automated home that can control the environmental conditions to reduce the energy consumption. The goal is to reduce the resource used by the heating or cooling, the lighting or air rotation inside the house, etc. For instance, on a day when the temperature is shifted from warm to cold, the home might determine that the optimal warming strategy is to open the windows and blinds so that the

warm air can come inside while shutting down the heaters. This scenario seems to be very simple but in practice, it is more complicated in the real world situations. There are cases that the automatic system needs to consider in order not to fail in "do the right thing", for example: the outside is so noisy while there are people reading, someone does not want the blinds being opened 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 factors. Such dependencies are also changed from situation to situation.

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

4. The Integrated Context Model

Our context model is influenced by the Probabilistic Relational Model (PRM) developed by Friedman et al. in [11] and the Probabilistic Frame-based systems of Koller et al. in [12]. However, we have made some modifications in comparison with the original model to make it suitable to our requirements.

Our Integrated Context Model consists of two parts:

- *A relational schema* specifies all entities and relationship within a domain. The relational schema represents the structural and relational knowledge of the domain.
- *Probabilistic models* annotate the conditional probabilistic dependencies between entities in that domain. The probabilistic models represent the uncertain knowledge of the domain.

4.1. The Relational Schema

The basic unit of our context model is a class X . Classes are organized in is-a hierarchy, where one class may be a sub class of another (its super class). A class includes a set of relations R_1, \dots, R_n and attributes A_1, \dots, A_n . A relation R defines a binary relation between two classes X and Y . Relations are typed. A relation $R(X, Y)$ can be considered as an attribute R of class X whose value is class Y . We can compose a relation-chain $T=R_1, \dots, R_n$ from several binary relations separated by period. A relation-chain refers to the last class in the chain. Each relation R_i in the chain must be appropriately typed.

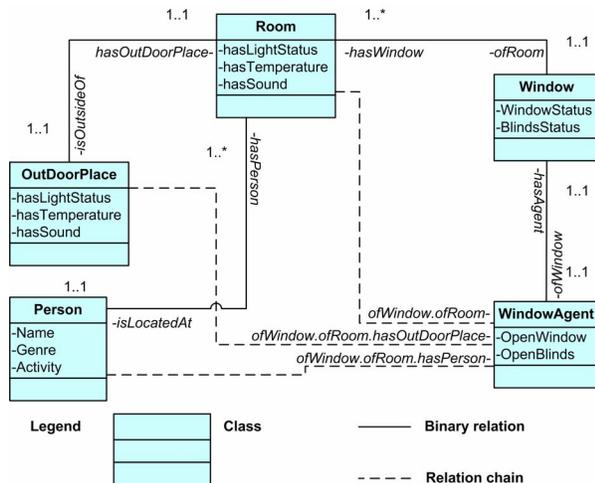


Figure 1: The Relational Schema

A class X has a set of attributes A_i . Each attribute $X.A_i$ takes values in some fixed domain of values $V(A_i, X)$. An attribute-chain is formed by appending a relation-chain with an attribute of the last class in chain. An attribute-chain is a reference from a class to the other class's attribute.

An attribute of a class may have associated constraints. A standard constraint is value-type which specifies a value restriction on the values of an attribute.

Considering the simple smart home scenario in section 3, Figure 1 specifies the entities and their relationships used to model that scenario. The domain contains five classes, including *Window*, *WindowAgent*, *Room*, *OutdoorPlace* and *User*. There are three relation-chain from class *WindowAgent* to three classes *Room*, *OutdoorPlace* and *Person* as shown in the figure.

4.2 The Probabilistic Model

For representing the uncertain information, we annotate a class with local probabilistic model. This type of class is called p-class. A p-class has a set of attributes, each having its associated constraints. A p-class can also have attributes which do not participate in the probabilistic model. For example, the attribute *Person.Name* or *Person.Type* has fixed and independent value and does not include an associated probability model. The constraints of those attributes are simply lists of possible values.

An attribute which participates in the probabilistic model is called a p-attribute. There are two types of p-attribute: simple p-attribute and complex p-attribute.

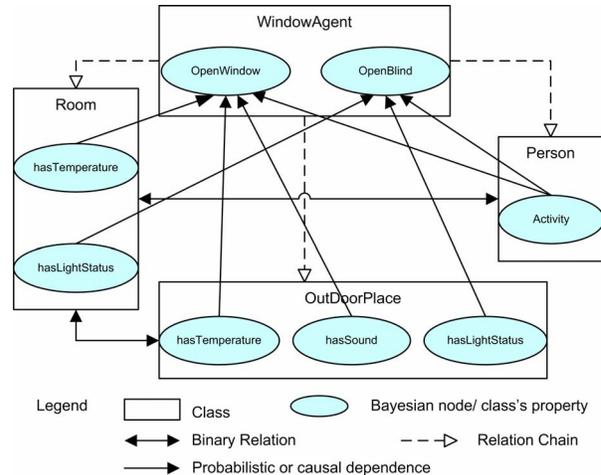


Figure 2: The Probabilistic Model

The simple p-attribute's values are independent from other attributes. It is similar to a root node in Bayesian networks. A normal p-attribute must have two constraints: (1) a *Value-Range* constraint which specifies an enumerated list of possible values and (2) a *Probability-Distribution* constraint which specifies the probabilities associated with values specified in the value-range constraint.

A complex p-attribute, besides having two constraints: *Value-Range* and *Probability-Distribution* as the simple p-attribute, has an associated probabilistic model. This probabilistic model is specified by two special constraints: (1) a *Parents* constraint and a *CPT* constraint. The *Parents* constraint lists all the p-attributes of classes or its parents on which the value of this p-attribute depends. Each parent is specified by an attribute-chain referring to a simple or complex p-attribute. For example, as in figure 2, the parents of the p-attribute *WindowAgent.OpenBlind* are the attribute-chain *ofWindow.ofRoom.hasLightStatus* and the *ofWindow.ofRoom.hasOurDoorPlace.hasLightStatus* attribute-chain. The *CPT* constraint specifies the conditional probability distribution (CPD) over values of the attribute given its parents. The *CPT* constraint is represented by 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. For simplicity, we assume that the *CPTs* are represented as fully specified functions of parent values.

Figure 2 shows the conditional probabilistic dependencies given the relational schema defined in figure 1.

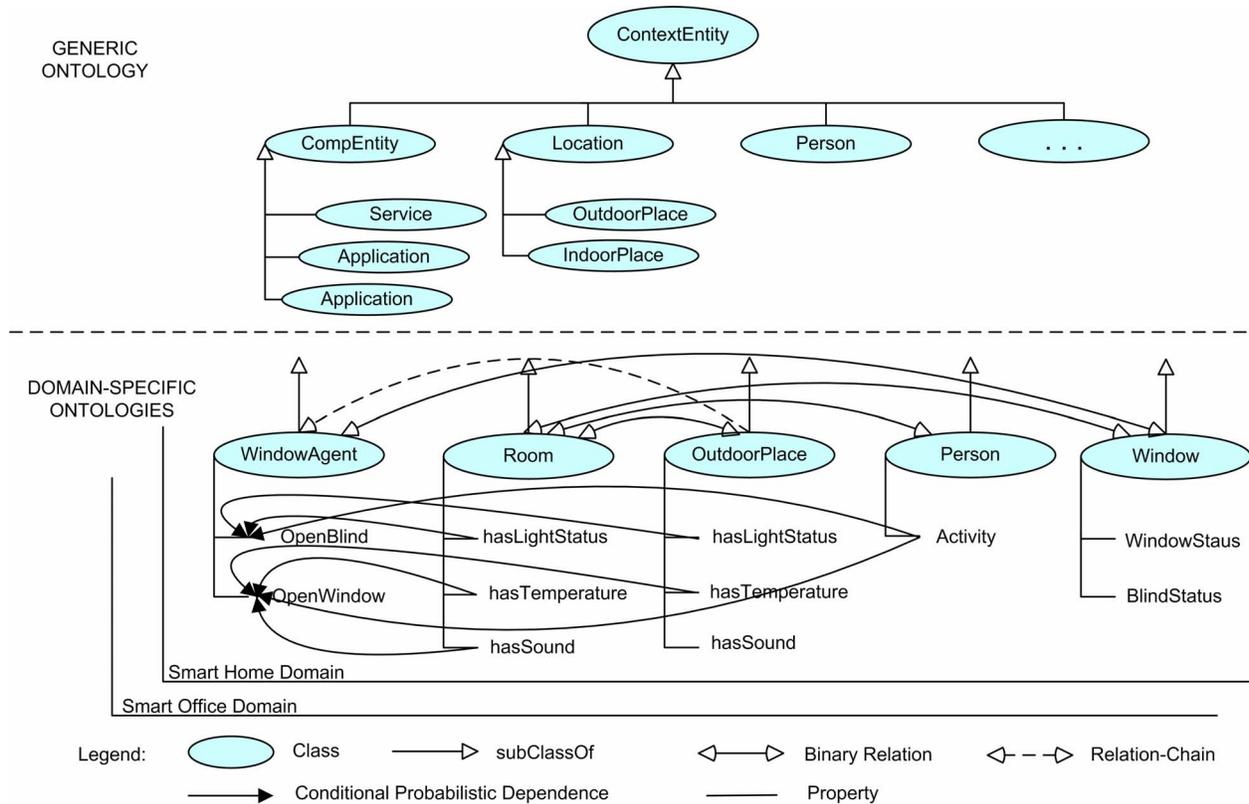


Figure 3: An integrated context ontology

5. The Context Ontology

Based on the integrated context model, we can build context ontology for a specific type of domain. This context ontology not only provides the vocabulary for representing structural and relational knowledge in that domain but also represents the probabilistic knowledge which can be used for probabilistic reasoning.

By using the relational schema, the context ontology defines basic classes and relations representing concepts in a domain. The uncertain context is represented by associating the class's attributes with local probabilistic model. The probabilistic model can be considered as a template of a Bayesian network for a particular type of domain. Given a particular application with particular entities and relations, the probabilistic model is translated into a particular Bayesian network for reasoning. For example, we can define integrated context ontologies that characterize domains like smart-home, smart-office or smart-hospital. Those context ontologies can be reused for every new application without redefinitions even the probabilistic models.

An obvious benefit of representing probabilistic models into ontology is that ontology allows the knowledge captured by probabilistic models explicitly available. This is completely different to previous approaches [4], [7] as they define probabilistic models in an instance-based manner. Based on our integrated context model, the system can derive Bayesian networks corresponding to several but similar applications. Each Bayesian network reflects and adapts to the current changes in its application-domain.

Our context ontology is designed into two layers, as in Figure 3, including:

- **Generic ontology:** specifies all the background knowledge that is valid across several domains. For example, the generic ontology specifies general concepts like Person, Location, Place, Agent ...
- **Domain-oriented ontologies:** capture the specific knowledge characterized by a particular type of domain. A domain-oriented ontology consists of two parts: (1) relational schema which captures all relations, relation-chains of the domain; and (2) probabilistic models which capture all conditional probabilistic dependencies between properties in that domain.

For a representation language of the context ontology, we adopt using the Web Ontology Language (OWL) [13]. OWL enables the definitions of domain ontologies and sharing vocabularies and models of domain in term of classes, properties and relations. Thus, our probabilistic relational model can be easily represented using OWL. Besides, we can utilize the work of extending OWL classes with probabilistic markups in [14].

6. The Context-Aware Middleware

In this section, we present briefly the structure of our Context Aware Middleware for Ubiquitous Services (CAMUS) architecture. CAMUS provides a framework of developing context-aware applications. It consists of the following modules as following:

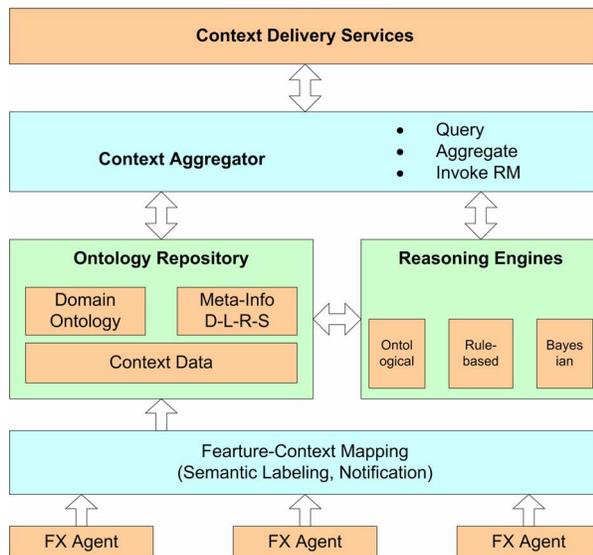


Figure 4: CAMUS architecture

- *Feature Extracting Agents* are responsible of gathering raw context data from sensors. These modules serve as an hardware abstraction layer to the physical sensors
- *Feature Context Mapping* translates the raw context data from FX Agents into semantical context to store it in the ontology
- *Ontology Repository*: provides basic storage services for the context ontology and context information. It uses OWL as a representation language.
- *Reasoning Engines*: provide reasoning services which infer and reason about implicit high-level context given explicit, low-level context. Given the integrated context model, we can support two types of reasoning: (1) ontological, rule-based

reasoning for consensus contexts and (2) probabilistic (or Bayesian network) reasoning for uncertain contexts.

- *Context Aggregator*: provides some basic services like context querying service or a simple service which detects whether user is awoken to perform certain actions
- *Context Delivery Services*: manages the context aggregators and delivers it to appropriate applications.

7. Conclusion

This paper describes our approach of representing uncertain context by incorporating probabilistic models into the context ontology. 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 Context Aware Middleware for Ubiquitous System (CAMUS), which attempts to provide an easy, reusable infrastructure to develop ubiquitous context-aware applications. We are exploring method to integrate multiple reasoning methods from AI area and their supported representation mechanism into the context reasoning and management layer.

ACKNOWLEDGMENT

This research was supported by the MIC (Ministry of Information and Communication), Republic of Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

References

- [1] Abowd G. D., Ebling M., Hunt G., Lei H. and Gellersen H., "Context-Aware Computing", *IEEE Pervasive Computing Journal*, Volume 1, Issue 3, July-Sept. 2002, pp. 22 - 23
- [2] Satyanarayanan, M., "Coping with uncertainty", *IEEE Pervasive Computing Journal*, Volume 2, Issue 3, July-Sept. 2003, pp. 2
- [3] Benford, S.; Anastasi, R.; Flinham, M.; Greenhalgh, C.; Tandavanitj, N.; Adams, M.; Row-Farr, J., "Coping with uncertainty in a location-based game", *IEEE Pervasive Computing Journal*, Volume 2, Issue 3, July-Sept. 2003, pp 34-41
- [4] Ranganathan A., Al-Muhtadi J. and Campbell R.H., "Reasoning about Uncertain Contexts in Pervasive

Computing Environments", *IEEE Pervasive Computing Journal*, Volume 3, Issue 2, April-June 2004, pp. 62 - 70

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

[6] Chen, H., Finin, T., Anupam J., Kagal, L.; Perich, F. and Dipanjan C., "Intelligent Agents Meet the Semantic Web in Smart Spaces ", *IEEE Internet Computing Journal*, Volume 8, Issue 6, Nov.-Dec. 2004, pp. 69 - 79

[7] Gu T., Pung H.K. and Zhang D.Q., "A Bayesian approach for dealing with uncertain contexts", *In the Proceeding of the Second International Conference on Pervasive Computing (Pervasive 2004)*, Vienna, Austria, April 2004.

[8] A. Schmidt, K.A. Aidoo, A. Takaluoma, U. Tuomela, K.V. Laerhoven, W.V. Velde, "Advanced Interaction in Context", *Proc. 1st Int'l Symp. Handheld and Ubiquitous Computing (HUC 99)*, LNCS 1707, Springer-Verlag, 1999, pp. 89–101.

[9] H. Lei, D.M. Sow, J.S. Davis, G. Banavar and M.R. Ebling, "The design and applications of a context service", *ACM SIGMOBILE Mobile Computing and Communications Review*, ACM Press, Volume 6, Issue 4, Oct 2002. , pp 44-55

[10] P. Gray, D. Salber, "Modeling and using sensed context in the design of interactive applications", *8th IFIP Conference on Engineering for Human-Computer Interaction*, LNCS 2254, Springer-Verlag, 2001, pp. 317-336.

[11] N. Friedman , L. Getoor , D. Koller and A. Pfeffer, "Learning Probabilistic Relational Models", *Proceeding of 16th International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, August 1999, pp. 1300-1307.

[12] D. Koller and A. Pfeffer, "Probabilistic frame-based systems", *Proceeding of the 15th National Conference on Artificial Intelligence*, Madison, Wisconsin, July 1998, (pp. 580-587)

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

[14] Z. Ding, and Y. Peng, "A Probabilistic Extension to Ontology Language OWL", *Proceeding of the 37th Annual Hawaii International Conference on System Sciences*, Hawaii, January 2004, pp. 111-120.