CAMPS: A Middleware for Providing Context-Aware Services for Smart Space*

Weijun Qin, Yue Suo, and Yuanchun Shi

Key Laboratory of Pervasive Computing, Ministry of Education,
Department of Computer Science and Technology,
Tsinghua University, Beijing 100084, China
{qinweijun99, suoy}@mails.tsinghua.edu.cn, shiyc@tsinghua.edu.cn

Abstract. Context-awareness enhances intelligent behaviors in pervasive computing environments, although it is still a great challenge to enable context-awareness due to lack of effective infrastructure to support context-aware applications. In this paper, we present an agent-based middleware called CAMPS for providing context-aware services for Smart Space in order to afford effective supports for context acquisition, representation, interpretation, and utilization to applications. In CAMPS, a formal context model, which combines First Order Probabilistic Logic with OWL ontologies, has been investigated to facilitate context modeling and reasoning about imperfect and ambiguous contextual information and to enable context knowledge sharing and reuse. A context inference mechanism based on an extended Bayesian Network approach has been studied to enable automated reactive and deductive reasoning. In addition, we implement a prototype and study on our experience in smart classroom application.

1 Introduction

It's widely acknowledged that Smart Space is a typical open, distributed and heterogeneous pervasive computing system, which aims at creating a ubiquitous, human-centric environment with embedded computers, information appliances, and multimodal sensors that facilitates human to achieve task efficiently by offering abundant information and assistance from computers. A prominent characteristic of Smart Space for supporting human-centric computing is that it senses and reacts to context, information sensed to characterize the situation of the people, activities, physical environment, and computing entities that is considered relevant to the interaction between user and application [1].

Researchers have been investigating on many research issues of context-aware computing, e.g. context modeling, representation, context inference and knowledge sharing, etc. and developing tools and architecture that make efforts to investigate a number of effective and powerful ways to acquire, represent, and

* This work has been funded by Program for New Century Excellent Talents in University of China (NCET-04-0079).

Y.-C. Chung and J.E. Moreira (Eds.): GPC 2006, LNCS 3947, pp. 644–653, 2006.

make use of sensed and inferred data for providing context-aware services to applications. However, developing context-aware middleware to enable computer applications to make use of contextual information and to enhance human's task is still a great challenge. To address these issues above, we have built a middleware called CAMPS which provides context-aware services to allow applications in a smart space environment to be context-sensitive. Some of the key features of CAMPS are:

– An agent-based, loose-coupling middleware that enables the gathering and management of contextual information from various sensors and software entities, and provides appropriate context-aware services for applications.
– A formal context model that describes and represents various kinds of contextual entities(e.g. person, location, activity, environment, platform, etc.) in a smart space environment.
– A context inference mechanism that supports the deduction new and relevant high-level contextual information to the use of the applications from low-level sensed context data.

The rest of the paper will be organized as the following: Section 2 gives an application scenario and presents the features of contextual information in Smart Space. The ontology-based context model is investigated in Section 3. Section 4 describes the system architecture of CAMPS middleware and Section 5 describes context inference mechanism. A case study on making use of context-aware services is on discussion in Section 6. Section 7 gives related works which encourage our approach, and finally Section 8 makes a conclusion.

2 Context-Awareness in Smart Space

Ongoing research on building context-aware application for Smart Space has been more and more significant and compelling. We discuss the *Smart Cameraman* application in our Smart Classroom system [2] in order to illustrate the nature of context-awareness required by applications in Smart Space, and then return to this application scenario throughout the paper in order to illustrate our middleware architecture for supporting context-aware service.

Ross is giving a lesson in Smart Classroom. A remote student Joey is watching the live-video of overview of the classroom on his laptop computer, while another remote student Monica is watching it via her PDA. Ross posts a question on the Blackboard (shared whiteboard). At that moment the live-video on Joey's laptop turns to the close-up of the "Blackboard" while Monica's PDA displays only the question written on Blackboard. After that, Ross asks Chandler, a local student in the classroom, to answer the question. Chandler stands up and gives his answer. At that moment, the live-video on both Joey and Monica's display screen turns to the close-up of Chandler's posture.

From the description of a context-aware application above, it's obvious that there are many different types of contextual information that can be used by applications in a smart space environment, including physical contexts (e.g. time, location), personal contexts (e.g. identity, preference, mood), device contexts (e.g. display size, power), activity contexts (e.g. class, meeting schedule). Besides those, other types of information are still considered as crucial context in a smart space environment which may be invisible to the participants e.g. systematic contexts (e.g. CPU power, network bandwidth), and application contexts (e.g. agents, services), environmental contexts (e.g. light, temperature), etc.

In order to distinguish the specific properties or attributes of different contextual information, we consider that contexts can be classified as three categories: *Sensed Context*, *Profiled Context*, and *Derived Context*. *Sensed Contexts*, which are usually captured from the physical sensors in the real world, e.g. RFID, location tracker, are a type of temporal sensitive, imperfect and ambiguous information. *Profile Contexts*, which are usually predefined from user or environment profile information, e.g. user profile, are more static but incomplete. *Derived Contexts*, which are usually deduced from the other basic information, are imprecise with inaccuracy.

3 The Context Model

It's acknowledged that a well-designed context model plays an important role to access the context in any context-aware system [3]. In our approach, the basic structure of context is represented as first-order probabilistic logic in order to measure the ambiguity of contexts, which combines the expressive power of first-order logic with the uncertainty handling of probabilistic theory [4]. Referred as a sharing understanding of specific domains, ontology is a formal explicit description of concepts and relationships [5]. In our approach, we adopt an ontology approach to model conceptual contexts in a smart space environment for the following reasons: i) ontologies with fully expressive power allow context representation semantically and explicitly; ii) a common ontology enables entities in Smart Space, e.g. agents, devices, to share, reuse and interoperate context knowledge; iii) ontologies provide various complex efficient inference mechanism to deduce high-level context from low-level, raw context data, and to check inconsistent contextual information due to imperfect sensing [6].

3.1 The Basic Structure of Context

In our model, First Order Probabilistic Logic (FOPL) is adopted to represent the basic structure of context which follows the notion of combining first order logic and probabilistic models in machine learning community [4]. Before representing the basic structure of context, we first introduce several definitions of terminology *Field*, *Predicate*, *ContextAtom* and *ContextLiteral*.

- $Field \in F^*$, where a $Field$ is a set of individuals belong to the same class, e.g., $Person = \{Ross, Joey, Chandler\}$, $Room = \{Room526, Room527\}$.
- $Predict \in V^*$, where a $Predict$ indicates the relationship among the entities or the properties of an entity, e.g. $location$, $coLocate$.

- $ContextAtom \in A^*$, where $ContextAtom$ is represented as the form of $predicate(term, term, \ldots)$ in which a $term$ is a constant symbol, a variable symbol, or a function followed by a parenthesized list of $terms$ separated by the commas, and a $predicate$ acts on $terms$. For example, $location(Ross)$ indicates $Ross'$ location.
- $ContextLiteral \in L^*$, where $ContextLiteral$ is represented as the form of $contextAtom = v$ in which $contextAtom$ is the instance of $ContextAtom$ and v indicates the status of $contextAtom$ or the value of the $terms$. For example, $location(Ross) = Room527$ indicates that $Ross$' location is $Room527$.

The structures and properties of this basic model are described in an ontology language in order to define the conceptual contexts in rich semantic level. In our approach, we propose to represent basic context structure in Web Ontology Language (OWL) [7]. Influenced by Ding's approach of representing probabilities in OWL [8], we define two OWL classes: *PriorProb*, *CondProb*. A prior probability $Pr(L_1)$ of a context literal L_1 is defined as the instance of class *PriorProb*, which has two mandatory properties: *hasContextLiteral* and *hasProbValue*. A conditional probability $Pr(L_1|L)$ of a context literal L_1 is defined as the instance of class *CondProb*, which has three mandatory properties: *hasContextLiteral*, *hasProbValue* and *hasCondition*.

3.2 The Context Ontology

In our model, we divide context ontology into two sets: core context ontology for general conceptual entities in Smart Space and extended context ontology for domain-specific environment, e.g. classroom domain. The core context ontology attempts to define very general concepts for context in Smart Space that are universal and sharable for building context-aware applications. The extended context ontology attempts to define additional concepts and vocabularies for supporting various types of domain-specific applications.

The core context ontology investigate seven basic concepts of user, location, time, activity, service, environment, and platform, which are considered as the basic and general entities existed in Smart Space as shown in Figure 1. Part of the core context ontology is adopted from several different widely-accepted consensus ontologies, e.g. DAML-Time [9], OWL-S [10], etc. The instance of Smart Space consists of classes of *User*, *Location*, *Time*, *Activity*, *Service*, *Environment* and *Platform*.

- *User:* As user plays an important and centric role in the smart space applications, this ontology defines the vocabularies to represent profile information, contact information, user preference and mood which are sensitive to user's current activity or task.
- *Location*, *Time* and *Activity:* Note that the relevancy among location, time, and user's activity facilitates the validation of inconsistent contextual information because these contexts might be sensed by various sensors with different accuracies.

- *Platform* and *Service:* The platform ontology defines descriptions and vocabularies of hardware devices or sensors, and software infrastructure in a smart space. The service ontology defines the multi-level service specifications that platform provides in order to support service discovery and composition.
- *Environment:* The environment ontology defines the context specification of physical environment conditions that user interacts with, such noise level, light condition, humidity and temperature, etc.

The extended context ontology extends the core context ontology, and defines the details and additional vocabularies of which apply to various different domains. The advantage of extended context ontology is that the separation of domain reduces the scale of context knowledge and burdens of context processing for pervasive computing applications, and facilitates the effective context inference with the limited complexity [5].

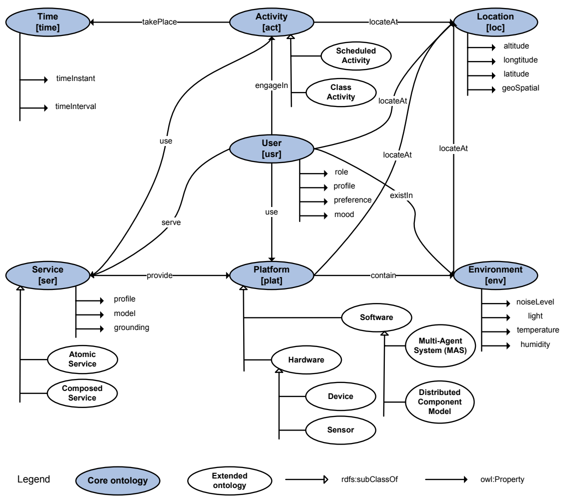

Fig. 1. Context ontology of Smart Space

4 The CAMPS Architecture

The CAMPS is an agent-based context-aware middleware that provides supports for applications to make use of contextual information in a smart space environment. The CAMPS middleware consist of several individual, collaborating agents as depicted in Figure 2.

- *Context Wrapper(CW) Agent.* The *CW Agent* acquires various types of raw context data from different sensors, devices, profiles and software agents.
- *Context Provider(CP) Agent.* The *CP Agent* abstracts context data from heterogeneous source via different types of *CW Agent*, and represents contextual information using ontologies for knowledge sharing and reuse.

– *Inference Engine(IE) Agent*. The *IE Agent* provides inference mechanism including reactive method, first order probabilistic logic and bayesian networks, to infer high-level context from low-level data.
– *Knowledge Base(KB) Agent*. The *KB Agent* stores inference rules, observed facts, and ontologies for context data management and maintenance.
– *Query Filter(QF) Agent*. The *QF Agent* provides query interface to upper applications to query or subscribe the context-aware services with support of system-level coordination mechanism using formal query language.

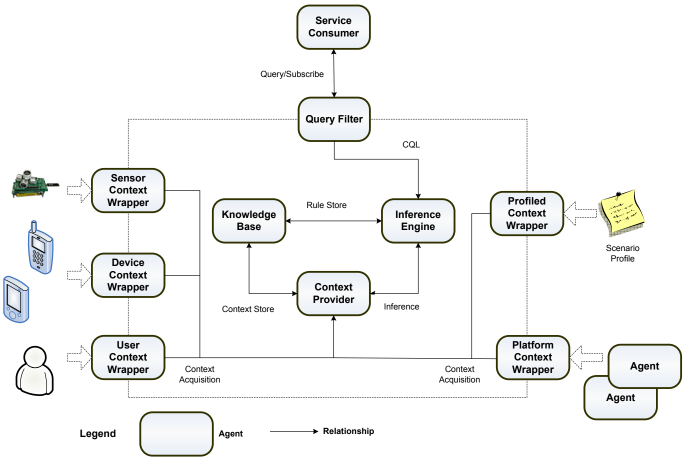

Fig. 2. The CAMPS middleware architecture

The most important considerations of our design architecture for providing context-aware services are mainly relied on some considerations below:

– *Loose coupling*. Contextual Information in a smart space environment is very dynamic and heterogeneous. From the benefits of loose coupling feature, the system can adopt suitable plug-in module to meet different demands of context-aware applications for modeling and reasoning with different types of context knowledge with the least cost of system integration.
– *Scalability*. The middleware architecture with component abstraction and encapsulation provides an easy way to enable context-aware services scalable. By customizing the scenario profile and deploying various types of sensors, *CW Agent* can capture abundant contextual information from different sources to be more adaptive to the real smart space environment.
– *Invisibility*. With notion of the separation of application procedure and underlying services, the middleware provides *QF Agent* module for enabling underlying system functionalities(e.g. context data storage, sensor distribution, inference engine) invisible to the upper applications.

5 Context Inference Mechanism

The inference mechanism of supporting context-aware reasoning is *ContextLogic* that follows the idea of Knowledge-Based Model Construction (KBMC), which is considered as a formal inferential system based on first order probabilistic logic. It consists of formal representation of context knowledge and rules, and the upper inosculation and inference to the knowledge. In this section, we introduce the design approach of *KB Agent* and *IE Agent* which extends Bayesian Network (BN) reasoning arithmetic with restriction of syntax and semantic hypotheses in order to enable the complexity of context reasoning within the acceptable restricted range.

The *KB Agent* takes charges of context persistence, maintenance and management. The construction of *KB Agent* consists of *Field Definition*, *Predicate Definition*, *Observed Facts* and *Rule Definition*. Context rules are the form of $Pr(L_h|L_{b_1}, L_{b_2}, \ldots) = c : -L_{C_1}, L_{C_2}, L_{C_3}, \ldots$, which means that in the constraints of L_{C_1}, L_{C_2}, L_{C_3}, etc. and under the condition of L_{b_1}, L_{b_2}, the probability of L_h is the value of c. Note that L_{C_i} denotes only context fact and others denote arbitrary *ContextLiteral*, e.g. the statement $Pr(TeacherStatus(Teacher) = talking|Speaking(Student) = false) = 0.7 : -IsBlackboardTouched(Room527)$ denotes the rules that when the blackboard of $Room527$ has not been touched, the probability value of that the teacher is talking equals 0.7 under the condition of that the student is silent. We propose XML-based database to store and manage all the definition and information of *Knowledge Base*.

The *IE Agent* takes charges of inferring high semantic context knowledge from low-level context facts with restricted constrains and deductive rules. For example, from the rule of $Pr(Speaking(Joey) = true|Speaking(Ross) = true) = 0 : -status(Roos527) = onMeeting$, we can deduce that while $Ross$ is speaking on the meeting in the $Room527$, it's impossible that $Joey$ is also speaking. In CAMPS, We develop a inference module called *ContextLogic* for converting the context description in *Knowledge Base* into BN's DAG and to calculating the probabilistic distribution. In order to reduce the scale of BN's DAG, we investigate an approach to build DAG according to the context query's content. To achieve the above goal, several constrains are involved, including valid rule of the syntax, the independence hypothesis of causal set that extends definition of causal independence in [11], the hypothesis of average distribution of residual probability, and the conditional independence hypothesis, in order to avoid generating unnecessary node of the net so as to minimize the scale of BN's DAG and ascertain exclusively the distribution of the answer. Therefore, with the help of the constraints and hypothesis, the complexity of inference on BN's DAG is under control within acceptable limited range.

The *QF Agent* classifies the input query into two categories: i) query for context facts. In this case, no complex inference mechanism is involved, but a reactive mapping approach is adopted to provide the answer to the query; ii) query for context knowledge. In this case, the above inference approach is adopted to infer the answer. We defined the basic formal of context query language (CQL) as $?L_{C_1}, L_{C_2}, \ldots$, where L_{C_i} can have either *Field Variable (FV)* or *Field Element (FE)*. The advantage of defining a well-formalization CQL is that we can use it to query in *Context Base* in a uniform way, and the distribution and storage structure of data and the complicated inferential process are transparent to the upper applications and users.

6 Case Study

As depicted in the application scenario, a *Smart Cameraman* module is designed to change the live-video scene adapted to situational context initiatively according to the clue of class activity in local classroom by switching an array of cameras. Distinct with previous version applied in Smart Classroom project [2], we adopt Smart Platform [12], a software infrastructure of Smart Space with multi-agent architecture, as a supporting platform to the middleware CAMPS for providing reliable data communication and module coordination mechanism.

In this case, context-awareness that CAMPS provides embody two aspects: i) CAMPS capture the contextual information relevant to user's activity and provide the clue of class activity to *Smart Cameraman* module. ii) CAMPS deliver various customized video respectively to remote student individuals with different quality due to different capabilities of computers or devices, e.g. size of display screen, network bandwidth.

To demonstrate Smart Cameraman scenario, we define several context rules for this module and develop case generator to simulate a variety of situations as Figure 3(d) shows:

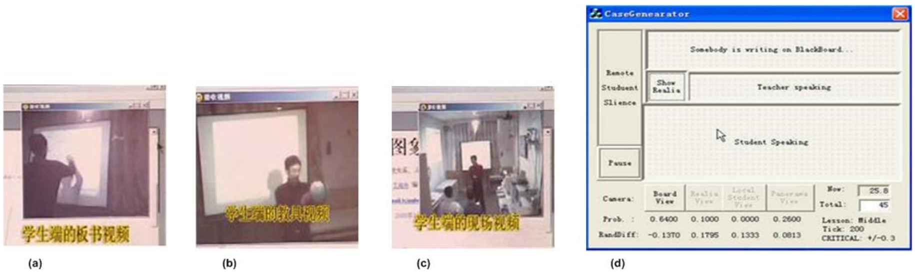

Fig. 3. Different scenes delivered to remote students according to the class context: (a)teacher writing on MediaBoard; (b)teacher showing a model; (c)teacher having a discussion with local students; (d)Case generator module

- *Teacher writing on the MediaBoard*. When the teacher is writing comments on the MediaBoard, the module may select a close-up view of the board, as Figure 3(a) shows.
- *Teacher showing a model*. When the teacher holds up a model, the model may zoom in on it, as Figure 3(b) shows.

– *Remote student speaking*. When a remote student is speaking, live video of the student may be delivered to other remote students.
– *Other*. In all the other situations, the module may select the overview of the classroom, as Figure 3(c) shows.

Compared with the previous version of *Smart Cameraman* in Smart Classroom project, several great systematic improvements have been done with the supports of context-aware middleware CAMPS as depicted in Table 1. The enhanced *Smart Cameraman* module has better scalability and adaptability performance, more expressive power of context representation, inference and discovery, and much easier to maintain and upgrade for the independence relationship among the components.

Table 1. Comparison on systematic improvements of *Smart Cameraman* module

***Smart Cameraman* Module**	**Module Deployment**	**Context Model**	**Inference Mechanism**	**Query Interface**
Previous Version	Toolkit	XML-based Event Description	IF-THEN-ELSE Statement	Ad hoc Manner
Enhanced Version	Agent-based Component	Ontology	FOPL Bayesian Network	Formal Context Query Language

7 Related Works

Over the past few years, a number of works have been done in the area of context-aware computing. Significant work has been done by Dey, et al. in defining the concepts of context and context-awareness, identifying categories of context and features of context-aware applications and developing a conceptual framework for supporting rapid prototyping of context-aware applications [1].

Chen et al. introduce the Semantic Web technologies and ontologies in building an architecture for supporting context-aware systems, investigate the Standard Ontology for Ubiquitous and Pervasive Applications (SOUPA) that uses OWL to represent the entities in a smart space environment, and develop the Context Broker Architecture (CoBrA) that is an agent-based context-aware framework to support ubiquitous agents, services and devices [13].

Gu et al. investigate a Bayesian approach for dealing with uncertain contexts that proposes a probabilistic extension to an ontology-based model for representing uncertain contexts, and use Bayesian Network to reason about uncertainty. A serviced-oriented context-aware middleware has been investigated in order to enable building and rapid prototyping of context-aware services [6].

8 Conclusion

In a conclusion, we have presented a middleware for providing context-aware services for Smart Space. The middleware supports the high-level abstraction of context data with the power of formal context model which combine with first-order probabilistic logic and ontologies, and allows context inference based on extended Bayesian Network to provide more precise context information adapted to changing, heterogeneous smart space environment. Our ongoing research is investigating description logic approaches with more expressive power to make middleware robust and extensible.

References

1. Dey, A.K., Salber, D., Abowd, G.D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human Computer Interaction (HCI) Journal - Special Issue on Context-aware Computing **4**(2-4) (2001) 97–166
2. Shi, Y.C., Xie, W.K., Xu, G.Y., Shi, Y.T., Chen, E.Y., Mao, Y.H., Liu, F.: The smart classroom: Merging technologies for seamless tele-education. IEEE Pervasive Computing Magazine **2**(2) (2003) 47–55
3. Strang, T., Linnhoff-Popien, C.: A context modeling survey. In: Proc. UbiComp workshop on Advanced Context Modeling, Reasoning and Management. (2004)
4. Poole, D.: First-order probabilistic inference. In: Proc. IJCAI. (2003) 985–991
5. Guarino, N.: Formal ontology and information systems. In: Proc. FOIS. (1998) 3–15
6. Gu, T., Pung, H.K., Zhang, D.Q.: A service-oriented middleware for building context-aware services. Journal of Network and Computer Applications **28**(1) (2005) 1–18
7. OWL Web Ontology Language Guide, http://www.w3.org/TR/owl-guide/.
8. Ding, Z.L., Peng, Y., Pan, R.: Bayesowl: Uncertainty modeling in semantic web ontologies. Soft Computing in Ontologies and Semantic Web (2005) 27
9. DAML-Time, http://www.cs.rochester.edu/ ferguson/daml/
10. OWL-S: Semantic Markup for Web Services, http://www.daml.org/services/owl-s/1.1/
11. Rish, I., Dechter, R.: On the impact of causal independence. In: Stanford Spring Symposium on Interactive and Mixed-Initiative Decision Theoretic Systems. (1998) 101–108
12. Xie, W.K., Shi, Y.C., Xu, G.Y., Mao, Y.H.: Smart platform - a software infrastructure for smart space (siss). In: Proc. ICMI. (2002) 429–436
13. Chen, H., Finin, T., Joshi, A.: An ontology for context-aware pervasive computing environments. Knowledge Engineering Review - Special Issue on Ontologies for Distributed Systems **18**(3) (2004) 197–207