

An Ontology-based Context Model for Building Context-aware Services

Wang Wusheng, Li Weiping, Wu Zhonghai, Chu Weijie, Mo Tong

Peking University

Beijing, China

{wws, wuzh}@pku.edu.cn, {wpli, chuwj, motong}@ss.pku.edu.cn

Abstract—One of the most important points in context-aware computing is to build a reasonable context model. Currently, many models were not built in a common structure. It is hard to build other models depending on the existing ones. In this paper, we build a common ontology-based context model with Web Ontology Language (OWL), which provides an efficient and convenient way of ontology-building by adding domain-specific concepts into this common model. We show the structure of the model and explain the main concepts of it. With the common model, we propose our solutions to solve the problems of position and context history that may exist in the process of modeling a specific domain. Finally, we show a context model of Beijing Capital International Airport (BCIA), which was built on the common model. A simple reasoning test is run on the BCIA model with Semantic Web Rule Language (SWRL).

Keywords—context-aware computing; common context model; OWL

I. INTRODUCTION

With the rapid development of information technology, electronic products become a vital part of people's ordinary life. However, we are not yet satisfied with current status, thus we are trying to improve the intelligence of computer systems so that they could provide more personalized and comprehensive services. Context-aware systems are designed to solve the problem by exploiting the changing environment with a new class of applications, which are aware of the context of the program they are running on [1].

However, there is no consensus about what context is, and many people have their own definitions of context. Schilit and Theimer define context as location of use, the collection of nearby people and objects, as well as the changes of those objects over time [2]. Jason Pascoe claims that context mainly refers to the environment and the program or the device itself [3]. Context is also defined as the states of user and IT device, including surroundings, situation, and to a lesser extent, location [4]. We do not intend to argue what context is in this paper, whereas we focus on how to classify the different parts in domain-specific context. We divide context into *PreDefinedContext*, *SensedContext* and *swrla:Entity*, and each kind of context can be classified in detail.

In some systems, context is modeled with key-value pairs or object-oriented patterns, but context in these forms cannot be shared efficiently among different devices or services. In this paper, we model context in the form of Web Ontology

Language (OWL) [5]. Endorsed by the World Wide Web Consortium (W3C), OWL is a family of knowledge representation languages for authoring ontologies. The languages are characterized by formal semantics and RDF/XML-based serializations for the Semantic Web [6]. Context written in the form of OWL can easily be shared. Whether they are in the same domain or not, other applications can understand the context. Another important reason for us to choose OWL is that it makes reasoning possible. A number of rule languages are supported for reasoning the contextual information in context models, and reasoning is another important part in context-aware computing.

The rest of the paper is organized as follows: section II introduces related work; the structure of the common ontology-based context model is proposed in section III; in anticipation of possible problems during the building of domain-specific ontology, we show our solutions with the common model in section IV; an example of domain-specific ontology built with the common model is showed in section V, and a conclusion in section VI.

II. RELATED WORK

The interest of research focusing on context modeling and reasoning is growing. Different modeling methods are used including key-value pairs, programing objects and markup language such as XML. Bill Schilit, et al. model context in key-value pairs, and according to them, information which enters into the computer is tagged with context keys, which facilitates its future retrieval [1]. Pervasive Profile Description Language (PPDL), a XML-based language, is used to model context [7]. Joseph Bauer tries to describe context with Unified Modeling Language (UML) [8]. The Active Object Model is used in the GUIDE project [9]. However, all the above models do not support context reuse well.

Since gathering, evaluating and maintaining context information is expensive, reusing and sharing of context information between context-aware applications should be considered from the beginning [10]. Ontology-based context model is designed for knowledge representation and reasoning, and it provides easy knowledge reuse. In this paper, we model context with OWL, which essentially describes the concepts and relationships, and also supports reasoning conveniently.

Many ontology models are built with OWL in specific domain, such as tourism [11], medical diagnose [12] and

home environment [13]. The architectures of these models are hard to reuse and it is difficult for readers to understand the meanings of such models because they are not built under a common model. In this paper, we propose a common ontology-based context model, which is helpful in ontology-building.

III. COMMON ONTOLOGY-BASED CONTEXT MODEL

In OWL, the most basic concepts in a domain should correspond to classes [14]. The relationships between classes are represented by ObjectProperty, while DatatypeProperty defines the relationships between classes and data values. The root of all classes is called *owl:Thing*, and in our model, the subclasses of *owl:Thing* are *PreDefinedContext*, *SensedContext* and *swrla:Entity*. The structure of the common model is shown in Fig. 1.

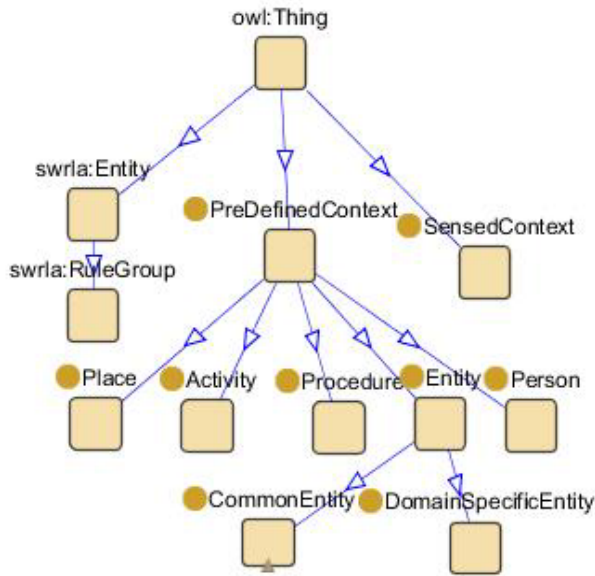


Figure 1. The structure of the common model

A. *PreDefinedContext*

PreDefinedContext refers to the definite context before the running of the application, which is not necessarily to be subjected to modification while the application is running, unless the application needs to be maintained occasionally. *PreDefinedContext* includes *Place*, *Activity*, *Procedure*, *Entity* and *Person*. In a specific model, other classes can be added if necessary.

1) *Place*: One of the most important contexts in context-aware applications is *Place*, which is why context-aware was originally called location-aware. *Place* refers to the locations that the individuals can be in or the areas that the model is built for, such as a room, a building or a playground. The relationships between instances of *Place* are needed within a model. For example, it should be defined that *Room 201* is on the second floor. ObjectProperty is used to describe the relationships. Referring back to the previous example, we can define an

ObjectProperty called *OnTheFloorOf*, whose domain is *Room* and range is *Floor*. With *Place*, we can find out information such as our current location or who are in the same building with us, etc.

2) *Activity*. Context-aware applications automatically detect the action of a person and provide suitable services for him. *Activity* defines the actions that a certain person may take in the model, such as *GoToSchool*, *DoHomework*, and so on.

3) *Procedure*. In order to assist a person effectively, we need to know what a person has done and what to be done next. *Procedure* is represented by a series of *Activities* or *Procedures*, which should be done orderly to finish a job. *Procedure* can be part of another *Procedure*. For example, *MakeAPhoneCall* is a *Procedure*, and it involves *Activities* like *DialTheNumber*, *TalkWithAnotherPerson* and *HungUp*. Furthermore, *MakeAPhoneCall* can also be included in the *Procedure* of *TelephoneInterview*. By defining *Activity* and *Procedure*, we can find out what a person is doing and what will he do next, and we can also reason what can be done in order to assist him.

4) *Person*. Most context-aware services involve persons. The services sense the situations of persons or the environment around persons, and reason what a person is doing and what to be done next. So *Person* is an important context in context-aware systems. Thus we define *Person* as a subclass of *PreDefinedContext*. In domain-specific context models, *Person* can be classified in detail if necessary, such as *Teacher*, *Student*, *Nurse*, *Farmer* and so on.

5) *Entity*. *Entity* refers to the things that are difficult to be defined as *Place*, *Activity*, *Procedure* or *Person*, such as *Ticket*, *Company*, etc. *Entity* contains *CommonEntity* like *Transport*, and *DomainSpecificEntity*.

B. *SensedContext*

Compared with *PreDefinedContext*, *SensedContext* refers to the context that will be acquired during runtime from sensors, such as *TheLocationOfAPerson* and *TheTemperatureOfARoom*. *SensedContext* is in contrast to *PreDefinedContext*, so they disjoint with each other. The reason why we separate *SensedContext* from *PreDefinedContext* is that they need different technologies to deal with. *PreDefinedContext* is inputted into application in advance and the application would only read it, without any revision. *SensedContext* is added into the model dynamically when the application is running. The output of the application depends on what *SensedContext* is obtained. Several items will be marked in an instance of *SensedContext*. For example, when a record of *SensedRoomTemperature* is acquired, the room number, temperature and time should be logged. The relationship between *SensedRoomTemperature* and *Room* is built by ObjectProperty, and *SensedRoomTemperature* is connected with temperature as well as time through DatatypeProperty.

C. Swrla:Entity

Reasoning is another important part in context-aware applications and most reasoning jobs are done with rules. We use Semantic Web Rule Language (SWRL) [15] to add rules into the model. Rules in the form of SWRL can be written in either OWL XML Presentation Syntax or Human Readable Syntax. SWRL can be merged into OWL easily. Swrla:Entity is the set of all the rules in the model.

IV. SOLUTIONS TO KEY PROBLEMS

When domain-specific models are built, several problems may occur. We propose our solutions to some key problems with the common model in this section.

A. Position

For instance, we can define specific rooms in a building by defining an ObjectProperty to link *Room* and *Building*. However, it is difficult to define the distances between all the rooms in the building. Of course ObjectProperty can be used here, but it is hard to decide how many ObjectProperties should be built beforehand. Besides, this geographical information may be shared with other applications, such as position fixing system and guided system. So we should find out a convenient way to present the positions of *Place*.

To solve this problem, a class called *Position* is defined as the subclass of *CommonEntity*. *Position* is divided into several *Zones* [16]. A *Zone* can be any area that is needed in the model, and it can be as small as possible. The properties of *Zone* such as *ApexCoordinate* and relationships between *Zones* such as *ConnectTo* are defined by DatatypeProperty or ObjectProperty. An ObjectProperty called *InThePositionOf* is defined in order to present the correspondences between *Zone* and *Place*.

B. Context History

Histories of context, especially the *SensedContext*, are very important for reasoning. So the system should record *SensedContext* information in time sequence. The management of context histories is difficult if the number of updates is very high [10]. When *SensedContext* comes, it must be written into the model immediately and automatically. Thus we need a program to acquire the contexts and add them into the model. To distinguish the histories of the same context, a sequence number should be given to each record.

V. AN EXAMPLE: BEIJING CAPITAL INTERNATIONAL AIRPORT MODEL

With the common ontology-based context model, a domain-specific context model-- Beijing Capital International Airport (BCIA) model is built. The purpose of this model is to provide passengers with corresponding information and suggestions in their procedures of departure and arrival. The model is so complicated that it is difficult to show the whole structure here, thus only the core parts are presented.

In *PreDefinedContext*, all the facilities of BCIA are inputted into the model, and the activities are added as well.

Entity, *Procedure* and *Person* in BCIA model are shown in Fig.2.

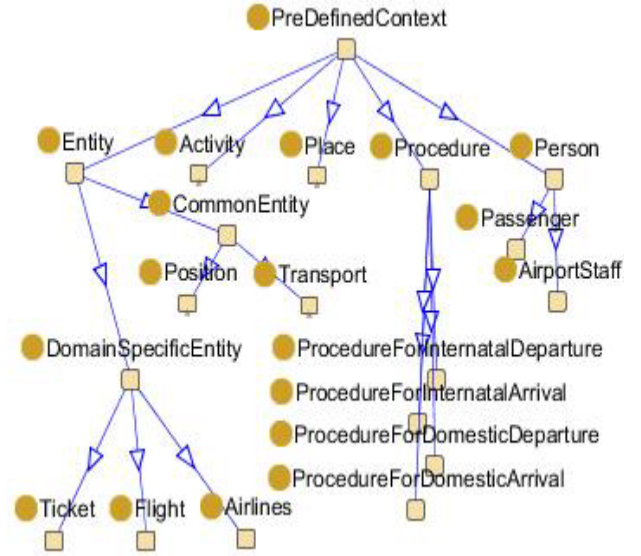


Figure 2. A part of BCIA model

The only *SensedContext* in BCIA model is *SensedPositionOfPassenger*, which is used to record the passengers' positions. *Passenger*, *Place* and time are associated with it by ObjectProperty or DatatypeProperty.

We add a number of rules into this model. Because most of the rules are similar to each other, we present only one SWRL statement in this paper, as below.

$$\begin{aligned} & Passenger(?P) \wedge PassengerHaveTicket(?P, ?T) \wedge \\ & FlightHaveTicket(?F, ?T) \wedge \\ & FlightDepartureTerminalBuilding(?F, \\ & TerminalBuilding_BCIA_T2) \wedge \\ & InternationalOrDomestic(?F, "Domestic") \wedge \\ & TicketDepartureDate(?T, ?TDD) \wedge \\ & swrlb:greaterThan(?TDD, "2010-10-30") \rightarrow \\ & PersonInTheProcedureOf(?P, \\ & ProcedureForDomesticDeparture_BCIA_T2) \end{aligned}$$

The statement shows that if a person *P* buys a ticket *T*, *T* corresponds to flight *F*, *F* is a domestic flight and will leave BCIA from *TerminalBuilding_BCIA_T2*, and *P* will take *F* after 2010-10-30, then *P* is in the procedure of *ProcedureForDomesticDeparture_BCIA_T2*.

Passenger Weiping spends 1210 RMB on Flight MU5102, for November 2, 2010. MU5102 is a domestic flight which needs to be boarded from terminal building 2, and travels from Beijing Capital International Airport to Shanghai Hongqiao International Airport. In BCIA model, the reasoning result shows that Weiping is in the procedure of *ProcedureForDomesticDeparture_BCIA_T2*, which is correct.

VI. CONCLUSION

Context modeling is important in context-aware applications, and a number of models are built in different

methods and different structures. In this paper, we propose a common context model with OWL and present the solutions with the common model to solve the problems of position and context history. A BCIA model built with the common model is given, in which some reasoning rules described by SWRL is defined to capture the changing activity of a certain passenger. The result shows that the common ontology-based model is reasonable and practical.

ACKNOWLEDGMENT

Research in this paper is supported by the Danish Strategic Research Council (Grant NO. 2106-08-0046) and the NSFC (NO. 61033005).

REFERENCES

- [1] Bill N. Schilit, Norman Adams and Roy Want, "Context-Aware Computing Applications," Proc. Mobile Computing Systems and Applications, Dec. 1994, pp. 89–101, Santa Cruz, CA, US, doi: 10.1109/MCSA.1994.512740.
- [2] Bill N. Schilit and Marvin M. Theimer, "Disseminating active map information to mobile hosts," IEEE Network, Vol. 8(5), pp. 22-32, Sep/Oct 1994.
- [3] Jason Padding, "Adding Generic Contextual Capabilities to Wearable Computers," Proc. IEEE International Symposium on Wearable Computers, Oct. 1998, pp. 92 – 99, Pittsburgh, PA, doi: 10.1109/ISWC.1998.729534.
- [4] Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomelainen, Kristof Van Laerhoven and Walter Van de Velde, "Advanced Interaction in Context," Proc. First International Symposium on Handheld and Ubiquitous Computing, Springer Verlag, 1999, pp. 89–101, Karlsruhe, Germany.
- [5] <http://www.w3.org/TR/owl-features/>, Accessed on October 17, 2010.
- [6] http://en.wikipedia.org/wiki/Ontology_Web_Language, Accessed on October 17, 2010.
- [7] Ekaterina Chtcherbina and Marquart Franz, "Peer-to-peer coordination framework (p2pc): Enabler of mobile ad-hoc networking for medicine, business, and entertainment," Proc. International Conference on Advances in Infrastructure for Electronic Business, Education, Science, Medicine, and Mobile Technologies on the Internet, Jan. 2003, L'Aquila, Italy.
- [8] Joseph Bauer, "Identification and Modeling of Contexts for Different Information," Mar 2003, Berlin.
- [9] Keith Cheverst, Keith Mitchell and Nigel Davies, "Design of an object model for a context sensitive tourist GUIDE," Computers & Graphics, Vol. 23(6), Dec 1999, pp. 883-891.
- [10] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan and Daniele Riboni, "A Survey of Context Modeling and Reasoning Techniques," Pervasive and Mobile Computing, Vol. 6(2), Apr. 2010, pp. 161-180.
- [11] Federica Paganelli, Gabriele Bianchi and Dino Giuli, "A Context Model for Context-aware System Design towards the Ambient Intelligence Vision: Experiences In the eTourism Domain," Lecture Notes in Computer Science, Vol. 4397, 2007, pp. 173-191, doi: 10.1007/978-3-540-71025-7_12.
- [12] Marek Jaszuk, Grażyna Szostek and Andrzej Walczak, "An ontology building system for structuring medical diagnostic knowledge," Proc. The 3rd International Conference on Human System Interactions, May 2010, pp. 203 – 210, Rzeszow, doi: 10.1109/HSI.2010.5514567.
- [13] Tam Van Nguyen, Wontaek Lim, Huy Nguyen, Deokjai Choi and Chilwoo Lee, "Context Ontology Implementation for Smart Home," Proc. The 2nd International Conference on Ubiquitous Information Technologies & Applications, 2007, Java Island, Indonesia.
- [14] <http://www.w3.org/TR/owl-guide/>, Accessed on October 17, 2010.
- [15] <http://www.w3.org/Submission/SWRL/>, Accessed on October 17, 2010.
- [16] Weiping Li, Weijie Chu, Frank Tung and Zhonghai Wu, "A Uniform Device Information Access for Context-aware Middleware," Proc. IEEE International Conference on Web Services, July 2010, Miami, FL, doi: 10.1109/ICWS.2010.43