A Novel Context-aware Modeling and Reasoning Method based on OWL

Bo Hu
Institute of Command Automation,PLA University of Science and Technology,NanJing,China
Email: jsnjhb@126.com

Zhi-Xue Wang and Qing-Chao Dong
Institute of Command Automation,PLA University of Science and Technology,NanJing,China
Email: wzxcx@126.com

Abstract**—**Computing is moving toward a pervasive context-aware environment. This paper proposes an approach to model and reason the Context-aware knowledge using Web Ontology Language (OWL) that towards the problem of uncertain model and lack of automatic reasoning support. It suggests a novel context model framework in the semantic restrictions of the ontology, which dividing the context model into two-level structure: Meta Ontology and Domain-specific Ontology, according to the abstract hierarchy. A key component in our framework is the explicit context model defined in OWL, and the objective of our model is to provide flexible extensibility to add specific concepts in different application domains. After that, the paper proposes an algorithm which converting the OWL model to Description Logic to implement automatic context reasoning. Finally, a case study is given to illustrate the practicality of the method.

Index Terms**—**Context model, Context Reasoning, OWL, Ontology

I. Introduction

Pervasive computing is receiving more and more interest among researchers recently which including devices, services, operator and software platform technique behind them. It will seamlessly cooperate in support of human requirements and needs, negotiating for services, reflect our behalf, and providing services in an anytime, anywhere way. To facilitate this way, an important step is the development of the infrastructure that can sustain prevalent sensors, services, and devices become aware of their contexts including the ability to reason and to share the contexts [1, 2]. By the context, we must understand of an environment and the people, devices, objects and software agents it contains. This understanding necessarily extends to modeling the activities and tasks that are taking place in a location as well as the intentions of the human and software agents involved.

A number of researches have been built in the past aim to support context-aware computing. Karen Henricksen[3] proposed a context model language CML, which divided the context into four types(sensed、static、profiled and derived), Quan Z.Sheng[4,5] proposed a UML-based model method called ContextUML focusing on context-aware Web services. But the methods mentioned-above both design the context model in a specific domain, not propose a set of general context modeling technique, lack the automatic reasoning support for the context-aware, so as to violate the basic principle of pervasive computing to reduce the human-computer interaction. Tao Gu[6] propose an ontology-based context modeling method using OWL, which can carry out semantic context reasoning. But restricted by the technical condition at that time, it also lack the support of automatic reasoning tool. Dong Liu[7] propose another ontology-based modeling method called CACO and a reasoning arithmetic based on rules. A significant source of weakness is that they are not built on a foundation of common ontologies with explicit semantic representation [8].

In order to tackle the problem in context-aware modeling and reasoning, we first propose a context model framework based on ontology, dividing the model into Meta Ontology level and Domain-specific Ontology level according to the abstraction hierarchy. The Meta Ontology is the high abstract level which extracting the basic element of the context knowledge. The Domain-specific Ontology is the lower abstract lever which focusing on different domain knowledge, directed by the Meta Ontology, and it is the instantiation of the Meta Ontology. After that, we design an algorithm to convert the context model to the knowledge described in OWL DL, then using the reasoning tool to accomplish the context reasoning. The rest of the paper is organized as follows. Section 2 discusses some related work. Section 3 exposes the context modeling framework based on ontology. Section 4 presents the convert algorithmC

Corresponding author:jsnjhb@126.com.

doi:10.4304/jcp.8.4.943-950

converting the context model to OWL DL, and gives a case study to illustrate our method, and we conclude in Section 5.

II. RELATED WORK

In this section, we review and discuss some important context models. We classify those context models into three categories:

Application-based method: Many existing context-aware systems model only for specific applications. These models typically are proprietary and exploratory, and lack formality and expressiveness. The HP's Cooltown project proposed a web-based context model in which each project (person, place and thing) has a corresponding web description that can be retrieved using a URL. The Context Toolkit project transmits low-level context acquired form physical sensors to the form of XML-encoded name-value pairs.

Model-based method: This category o models commonly uses conceptual modeling methods to represent context. A formal context model based on ER model was proposed by several projects[9,10], and context can be easily managed with relational databases. Henricksen et al.[11] model contexts and their additional features (classification and temporal characteristics) using both ER model and UML diagrams. This model was further reformulated with the extended Object-Role Modeling(ORM)[12].

Ontology-based method: Some work in the field of context-awareness ignore issues about quantitative concepts including temporal characteristics and quality of context, and focused more on constructing an ontology for context in a specific domain to reach the goals of knowledge sharing across distributed systems. The Comprehensive Structured Context Profiles (CSCP)[13] was developed based on RDF to represent context by means of session profiles. Chen et al. defined a context ontology based on OWL to support ubiquitous agents in their Context Broker Architecture (CoBrA), this context ontology only covers contexts in campus space, while has no explicit support for modeling general contexts in heterogeneous environments. Rangannathan et al.[14] developed a middleware for context awareness and semantic interoperability, in which they represented context ontology using DAML+OIL.[15]

We conclude about the three categories mentioned above, the application-based method lacks the formal basis and does not support knowledge sharing across different domains. Though the model-based projects support formality and some of them capture temporal aspect of context information, they do not address issues including knowledge sharing and context reasoning. The ontology-based method focuses on context ontology and explores the potential capability of context reasoning based on Semantic Web technologies. However, the existing context ontologies lack of generality and have not addressed important issues including context classification, context dependency and quality of context which will be useful in context reasoning. In this paper, we present our ontology-based context model using OWL that addresses these shortcomings.

Ⅲ. THE CONTEXT ONTOLOGY AND OWL

Semantic Web is a vision of the next generation World Wide Web in which knowledge is represented in well-defined ways, to better enable people and computers to work together. The origin of the Semantic Web research goes deep in the roots of Artificial Intelligent research. From 2001, the World Wide Web Consortium (W3C) established the Web Ontology Working Group with the goal of introducing Semantic Web language to the web research community. The group has specified OWL which is based on DAML+OIL. Compared with RDF of RDF-S, the OWL is much more expressive, it allows us to build more knowledge into the ontology. OWL is a language for defining ontology which is referred as the shared understanding of some domains. And from a system implementation point of view, there are a lot of ontology inference engines in support for the OWL language (e.g.,FaCT,RACER and so on).For this reason, we choose the OWL to model context ontology.

The diversity of the context knowledge makes the expressing of each domain in different way, how to extract the commonness from the different domain to form a unified abstract logic model, so as to make the context modeling and reasoning in one framework, is an important problem in the context modeling. The ontology technique can offer the common understanding of the domain knowledge, and give the formal semantic meaning to the terminology of the model, so we can use the ontology to build the context model to deal with the problem of context-aware knowledge acquisition.

Our method tackles the problem in two steps. First, we build the context model which dividing the context into two-level structure as Meta Ontology and Domain-specific Ontology using ontology, and then formalize the model in the way of OWL DL.

A. Context Modeling Framework

In order to raise the re-usability and flexibility, the main effect of our context model framework is to unify the overall infrastructure of different domains, makes the development of context model within the same structure. There are several definitions about context so far, we take the widely accepted one: Context is used in depicting the physical entity, including person, equipment, location, time etc. Furthermore, except the entity mentioned above, we believe the component of context should also include the goals of the user 、 the mission and the relations between them. With these component, context-aware computing can not only aware the explicit knowledge but also the implicit knowledge hided in those explicit knowledge.

As we mentioned above, it is difficult to express components from different domains in the same hierarchy level. Based on [16], and take the advantage of ontology and architecture framework technique, we propose a context model framework which dividing the context model into two-level structure. Fig. 1 shows the details of the framework. The upper ontology is a high-level ontology which captures general features of context entities. The lower ontology is a collection of ontology set which defines the details of the context concepts and their features in each specific domain, such as communication domain, traffic domain, and agriculture domain.

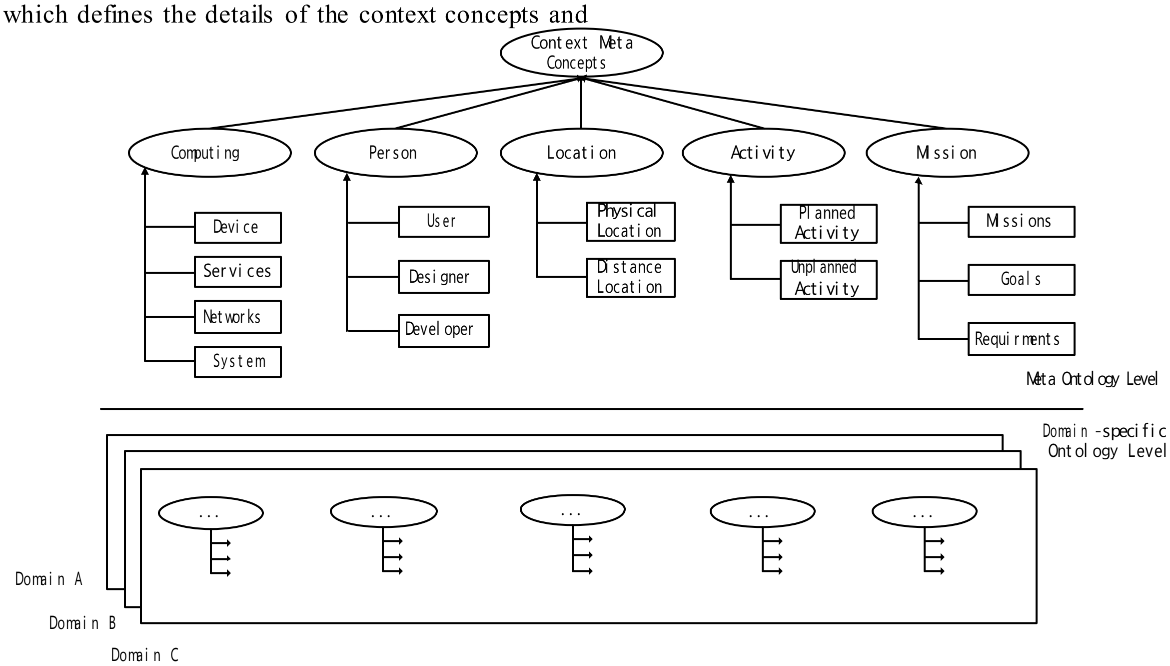

Figure 1 Context modeling Framework

B. Meta Context Ontology

The object of the higher-level Meta context ontology is to define context Meta concepts and their relations clearly, to forms a formal context knowledge structure, and promises the veracity and pertinence of the development based on context. The concepts in this level should cover all domains, so the Meta context ontology extracts the basic element of the context.

Definition 1. Meta Context Ontology is a tuple with four factor : <MetaConcepts, MetaRelations, MetaAttributes ,MetaRules>.

Fig. 2 shows the details of the Meta context ontology. MetaConcepts is a finite set of the context ontology concepts from the Meta ontology level; MetaRelations is a finite set of the context ontology relations between MetaConcepts; MetaAttributes represent the attributes of the MetaConcepts and MetaRelations, such as the name of the concept and relation; MetaRules represent the basic rules of the MetaConcepts and MetaRelations must satisfy. As we see in Figure1, the Meta context Ontology has the inherited mechanism, all new sub-class inherits from the Meta Context Ontology, such as sub-concept "Device" inherits from MetaConcepts "Computing".

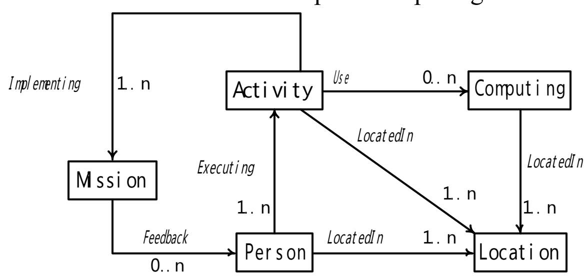

Figure 2 Meta Context Ontology

We use OWL DL represent the Meta context ontology, some demonstrations will be given to illustrate how OWL DL represent the ontology, including MetaConcepts、MetaRelations and MetaAttributes. There is a MetaRelation between MetaConcepts "Person" and "Activity" called "Employing", the OWL DL form is:

```
<owl:ObjectProperty rdf:about="#Employing">
<rdfs:range rdf:resource="#Computing"/>
<rdfs:domain rdf:resource="#Activity"/>
</owl:ObjectProperty>
```

The MetaConcepts "Computing" has MetaAttributes "ChannelStatus" which type is "string", the OWL form is:

```
<owl:DatatypeProperty rdf:ID="ChannelStatus">
<rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSche
ma#string"/>
<rdfs:domain rdf:resource="#Computing"/>
</owl:DatatypeProperty>
```

The sub-concept "Device" inherits from MetaConcepts "Computing", the OWL form is:

```
<owl:Class rdf:ID="Device">
<rdfs:subClassOf>
<owl:Class      rdf:about="#Computing"/>
</rdfs:subClassOf>
</owl:Class>
```

C. Domain-Specific Context Ontology

Domain-specific context ontology defines the concepts and relations within the given domain, and it is constrained by the Meta context ontology. The domain concepts is the instance of the MetaConcepts, for example, "communication channel" 、 "available bandwidth" is the instance of the MetaConcepts "Computing"; "routing switch" 、"service control" is the instance of the MetaConcepts "Activity". Fig. 3 shows a sample of context ontology in communication domain.

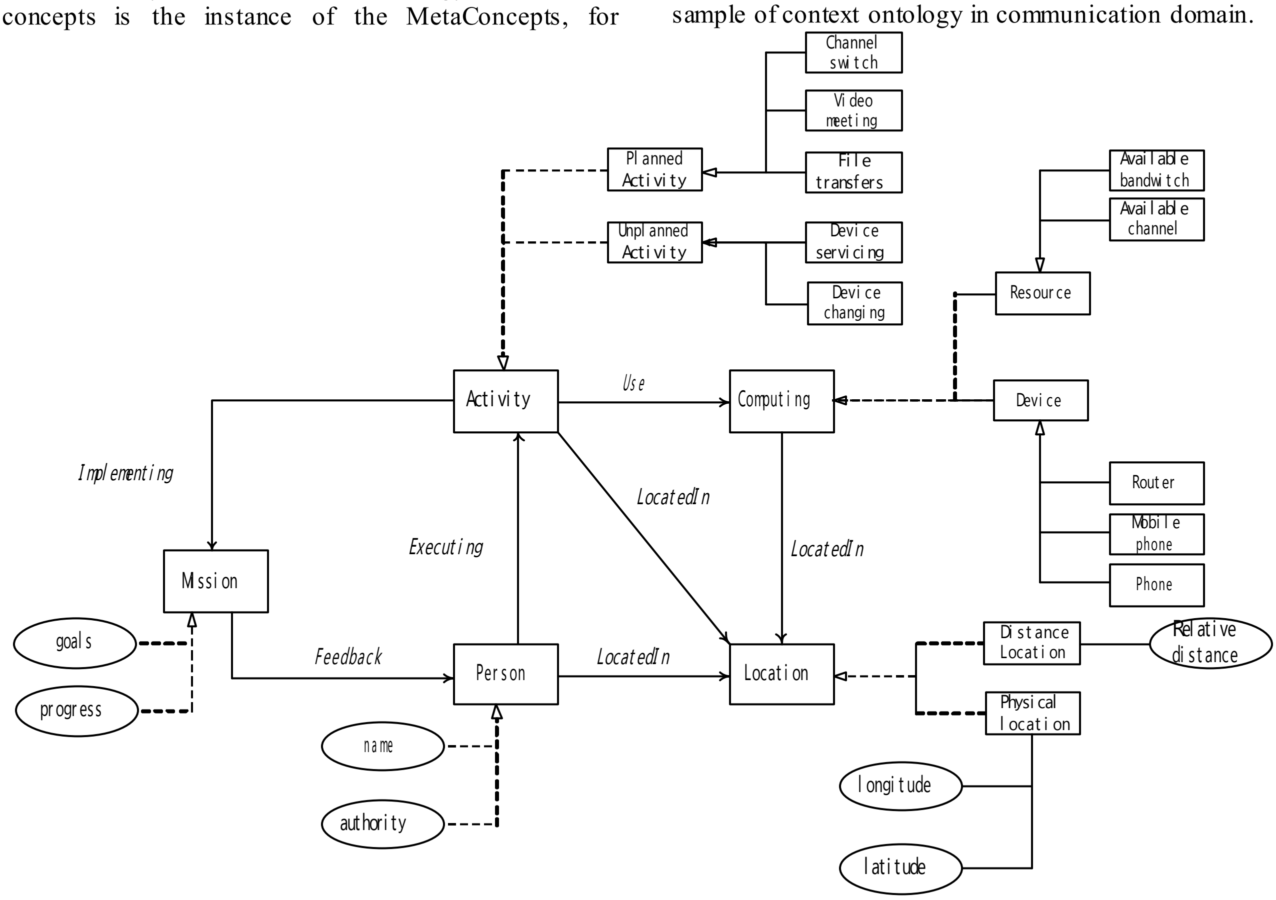

Figure 3 Communication domain context ontology

Definition 2. Domain-Specific Context Ontology is a tuple with four factor : <DomConcepts , DomRelations, DomAttributes, DomRules >.

DomConcepts is a set of concepts within specific domain; DomRelations is a set of relations between DomConcepts; DomAttributes represent the attributes of the DomConcepts and DomRelations; DomRules represent the domain rules of the DomConcepts and DomRelations must satisfy.

DomConcepts is the instance of MetaConcepts in special domain. OWL DL uses the expression Individual(o, type(C)) represents that individual o is the instance of concept C：

$$DomConcepts \equiv \{o \mid Individual(o, Type(MetaConcepts))\}$$

The rules set in context ontology including MetaRules and DomRules is another important part of the context, the context-aware computing infer implicit knowledge from explicit knowledge according to the rules define. The form of rules is different from concepts、relations and attributes, we will discuss the rules separately later.

D. Defining Rules

The reasoning technique based on rules are often researched in Artificial Intelligence 、 Expert Systems, recently some researchers introduce the technique into context-aware computing. Yang[17] implements a context-aware system based on Jess rules engine, his system uses the context knowledge such as location、device、the environment of Web Services implementing、communication protocol and so on to reasoning the implicit knowledge according to the rules he defining . But, the Jess rules engine does not have the formal criterion, lack the universal semantics.

SWRL（semantic Web rule language）[18] is a Web rule describing language put forward by W3C, coming from the initial thought in DAML rule language design, it combines the advantages of OWL DL and RuleML. Most of all, SWRL is compatible with Description Logic, so we can use the DL automatic reasoning tool and SWRL to fulfill our context reasoning. In our work, we use SWRL to define the rules in context ontology.

The rules in context ontology including MetaRules and DomRules, they are a class of constraints that concepts 、 relations and attributes must satisfy, their effects just as the switch sentence "if …then…". For example, one rule in MetaRules defines that, the relation "Executing" between concept "Person" and "Activity" must have at least one "Person". We formalize that rule in First-Order Predicate Logic now, and transfer the rule to SWRL in next section.

$$\forall x.Activity(x) \rightarrow (\#\{y \mid Person(y) \wedge Executing(x, y)\} \geq 1)$$

The MetaRules are the basic rules of the context ontology, at the same time, DomRules are user-defined rules focusing on specifical domain. For example, a DomRules defines three precondition:1)There are at least one people sit in the meeting room, and has a mission "ConnectingSomebady"; 2)There is a phone in this meeting room; 3)The communication channel of the phone is busy. If these precondition are satisfied, we can get the conclusion that this people is executing a activity(Calling). The First-Order Predicate Logic form of this rule is:

$$\begin{aligned} &\forall x, y, z\{Person(x) \wedge MeetingRoom(y) \wedge \\ &Telephone(z) \wedge LocatedIn(x, y) \wedge LocatedIn(z, y) \\ &\wedge ChannelStatus(z, "BUSY") \wedge \\ &HasMission(x, ConnectingSomebady)\} \\ &\rightarrow \exists a\{Activity(a) \\ &\wedge Executing(x, a)\} \end{aligned}$$

IV. CONTEXT REASONING

In context-aware computing, the context knowledge can be divided into explicit knowledge and implicit knowledge. The explicit knowledge such as "the channel is busy", "somebody in the room", "the bandwidth is 10M/S" can be collected from sensors or RFID equipments, those knowledge compose the low-level context, the role of context reasoning is to deduce high-level, implicit context from those low-level context. The implicit context sometime is much useful for application, such as "the meeting is going on", "the channel should change". The most universal context reasoning technique actually is to get the high-level context through infer the explicit context whether satisfy the precondition which rules define. Considering the decidability and formal semantics, we believe that the OWL DL which supporting the Description Logic(DL) together with SWRL are very powerful tools for reasoning with context knowledge, and they are sufficient for general pervasive context-aware systems as demonstrated later.

According to the summarization of the existing context reasoning approaches, we purpose a reasoning method based on DL. Compare with the former research, our method not only has the advantage of decidability and high efficiency, but also has a layered semantic structure. When the application domain has changed, we just need to change the Domain-Specific ontology not to change the reasoning method itself.

In our method, we use the automatic ontology reasoning tool Pellet to implement context reasoning. Before that, we design a transfer algorithm to map the context ontology represented by OWL DL to Description Logic, and the mapping progress can be dividing to three parts:1) The DL instance set Abox is filled with concepts , relations and attributes of the Domain-Specific context ontology;2) The DL axiom set Tbox is filled with concepts, relations and attributes of the Meta context ontology; 3) The MetaRules and DomRules are represented in SWRL, and then we use the ontology query language SPARQL to exam whether rule be achieved. The context reasoning method based on DL can fulfill important logical requirements, include concept satisfiability, concept subsumption, concept consistency and instance checking. We will discuss the details of the convert algorithm later.

A. The Convert Algorithm of the Context Ontology to DL

Fig. 4 shows the flow of the algorithm.

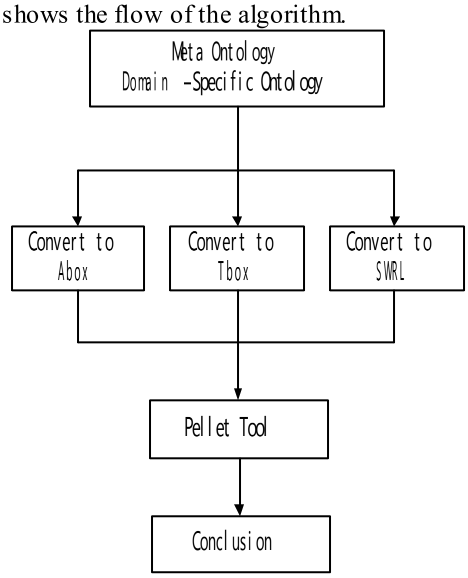

Figure 4 Flow of the convert arithmetic

The automatic conversion can be made in following steps.

Algorithm: Convert the ontology to DL formalization

Input:Meta: Context ontology and Domain-specific context ontology(in the form of tuple)

Output: Context ontology represent by DL and SWRL

Step 1: Create the Tbox.

Step1.1 For every MetaConcepts C in Meta ontology, creat a same name concept C in Tbox.

Step1.2 For every MetaRelations R between MetaConcepts A and B, creat a same name relations R in Tbox, then append an axiom $A \subseteq \forall R.B$ to Tbox.

Step1.3 For every concept B which is the sub-concept of the MetaConcepts A, append an axiom $B \subseteq A$ in Tbox.

Step1.4 For every attribute X (suppose the data type is T) of the MetaConcepts A, append an axiom $A \subseteq X.T$ in Tbox.

Step 2: Create the Abox.

Step2.1 For every DomConcepts c in Domain-Specific ontology, creat a same name instance of concept c in Abox.

Step2.2 For every DomRelations r between DomConcepts a and b, creat a same name instance of relation r in Abox.

Step2.3 For every attribute x in the Domain-Specific ontology,creat a same name instance3 of attribute x in Abox.

Step 3: Create the SWRL rules define.

Step 3.1 Change the precondition of the rules to antecedent of the SWRL: For every relation R between concept X and Y in the precondition, change the R to the predicate R(x,y) as the clause of the antecedent, x and y is the instance of the concept X and Y. If there are several clauses, add operator "And" ("∧") between the clauses.

Step 3.2 Using the same way, change the conclusion of the rules to the consequent of the SWRL.

To explain the progress of the algorithm, we will show a case study in next section.

B. A Case Study on Context Reasoning

We present a calling scenario in which the context reasoning infers from the implicit context knowledge detected by sensors to the explicit knowledge "somebody is calling". Applying the convert algorithm we purpose before, we can get the SWRL rule of the "somebody is calling".

LocatedIn（Person P， Location MeetingRoom）∧ LocatedIn (Device Telephone，Location MeetingRoom) ∧ChannelStatus（Device Telephone， "BUSY"）∧ HasMission（Person P，Mission ConnectingSomebady) →Executing（Person P， Activity Call）

The instance is coming from Domain-Specific context ontology: P is the instance of MetaConcepts Person; Telephone is the instance of MetaConcepts Device(concept Device is the sub-concept of Computing); MeetingRoom is the instance of MetaConcepts Location; Call is the instance of MetaConcepts Activity; ConnectingSomebady is the instance of MetaConcepts Mission.

The predicate LocatedIn, Executing is the MetaRelations, according to our method, the Tbox is filled with axiom of the predicate, and Abox is filled with the instance of the predicate.We can take the DL form of this context model into Pellet tool, now we can ask the Pellet about the question who the person is calling the phone. Fig. 5 shows the partial of the axioms and instances in Tbox,Abox.

Tbox:

Device⊆ *Computing*

Person⊆∀*LocatedIn*.*Location*

Computing⊆∀ *LocatedIn Location*

Person⊆∀*HasMission*.*Mission*

Device⊆ *ChannelStatus*.*string*

Person⊆∀*Executing Activity*

Abox:

< *LiMing* >: *Person*

< *WuGang* >:*Person*

< *Telephone* >: *Device*

< *Call* >: *Activity*

< *MeetingRoom*>: *Location*

< *ConnectingSomebady* >:*Mission*

< *LiMing*, *MeetingRoom*>: *LocatedIn*

< *WuGang*,*MeetingRoom* >:*LocatedIn*

< *Telephone*, *MeetingRoom*>: *LocatedIn*

< *Telephone*, *Busy* >:*ChannelStatus*

< *LiMing*, *ConnectingSomebady* >: *HasMission*

Figure 5 Partial of the the Tbox and Abox after converting

We use Pellet 1.5.0 tool to reasoning the context model. We define the query language using SPARQL to examine whether instance is executing any activity. The SPARQL is "SELECT ?a ?b WHERE {?a.xmlns:Executing ?b. }", After entering the SPARQL in Pellet console, we can get the query result of context reasoning which showed in console like Fig. 6, the person "LiMing" is executing a activity "Call". This case study is a small context model to illustrate the feasibility of our method, we can construct a lot of similar example under our context modeling framework and reasoning method.

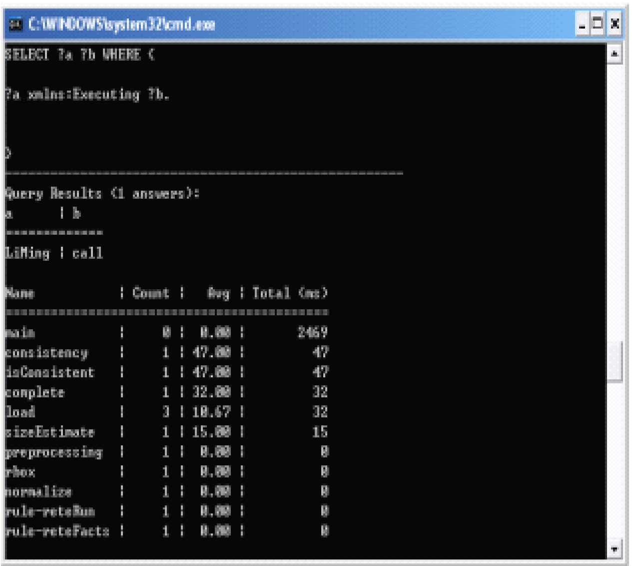

Figure 6 The result of context reasoning using Pellet

Ⅴ. CONCLUSION AND FURTURE WORK

Context modeling and reasoning is important parts of context computing, present research focus on how to raise the model reusability and reason automatization. The paper proposes an approach of context modeling and reasoning based on description logic, compared with the existing approach, our method has several advantages: 1) The context model framework has distinct theoretical model mechanism with reliable semantics; 2)The two-level context model using conception taxonomy to provide reusability and flexibility of context modeling; 3) The reasoning approach based on description logic provide decidability automatic reasoning support, which make the context reasoning more reliable, complete and sensible, and more useful to context computing.

The context modeling and reasoning approach for pervasive application attempts to cover the context-aware services requirements for modeling and structuring a formal, extensible and efficient context information model to collect, distribute and store information among communication systems and along of their applications attempting for solving one of the main problems in the pervasive applications area[19].

As part of our ongoing work, we are continuing to develop context-aware applications using our approach, not only in communications but also the other domains. We are also designing the extension to our context modeling framework and programming toolkits to provide efficient support and improved imperfect context knowledge. Finally, the use of a consistent context-aware application development has also allowed us to introducing tools that reduce the overall effort in programming context-aware applications by automating aspects of deploying and programming using our context framework.

ACKNOWLEDGMENT

This work is supported by the National High-Tech Research and Development Program of China (2007AA01Z126).

REFERENCES

[1] N. Gui.,et al., "Toward architecture-based context-aware deployment and adaptation," *Journal of Systems and Software*, 2011, pp.185-197.

[2] O.Kwon, "The potential roles of context-aware computing technology in optimization-based intelligent decision-making, " *Expert Systems with Applications*, 2006(31):629-642.

[3] K. Henricksen and J. Indulska, "Modelling and using imperfect context information," *In Proceedings of the 2nd IEEE Annual Conf.on Pervasive Computing and Communications Workshops*,pages 33–37,March 2004.

[4] Q.Z.Sheng and B. Benatallah, "ContextUML:A uml-based modeling language for model driven development of context-aware web services," *In Proceedings of the International Conf.on Mobile Business*,pages 206–212,July 2005.

[5] B. Benatallah, M. Dumas and Q. Z.Sheng, "Facilitating the Rapid Development and Scalable Orchestration of Composite Web Services," *Distributed and Parallel Databases, An International Journal*, vol:17(1):5-37,2005.

[6] T.Gu, H. Keng. Pung, and D. Q. Zhang, "A middleware for building context-aware mobile services," *In Proceedings of IEEE 59th Vehicular Technology Conference*,2004,volume 5,pages 2656–2660,May 2004.

[7] D. Liu, X. W. Meng, J. L. Chen, X. M. Xia, "Algorithms for Rule Generation and Matchmaking in Context-Aware System," *Journal of Software*, Vol.20, No.10, October 2009, pp.2655-2666.

[8] H. Chen, S. Tolia, C. Sayers, T. Finin, and A. Joshi, "Creating context-aware software agents," *In proceedings of the First GSFC/JPL Workshop on Radical Agent Concepts*,2001.

[9] A. Harter, A. Hopper, P. Steggles, A. Ward, P. Webster, "The Anatomy of a Context-Aware Application," *Wireless Networks* 8(2-3):187-197(2002).

[10] H. Wu, M. Siegel, and S. Ablay, "Sensor Fusion for Context Understanding," *In proceedings of IEEE Instrumentation and Measurement Technology Conference*, Anchorage, USA, May 2002.

[11] K. Henricksen, J. Indulska and A. Rakotonirainy, "Modeling Context Information in Pervasive Computing Systems," *In proceedings of IEEE International Conference on Pervasive Computing and Communications*, Zurich, March 2003.

[12] K. Henricksen, J. Indulska and A. Rakotonirainy, "Generating Context Management Infrastructure from High-level Context Models," *In proceedings of the 4th International Conference on Mobile data Management*, Melbourne, January 2003.

[13] A. Held, S. Buchholz, A. Schill, "Modeling of context Information for Pervasive Computing Applications," *In proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics(SCI)*, Orlando, FJ, July 2002.

[14] A. Ranganathan and R. H. Campbell, "A Middleware for Context-Aware Agents in Ubiquitous Computing Environments," *In proceedings of ACM/IFIP/USENIX International Middleware Conference*, Rio de Janeiro, Brazil, June 2003.

[15] I. Horrocks, "DAML+OIL: A Reason-able Web Ontology Language," *In proceedings of the 8th International Conference on Extending Database Technology*, Prague, March 2002.

[16] D. Zheng, "Research on Key Technologies of Component Oriented Middleware based on the Context-aware Service," *Thesis of National University of Defense Technology*, Changsha, China, 2008.

[17] S. J. H. Yang, J. Zhang, I. Y. L. Chen, "A JESS-enabled context elicitation system for providing context-aware Web services," *Expert Systems with Applications*, 2008,34(4):2254-2266.

[18] B. Parsia.,et al., "Cautiously Approaching SWRL," *Preprint Submitted to Elsevier Science[EB/OL].* (2005-04-19).http://www.mindswap.org/papers/Cautious SWRL.pdf.

[19] T. McFadden, K. Henricksen and J. Indulska, "Automating context-aware application development," *In Ubicomp'2004 Workshop on Advanced Context Modeling, Reasoning and Management*, Nottingham, September 2004.

Bo Hu was born in Nanjing,Jiangsu province,1983. He is a PhD candidate of Institute of Command and Automation, PLA University of Science and Technology University. He received B.S. and M.S. degree from Hefei Electronic Engineering Institute. His research interests are software engineering and computer network currently focusing

on pervasive computing and intelligent information system modeling.

Zhixue Wang was born in 1961.He is a professor of Institute of Command and Automation, PLA University of Science and Technology. He received B.S. degree from Hefei Polytechnic University, M.S. degree from National University of Defense and Technology, and used to be a visiting researcher in Faculty of Information Technology, University of Brighton, England. His research interests are software engineering, requirements engineering, theory and technology of command automation, currently focusing on domain-specific modeling and formal verification.

Qingchao Dong was born in 1982. He is a PhD candidate of Institute of Command and Automation, PLA University of Science and Technology University, where he received B.S. and M.S. degrees. His research interests are requirements engineering, focusing on specification and formal verification.