# A Context Model for Service Composition Based on Dynamic Description Logic

**Wenjia Niu[1,2], Zhongzhi Shi[1] and Liang Chang[1,2]**

[1]Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences,100080,Beijing,China

[2]Graduate School of the Chinese Academy of Sciences,100039,Beijing,China
{niuwenjia, shizz, changl}@ics.ict.ac.cn

Abstract: A service composition task for service broker is to discovery and compose provider's services to satisfy user's request. Many researchers model the context utilizing ontology-based or attribute-based method to assist service composition. We propose a new context model by combining the context logic with the dynamic description logic (DDL), where user' context, provider's context and broker's context are described by DDL separately and reasoned under the context logic. The reasoning results finally can be used to discovery and compose services intelligently. We evaluate this model on a simple, yet realistic example, and the results show that our context model provides a practical solution.

Key words: context model, context logic, semantic web service, DDL

## 1. Introduction

Since the term *context-aware computing* was first introduced in 1994 [15], a large number of definitions of the term *context* have been proposed in the area of computer science. Zimmermann [19] proposed an operational definition of context based on Dey's work [7], in which the context is

"*Context is any information that can be used to characterize the situation of an entity . Elements for the description of this context information fall into five categories: individuality, activity, location, time, and relations. The activity predominantly determines the relevancy of context elements in specific situations, and the location and time primarily drive the creation of relations between entities and enable the exchange of context information among entities*".

The context information in the semantic web services has been modeled to help discovery and compose services recently. However, this context information is rarely modeled as uniform context logic. For instance, the user preference context is modeled using description logic [3]. In general, the methods of modeling context fall into two categories: logic-based [2] [3] and non-logic-based [8] [9] [10] (e.g., attribute-based and ontology-based). The logic-based context model lacks operational definition of context and the non-logic-based model lacks logic representing and reasoning on context, so we try to integrate context logic and operational context in our context model.

The context logic [4, 12] is an extension of first order logic in which sentences are not simply true, but are true within a context. The key extension is a modality ***ist(context, formula)***, read "is true", which takes two arguments: a context and a formula. It asserts that the formula is true in the specified context. Contexts are logical individuals and, as such, can be quantified by logic languages. Description Logics (DLs) is a choice to describe contexts for its ability in representing and reasoning static knowledge. But in semantic web service, DLs cannot effectively represent and reason dynamic knowledge(e.g., service). A dynamic description logic (DDL) was proposed to represent and reason knowledge of static and dynamic [16], which can be taken as a proper logic base for semantic web services. So DDL is chosen to quantify the static and dynamic context information effectively. By combining the context logic and DDL, we proposed a DDL-based context model, in which web services are composed adapt to all contexts of user and provider and broker.

The remainder of this paper is organized as follows. Section 2 presents what's the context information of semantic web service composition. Section 3 presents the context modeling based on DDL and the context logic theory in semantic web services composition. In Section 4, we discuss the evaluation of our model through context reasoning in a realistic example. Section 5 overviews related work and conclusions.

## 2. Context in Semantic Web Services

According to the operational definition above, there are five main elements for description of an entity context[19]: *individuality*, *time*, *location*, *activity* and *relations*.

In the web service composition process, there exist three roles: the user, the service provider and the service broker. We generalize three contexts corresponding to the three roles separately, which are user context, provider context and broker context. The attributes of user context include user profile, user preference, time, location, and goal. The attributes of provider context include provider profile, time, location, and action. The attributes of broker context include broker profile, time, location, and resources. Fig.1 shows the description of context attributes of each context.

**User Context**
- ***User profile:*** corresponds to the user's personal information.
- ***User preference:*** corresponds to the user's preference on the service he wants to get.
- ***Time, Location:*** inform use's time and location when sending a service request.
- ***Goal:*** indicates what the users want to get from services.

**Provider Context**
- ***Provider profile:*** corresponds to the provider information, such as provider name.
- ***Time, Location:*** inform service provider's time and location when receiving a request.
- ***Action:*** indicates the function description of service.

**Broker Context**
- ***Broker profile:*** corresponds to the broker information, such as broker name.
- ***Time, Location:*** indicates broker's time and location when receiving a request from user.
- ***Resource:*** indicates service status and user status in composition process.

Fig. 1. Description of Context Attributes

Different from [13][14], the service's function description is defined as a context attribute in our model for two reasons. Firstly, according to the definition of context [19], the service's function is a kind of information that can be used to characterize the situation of the service. Secondly, the context information should be used for not only personalized application but also functionally composing web services.

## 3. DDL-based Context Model

In this section, we present our idea of extending the classical architecture of web services, which takes into account a context model of the service composition process. Then we introduce the context representation using the DDL language and the context reasoning in our context model.

### *3.1. Context model in Web Service Architecture*

Traditionally, the architecture of WS (see Fig.2) is composed of three entities: the service provider builds the service and publishes its description to a service broker. The user needs are translated into requests that are carried on by the broker. Once the service is found, the user will obtain direct interaction with the service.

Our contribution aims to add to this architecture a context model, which is dedicated to context representation and reasoning. This model is centralized in the broker. In Fig.2, different steps are proposed to integrating the context model in the classical architecture of web services. The different steps are: 1) Provider describe their services using web service description language (WSDL). 2) The user launches his request to the broker (with format SOAP). 3) The context model (CM) captures the users' context. 4) The CM captures the providers' context. 5) The CM captures the broker's context. 6) The CM logically represents and reasoning the contexts, and the reasoning results are transformed into a service composition scheme and delivered to the user. 7) The communication between the user and the provider is done in a traditional way via SOAP.

Our context model consists of two function modules: representing module is responsible to give a logic formalization of context and reasoning module is responsible to reason on context. These two modules can be integrated into a uniform context logic system, meant as the triple $\Sigma=(L,\Lambda,\Delta)$, where $L$ is a context logic language, $\Lambda$ is a set of axioms and $\Delta$ is a set of reasoning rules. As mentioned in Sec.1, the key syntax of context logic is ***ist(context, formula)***, which ***context*** represents a logical individual and, as such, will be described by the DDL language. Context embodies an individual's subjective perspective which characterizes the individual's situation, so user's context, provider's context and broker's context are described separately by the DDL language, but logically connected by *bridge rules(BR)* in context logic system. A distributed reasoning algorithm is taken to reason about contexts of user's, provider's and broker's. As for the capture of context and the transformation between logic language and SOAP format, they are out of this paper's scope..

## *3.2. Context Representation*

A DDL knowledge base consists of a TBox, an ABox and an ActionBox [5]. The Tbox contains assertions about concepts (e.g., *Person*) and roles (e.g., *hasAge*). The ABox contains assertions about individuals (e.g., *PETER*). The ActionBox contains assertions about actions (e.g., *BuyMovieTicket(JOHN, TICKET)*). $\pi$ is a action. An atomic action is a pair *(P,E)*, where, *P ,E* are two finite set of formulas used to describe precondition and effect accordingly.

We depict a simple scenario to show how to describe contexts in web services composition and what's the difference of each context.

***Example 1.***(The movie scenario) PETER are going to see a movie when he is driving, so he would like to get the movie information and buy a ticket online. To achieve this, he will publish his request to a service broker through his smart phone. After receiving the request, the broker will try to find and compose proper services for PETER. There exist two services *BuyTicket* service and *GetMovieInfor* service provided by a provider, which can meet the PETER's request. According to the TBox, contexts are described by the DDL language as follows.

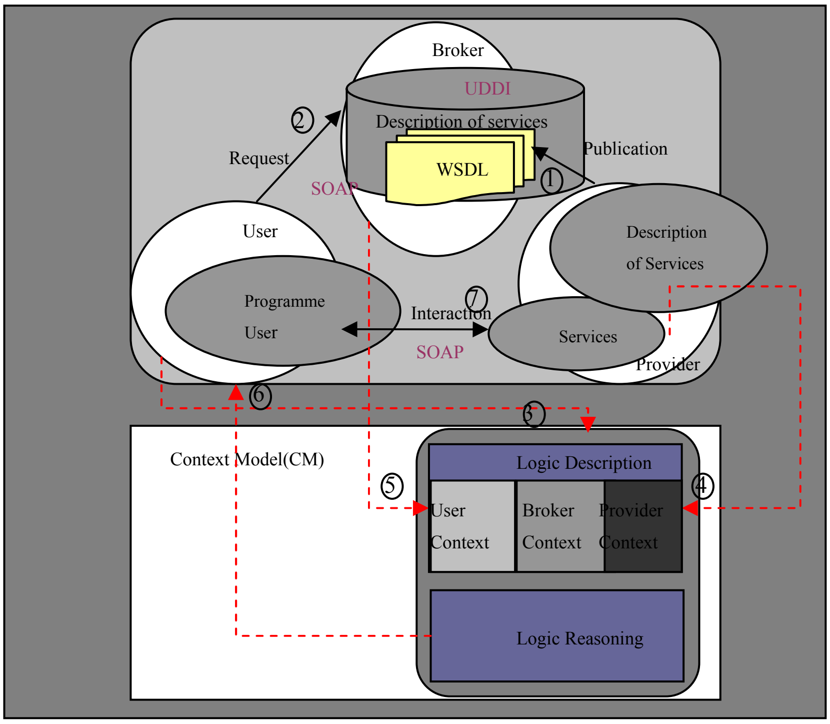


Fig. 2. Architecture of web service based on context

- **User context (*uc*)**

  Profile: $Person(PETER) \cap (InCar(PETER)) \cap Male(PETER) \cap (\exists hasAge.\{20\}(PETER)) \cap hasMoney(PETER,230)$

  Preference: $Movie(x) \cap \exists hasMovieGenre.\{Love\}(x)$

  Time: $BeijingTime(17{:}00)$

  Location: $District(HAIDIAN)$

  Goal: $Own(PETER,x) \cap Ticket(x) \cap hasInformation(PETER,y) \cap Information(y)$

- **Provider context (*pc*)**

  Profile: $ProviderName(PEOPLEMOVIE)$

  Time: $BeijingTime(17{:}00)$

  Location: $District(XUANWU)$

  Action: $BuyTicket(x,y) \equiv (\{Person(x), \neg Own(y), Ticket(y), hasMoney(x,z), Money(z)\}, \{Own(y)\})$

  $GetMovieInfor(x,z) \equiv (\{Person(x)\}, \{Movie(z), MovieGenre(y), (\exists hasMovieGenre.\{y\}(z)), BeginTime(t), (\exists hasBe-$

*ginTime.{t}(z)),TicketPrice(p), (∃hasTicket-Price.{p}(z))})*

- **Broker context (*bc*)**
  Profile: *BrokerName(SWSBROKER)*
  Time: *BeijingTime(17:00)*
  Location: District(XICHENG)

## *3.3. Context Reasoning*

In our context model, there are two intuitive patterns of contextual reasoning: localized reasoning and transform reasoning. With these two kinds of reasoning, the context reasoning can operate in a single context as well span several contexts.

**1)Localized Reasoning**

Localized reasoning refers the reasoning process is always in a single context, which contains whatever the reasoning process needs. Since a context is described by a DDL language in our context model, localized reasoning can be operated in a DDL reasoning system in which basic reasoning in DL and action reasoning are typically supported.

Action reasoning plays an important role in localized reasoning. There are four kinds of action reasoning: realizability, executability, projection and plan. To understand how the reasoning works, we still use the example mentioned in Sec.3.2 and suppose that the $D_{S,}$ the set of formulas to describe the state, is: *{Person(PETER)*,*(InCar(PETER))*,*Male(PETER), (∃hasAge.{20}(PETER))*,*hasMoney(PETER,230), Ticket(TITANIC-TICKET),¬own(TITANIC-TICKET), Movie(TITANIC), hasTicket(TITANIC,1);* The TBox $D_T$ is showed in Fig.4 and The RBox $D$ is supposed to be *null.*

*Realizability*: An action π is realizable *w.r.t.* the RBox $D_R$ and TBox $D_T$ iff there exists a model $M=(W,I)$ of both $D_R$ and $D_T$ such that there exists some states $w,w'\in W$ with $(w,w')\ \in\pi^I$. *Executability*: An action π is executable on states described by $D_S$ iff for any model $M=(W,I)$ of both $D_R$ and $D_T$, and for any state $w\in W$ with $(M,w)|=D_S$, there exists a model $M'=(W',I')$ of both $D_R$ and $D_T$,such that $W\subseteq W',I'(Wi)=I(Wi)$, for each $(M',w)|=D_S$, and $(w,w')\ \in\pi^I$ for some state $w'\in W'$. In the example 1, the action "*BuyTicket(PETER,TITANIC-TICKET)*" is executable, but the complex action "*BuyTicket(PETER,TITANIC-TICKET), BuyTicket(PETER, TITANIC-TICKET)*" is not executable.

*Projection*: A formula $\psi$ is a consequence of applying π on states described by $D_S$ iff for any model $M=(W,I)$ of both $D_R$ and $D_T$, and for any states $w,w'\in W$:if $(M,w)|=D_S$ and $(w,w')\ \in\pi^I$., then $(M,w')|=\psi'$.

*Plan*: Let $\psi$ be a formula and $\Sigma$ be a set of actions. Let $\pi_1,\ldots\pi_n$ be a sequence of actions with each action coming from $\Sigma$. Then, the sequence $\pi_1,\ldots\pi_n$ is a plan for $\psi$ relative to $D_S$ iff (i) the sequence-action $\pi_1,\ldots\pi_n$ is executable on states described by $D_S$ and (ii) $\psi$ is a consequence $\pi_1,\ldots\pi_n$ of applying on states described by $D_S$. For example, the action sequence "*BuyTicket(PETER, TITANIC-TICKET), GetMovieInfor(PETER, TITANIC-INFOR)*" is a plan for the goal *own(PETER,TITANIC-TICKET)*$\cap$*hasInformation(PETER, TITANIC-INFOR).*

**2)Transform Reasoning**

Context bridging allows us to state that a certain property holds between elements of two different contexts. In our model, the basic notion toward the definition of bridge rules are: a bridge rule from *i* to *j* is a statement of one of the six following forms,

$$i:C\xrightarrow{\subseteq}j:E,\; i:C\xrightarrow{\supseteq}j:E,\; i:C\xrightarrow{\equiv}j:E,$$
$$i:\alpha\xrightarrow{\subseteq}j:\beta,\; i:\alpha\xrightarrow{\supseteq}j:\beta,\; i:\alpha\xrightarrow{\equiv}j:\beta$$

, where $C$ and $E$ are either concepts or roles of the DDL languages $DDL_i$ and $DDL_j$ respectively, α and β are actions of $DDL_i$ and $DDL_j$ respectively.

The idea of transform reasoning is mapping a context logic into a global $DDL_G$, then utilizing the $DDL_G$'s reasoning to realize the context reasoning, which is similar to the reasoning of distributed description logic[1]. Suppose the family of the dynamic description logic is $\{DDL_i\}(i\in I)$, the bridge rules are $BR_{i,j}(i,j\in I)$, the individuals are $IN_{i,j}(i,j\in I)$, we proposed a reasoning algorithm named transform reasoning(see Algorithm 1).

## 4. Case Study

We now put our context model on the simple scenario introduced in Example 1 to show how reasoning and service composition work.

***Example 2.***(The movie scenario revisited) The context logic contains three contexts: user context, provider context and broker context, which are described in DDL languages $DDL_{uc}$, $DDL_{pc}$ and $DDL_{bc}$ respectively. The bridge rules are :

$$uc:Information\xrightarrow{\supseteq}pc:\exists hasMovieGenre.MovieGenre$$
$$uc:Information\xrightarrow{\supseteq}pc:\exists hasBeginTime.BeginTime$$
$$uc:Information\xrightarrow{\supseteq}pc:\exists hasTicketPRice.TicketPRice$$
$$uc:Person\xrightarrow{\equiv}pc:Person, uc:Movie\xrightarrow{\equiv}pc:Movie$$
$$uc:MovieGenre\xrightarrow{\equiv}pc:MovieGenre, uc:BeijingTime\xrightarrow{\equiv}pc:BeijingTime$$
$$uc:District\xrightarrow{\equiv}pc:District, uc:Ticket\xrightarrow{\equiv}pc:Ticket$$
$$uc:Money\xrightarrow{\equiv}pc:Money, uc:Own\xrightarrow{\equiv}pc:Own$$

**Algorithm 1. Transform Reasoning**

```
//define an operator # from concepts/roles/actions of DDL_i to DDL_G
M are concepts/roles/actions in DDL_i
#(i :M) define i :M as primitive concepts/roles/actions in DDL_G
If (ρ is a concept constructor taking k arguments in DDL_i)
Then #(ρ(M_1,…,M_k)) =i:ρ(#( i:M_1) ,…,#( i:M_k))
//generate TBox in DDL_G
For axioms C ⊆ E in DDL_i
   add #(i :C) ⊆#(i :E) into DDL_G
End For
For each bridge rules
 IF(i:C --⊆--> j:E)Then add #(i :C) ⊆#(j :E) into DDL_G
 IF(i:C --⊇--> j:E)Then add #(i :C) ⊇#(j :E) into DDL_G
 IF(i:C --≡--> j:E)Then add #(i :C) ≡#(j :E) into DDL_G
End For
add ⊥⊆C, C ⊆⊤C into DDL_G ,C are concepts/roles/actions
//generate ABox in DDL_G
For C(a)∈ABox_i , where C are concepts/roles/actions in DDL_i
  add #(i:C(i:a)) into DDL_G
End For
Action reasoning in DDL_G
```

According to the transform reasoning algorithm, the context logic can be mapped into a global logic $DDL_{G,}$ in which the TBox is as follows:

*uc:Information⊒pc.∃hasMovieGenre.MovieGenre*

*uc:Information⊒pc.∃hasBeginTime.BeginTime*

*uc:Information⊒pc.∃hasTicketPRice.TicketPRice,uc:Person≡pc:Perso*

*uc:Movie≡pc:Movie, uc:MovieGenre≡pc:MovieGenre,*

*uc:BeijingTime≡pc:BeijingTime*

*uc:District≡pc:District, uc:Ticket≡pc:Ticket, uc:Money≡pc:Money,*

*uc:Own≡pc:Own.*

Suppose the state $D_s$ in $DDL_{pc}$ is:

*{uc:Person(uc:PETER),uc:InCar(uc:PETER)*,uc:*Male(uc:PETER),*

*uc:∃hasAge.{20}(uc:PETER)*,uc: *hasMoney(uc:PETER,uc:230),*

*pc:Ticket(pc:TITANIC-TICKET),uc:¬own(uc:PETER,uc:TITANIC-TICKET),*

*pc: Movie(pc:TITANIC),pc: hasTicket(pc:TITANIC,pc:1)}*

Finally, according to action reasoning in $DDL_G$, it is found that the action sequence

"*pc:BuyTicket(pc:PETER,pc:TITANIC-TICKET),pc:GetMovieInfor(pc:PETER,pc:TITANIC-IFOR)*" is a plan for the user's goal of getting the information and buying an online ticket: *uc:own(uc:PETER,uc:TITANIC-TICKET)⌒uc:hasInformation(uc:PETER,uc:TITANIC-INFOR).*

The reasoning results show that the two services : *GetMovieInfor* and *BuyTicket* can be composed to meet the user's request.

## 5. Related Work and Conclusion

The context has been modeled as ontology-based model or list of attributes in context-aware computing and web services: S.K.Mostefaoui[15] proposed a framework by combination of context-aware computing and agent technology, in which contextual information is exploited for service discovery and composition; Z.Maamar[11] proposed an agent-based and context-oriented approach, in which agent is characterized by context information; Chen[6] describe a framework for an agent based pervasive computing environment, in which contexts are explicitly represented using ontology languages allowing independently developed agents to exploit common ontologies to share knowledge and interoperate; Qiu[14] proposed an ontology-based framework for the context-aware composition of web services, where the context model are structured based on the upper ontology OWL-S.

In semantic web area, the context logic theory is successfully introduced to model context for building a contextualized ontology[2] (C-OWL), whose contents are kept local, and mapped with the contents of other ontologies via context mappings. [17] integrated a context model in web services, in which comprehensive structured context profiles(CSCP) format is used to describe context information. 6.Conclusion

In this paper, we proposed a new context model in semantic web services composition. This context model aims to deliver a list of adapted web services according to user's and provider's and broker's context. By combining the context logic with DDL, our model can discovery and compose web services through logical reasoning. To our best knowledge, the combination of context logic and DDL to assist and achieve the service composition is a new try in web services area, and the case study evaluates our approach effective.

## Acknowledgements

This work was partially supported by the National Basic Research Program of China (No. 2007CB311004), the National Natural Science Foundation of China (No. 90604017, 60775035) and the National High-Tech Research and Development Plan of China (No. 2007AA01Z132).

## References

[1]A.Borgida, L.Serafini: Distributed description logics: assimilating information from peer sources. Journal of Data Semantics,1(1), 153-184(2003).

[2]P.Bouquet, F.Giunchiglia, Harmelen.: C-OWL: Contextualizing Ontologies. International Semantic Web Conference 2003, 164-179(2003).

[3]A.H. Bunningen, L. Feng, and P.M.G. Apers, Using Description Logics to Model Context Aware Query Preferences. *CTIT Technical Report TR-CTIT-06-17*(2006).

[4]S. Buvac, I. Mason: Propositional logic of context. Proceedings of the eleventh national conference on artificial intelligence(1993).

[5]L. Chang, F. Lin, Z.Shi: A Dynamic Description Logic for Representation and Reasoning About Actions. *KSEM 2007*, 115-127(2007)

[6]H.Chen, T.Finin, A. Joshi: An ontology for context-aware pervasive computing environments. *Special Issue on Ontologies for Distributed Systems. Knowledge Engineering Review*, 3(18), 197-207(2004).

[7]A.K.Dey: Understanding and Using Context. *Personal Ubiquitous Computing* 5(1), 4–7(2001).

[8]S.Iga, M. Shinnishi, M. Nakatomi.: Context Gallery: A Service-Oriented Framework to Facilitate Context Information Sharing. *APWeb 2006*, 1096-1106(2006).

[9]O.Khriyenko, V. Terziyan: Context Description Framework for the Semantic Web, *Context Representation and Reasoning Workshop at CONTEXT 2005*.

[10]C.Lee and S.Helal: Context Attributes: An Approach to Enable Context-awareness for Service Discovery. *SAINT 2003*, 22-30(2003).

[11]Z.Maamar, S.K.Mostefaoui, H.Yahyaoui: Toward an Agent-based and context-oriented approach for Web services composition. *IEEE Transaction on Knowledge and Data ngineering* ,17(5), 686-697(2005).

[12]J. McCarthy: Notes on formalizing context. Proceedings of the thirteenth international joint conference on artificial intelligence(1993).

[13] S.K.Mostefaoui, B.Hirsbrunner: Toward a Context-Based Service Composition Framework. Proceedings of the International Conference on Web Services(ICWS'03), 42-45(2003).

[14]L.Qiu, L. Chang, F. Lin, and Zhongzhi Shi: Context Optimization of AI planning for Semantic Web Services Composition. *Journal of Service-Oriented Computing and Applications,* 117-128(2007).

[15]B.N.Schilit, Adams, N.I., R.Want: Context-Aware Computing Applications. *Proceedings of the Workshop on Mobile Computing Systems and Applications*, 85–90 (1994).

[16]Z.Shi, M.Dong, Y.Jiang, H.Zhang: A Logic Foundation for the Semantic Web. *Science in China*, Series F, 48(2), 161-178(2005).

[17]B.Soukkarieh, F.Sedes: Integrating a Context Model in Web Services. *ICWS* ,1195-1196 (2007).

[18]T.Strang, C.Linnhoff–Popien: A Context Modeling Survey. *Workshop on Advanced Context Modelling, Reasoning and Management* (2004).

[19]A.Zimmermann, Andreas Lorenz, and Reinhard Oppermann:An Operational Definition of Context. CONTEXT 2007, 558-571(2007).