J. Software Engineering & Applications, **2010, 3, 776-783**
doi:10.4236/jsea.2010.38090 Published Online August 2010 (http://www.SciRP.org/journal/jsea)

Applications of Norm and Situation Calculus in the Semantic Web Service Composition*

Jun Zhao[1], Xiaozhu Gu[2]

[1]School of Economics and Management, Ningxia University, Yinchuan, China; [2]School of Mathematical and Computer Sciences, Ningxia University, Yinchuan, China.
Email: wwwzhaojun@163.com, guxiaozhugood@163.com

Received June 23rd 2010; revised July 8th 2010; accepted July 24th 2010.

ABSTRACT

Semantic Web service studies are carried out around the OWL-S, however, OWL-S model does not contain the description of preferences and constraints when we choose and use them in different organizations, different cultures, different sectors and actors because of OWL capacity limitations in rules. It means that Semantic Web service composition can not compose users' subjective services from the user requirements. The paper makes up for the semantic shortcoming of OWL-S through using norm semantic scalability, and achieves the formal description of the norm by using the situation calculus. Finally, the paper also takes a tourism composite service as an example how to extend the ability of the semantic description of OWL-S, which shows us the combination of the norm and the situation calculus. It is positive significance for eventually realizing semantic extension of OWL-S.

Keywords: *Norm, Situation Calculus, Web Service, Semantic Web, Web Service Composition*

1. Introduction

Web services are described using WSDL, but WSDL only from the syntactic layer of the Web service description. It does not support the rich semantic descriptions. Semantic Web can be regarded as an extension and expansion of the current Web technology. Web services are an increasingly important resource of web, it require semantic to provide wide range semantic support. OWL-S is bridges to connect Web services and semantic Web. Many of the current research on semantic Web services are carried around the OWL-S. However, as the capacity limitations of the rules of OWL, OWL-S model does not include the description of preferences and constraints in users' selection and use of services, which always manifests in the form of rules. The user's preferences and scene constraints can't be understood and processed well by the computer because of the expression limitations. It also means that intelligent system does not start from user requirements during service composition, and does not assemble more subjective service in accordance with users' will. Therefore, the article will extend the semantic of OWL-S through semantic capabilities of the norm. However, norm is non-formal and ambiguous. This paper attempts to use situation calculus to formally describe the norm in order to ensure the correctness of norm and the user correct expression of various scene constraints, and provide more accurate and subjective service in expression of the preferences.

2. Related Researches

Sheila A. McIlraith, Srini Narayanan and others of Stanford University apply the theory of situation calculus and Petri net to study the operational semantics of OWL-S. Firstly, they study the conversion relationship between the atomic service description and situation calculus. Then, they use situation calculus to study the formal semantics of atomic service descriptions in OWL-S. The situation calculus as an intermediate language is converted into the Petri net, and then the Petri net is used to study the implementation reasoning of the operational semantics of semantic Web services [1].

Andrea and Ferrara, who come from DIS-University di Roma use an approach of the process algebra to study the Web service composition, establish mutual mapping between process algebra and Business Process Execution Language for Web Services based on the research of Srini Narayanan and others [2].

SWSI (Semantic Web Services Initiative), mainly by DARPA and the EU (European Union) funded a project to build a combination of Semantic Web and Web services technology integration framework that enables ser-

*This research was supported by the Natural Science Foundation of China (No. 70961007).

vice providers and service requesters in the service discovery retrieval, matching, call, monitor and so on to achieve the greatest possible integration of automation and dynamic.

Yongshang Cheng and Zhijian Wangf, who come from College of Computer and Information Engineering, Hehai University have invented formal model of semantic Web services on the basis of colored Petri nets [3] .They formally describes several basic web service composition models in OWL-S, which is constructed to meet the demand of service, but it lacks of the expression of services semantic information.

Jun Liao, Hao Tan and Jinde Liu, who come from the University of Electronic Science and Technology, describe the Web service composition by using Pi calculus. The paper introduces an example using Pi calculus to describe the Web service composition [4].

From the above research and projects we can see that researchers have made some valuable results in the semantic Web and Web service composition context, and many research problems wait to be solved to provide more powerful semantics services especially in the semantic extensions, but the theoretical system of service composition engineering and implementation technology are still immature because the study of history is not long.

This paper introduces norm theory of Organizational Semiotics to solve the problem of semantic deficiency in Web service composition. In order to achieve a deeper level of semantic extension, it achieves the formal description of norm in semantic expansion by the theory of situation calculus in order to ensure the accuracy of semantic extension.

3. Applications of Norm and Situation Calculus in the Semantic Web Service Composition

Norm is also known as social norms which is the common rules of conduct and standards of every member in a social group or smaller groups, which is in the Oxford Encyclopedia of the explanation.

As rules in expression limitations of the OWL, OWL-S model does not include the users' the description of preferences and constraints in the selection and use of services. It try to introduce norm to provides a new theoretical platform in the semantic Web service composition in order to solve Web services Semantic shortage described by the OWL-S in the semantic Web service composition. This approach considers many the users' initiative and willingness. It joins the human factor in the service composition to provide a more personalized composition services.

Norm determines whether and when an event occurs, and also decided meeting the particular needs should call the areas of service layer of what services. It can describe the activities relationship of the system control. The norm detailed description includes the following sections [5].

Whenever< Conditions set >If< State Sets >Then <Agent>Is <Deontic Operator>To< Action sets >Consequence< Result Sets>

Conditions Sets specified the conditions when an Agent executed an act. Further instructions can be specified by If < State set>, which show that it happened in what state. <Agent> Refers to the responsibility Agent, Agent may be people here, also may be software. < Deontic logic operators> including allowing (Permitted), must (Obliged), prohibition (Prohibited)etc. <Action Sets> determined action sets applied in the case of conditions to met for Agent. <Result sets> is the results after the successful implementation of the specification, which is also the goals of users implement the act.

We can see that norm is different from the relations of causality [6]. Causality is usually only a brief description, if the conditions are met, then certain behavior could take place. Therefore causality is rigid and limited, and there is not available for human decision. On the contrary, norms are a better reflection of how people activities in the business environment. Therefore it is more suited to the description of user needs in the real business. Changes of user needs in the growing environment can be described through the norm. This required model can fully focus on changes factors. It will help to provide dynamically alternative solutions among the service layer and process layer on user needs.

Situation Calculus is a multi-type and first-order logic language, and has some second-order characteristics. It is a formal planning approach and describes a logical basis for dynamic system. For problem solving of the dynamic field and logic programming, it is first proposed in 1963 by McCarthy who is the master of artificial intelligence. Goal-oriented autonomous behavior reasoning can be carried out in a dynamic environment by situation calculus. In the situation calculus, a scenario is a snapshot of the world, and dynamic changes in the world are all the result of actions. Scenarios class is applied to express Situation Calculus, and the scene change is the results of the actions. In ontology, the action is the basic means to change the state of things. Therefore, a possible world active can be viewed as a string of action series. Situation calculus constitutes by the following three elements: *action, situation and flow.*

Action is the basic means to change station of things. All Changes in the state of things are the result of the implementation of the action sequences. A specific predicate *Poss* is used to indicate that an action is executable.

Situation is the dynamic world. All changes in the world are the result of action, which is a string of limited action sequences. Action function is expressed as: action situation → situation. If the action a is implemented under the scenario s, then another situation will be got, and the situation s can be expressed as do (**a**, **s**).

Flow is used to represent the properties of things in the world and the link between them [7]. When an action occurs, the flow will change. Flow ***F***(***x***, ***s***) takes scene s as its last parameter. The initial situation is expressed as s0. The value of the flow is constantly changing from the initial state of the world to the current world state.

In order to characterize the dynamic changing world, situation calculus theory uses the following actions: *Action Precondition Axiom, Successor State Axiom, Effect Axiom* etc. [8].

Action Precondition Axiom: Every action of the field has a corresponding action precondition axioms, describing a prerequisite for action to perform.

$Poss(a,s) \Leftrightarrow f_i$ f_i is a prerequisite sets for the implementation of a.

Successor *State Axiom*: Each flow in field has a corresponding *Successor State Axiom,* which is described the implementation of atomic actions how to affect the flow and state changes.

$$Poss(a,s) \wedge \gamma_F^{+}(x,a,s) \rightarrow F(x,do(a,s))$$

$$Poss(a,s) \wedge \gamma_F^{-}(x,a,s) \rightarrow \neg F(x,do(a,s))$$

$Poss(a,s)$ is a special flow and action a is enforceable under state s.

Formula $\gamma_F^{+}(x,a,s)$ is a positive effect axiom that describes the related collection of actions and conditions to makes the value of flow F true after the execution.

Formula $\gamma_F^{-}(x,a,s)$ is a negative effect axiom that describes the related collection of actions and conditions to makes the value of flow F false after the execution.

In addition to the above several axioms, there are other flows that can be used to describe atomic service or a combination of services as formal semantics of services in situation calculus. Other flows are the following types:

$Kref(a,s)$ expresses that the value of a is known under the scenario s.

$Kwhether(a,s)$ expresses that the true value of a is known under the scenario s.

$Knows(a,s)$ expresses that the function value of a is known under the scenario s.

This essay will introduce users Desirable Axiom and constraint Axiom in order to express the will degree of user requirements in the situation calculus [9-10]. We also attempt to define $\neg Constraint(a,s)$ which action a under the situation s can be implemented as long as the users' conditions are not limited [11].

A set of items compared norm expression with situation calculus to achieve the service semantic expansion and formal description. It is shown as **Table 1**.

Table 1. Mapping norm to situation calculus

Items of norm	Items of the situation calculus
Input	Conditions Input Function $Kref(a,s)$
Conditions set State set	Action Precondition Axiom: $Poss(a,s) \Leftrightarrow f_i$
Obliged	Desirable Axiom $Desirable(a,s)$
Permitted	$\neg Constraint(a,s)$
Prohibited	Constraint Axiom $Constraint(a,s)$
Action set	Action predicate $Poss(a,s)$
Result Set	Effect Axioms and Condition Output Function: $Kwhether(a,s)$ $Kwhether(a,s)$

4. Case Studies

4.1 Example Background

Tourism is an integrated industry which includes food, housing, transportation, travel, shopping and entertainment. A trip involves a number of services and tourism resources, and tourist information is rich in space and time content. Travel is the complex process restricted by artificial and natural factors. How to use Web services by the network to provide "personal travel" service and how to design personalized service process according to the needs of our customers become a problem. The problem has plagued further development of the tourism industry and prevented from increasing the overall quality of tourism services to increase tourists' satisfaction. Therefore the examples will make a combination of the weather check, ticket booking, tourist attractions, reservations and other travel services with personal characters. The service composition flow is shown in **Figure 1**.

The example mainly involves the following web services:

1) $InquireWeather(City, Date)$: Weather inquiry service.

2) $InquireDistance(Start, Destination)$: Check the distance between the two places. *Start* is the place of departure. *Destination* is the sightseeing place.

3) $BuyAirPlaneTicket(City, Date)$: Booking airline tickets.

4) $BuyTrainTicket(City, Date)$: Booking train tickets.

5) $BookScenicSpots(City)$: Attractions reservation.

6) $BookInsurance(TypesOfInsurance)$: Buy insurance.

7) $RentCar(CarModel)$: Service of cars on hire.

8) $BookHotel(City)$: Hotel reservation.

4.2 Norm Analysis

A detailed analysis of the norm is as below **Tables 2-9**.

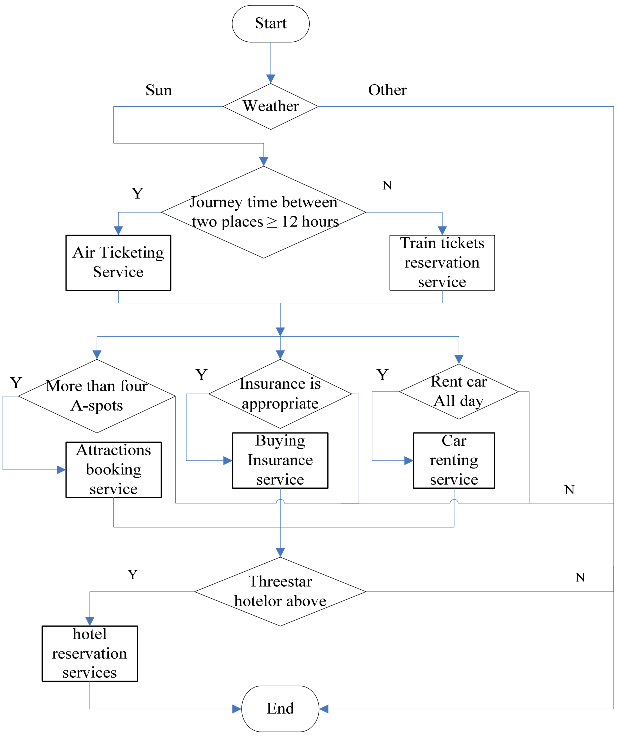

Figure 1. The chart of reservation service composition flow

Table 2. Weather norm

Norm 1	Searching start
In Charge of	User
Information Recognition	City, Date
Trigger Condition	User query the weather of sightseeing site Precondition: Enter valid city and date Then: Weather query service returns query results
Detailed Norm	**if** the input date and destination of users are valid **then** Weather enquiry service **is obliged to** return the query results

Table 3. Distance query norm

Norm 2	Searching start
In Charge of	User
Information Recognition	Start, Destination
Trigger Condition	User queries the distance between the start and destination Precondition: The query results of norm 1 is sun Then: Return the distance between start and destination
Detailed Norm	**Whenever** the weather of the indicated Date is sunny in norm1 **if** the input start and destination of users are valid **then** distance query service **is obliged to** return the distance between two places

Table 4. Airline reservation norm

Norm 3	Booking start
In Charge of	User
Information Recognition	Bank Card ID, User ID
Trigger Condition	Users make flight reservations Precondition: Output is sunny in norm1and the output is greater than or equal 1000 kilometers in norm 2 Then: Users input bank card password ,book ticket
Detailed Norm	**Whenever** Output is sunny in norm1and the output is greater than or equal 1000 kilometers in norm 2 **if** bank card ID and user ID are valid **then user is obliged to** input bank card password to transfer

Table 5. Train ticket reservation norm

Norm 4	Booking start
In Charge of	User
Information Recognition	Bank Card ID ,User ID
Trigger Condition	Users book train tickets Precondition: Output is sunny in norm1and the output is greater than or equal 1000 kilometers in norm 2 Then: Users input bank card password ,book ticket
Detailed Norm	**Whenever** Output is sunny in norm 1and the output is greater than or equal 1000 kilometers in norm 2 **if** bank card ID and user ID are valid **then user permitted** input bank card password to transfer

Table 6. Attractions reservation norm

Norm 5	Booking start
In Charge of Information Recognition	User City
Trigger Condition	Users reserve scenic spot Precondition: City is valid, the grade of attractions are four A-level or above Then: Return the attractions which are qualified
Detailed Norm	**Whenever** norn 3 and norm4 are successful **if** the grade of attractions are four A-level or above **then users permitted** input bank card password and transfer reserve attractions if the grade of attractions are below four A-level then norm 3 or norm 4 **is prohibited** to book

Table 7. Insurance booking norm

Norm 6	Booking start
In Charge of Information Recognition	User City
Trigger Condition	Users reserve insurance Precondition: City is valid, users choose appropriate insurance Then: Users input the bank card password and transfer; Users reserve insurance
Detailed Norm	**Whenever** norm 3 or norm 4 are successful **if** thegrade of attractions are four A-level or above **then** user **is obliged to** input bank card password and transfer in order to reserve attraction

Table 8. Car rental reservation norm

Norm 7	Booking start
In Charge of Information Recognition	User City
Trigger Condition	Users reserve the car which is rental Precondition: City is valid, the rental car is full-time Then: Users input bank card password in order to reserve rental cars
Detailed Norm	Whenever norm 3 or norm 4,norm 5,norm 6 are successful **if** the rental car is full-time **then** users **is obliged to** input bank card password in order to reserve the rental car

Table 9. Hotel reservation norm

Norm 8	Booking start
In Charge of Information Recognition	User Bank Card ID ,User ID
Trigger Condition	Users reserve hotel Precondition: Norm 3 or norm 4 and norm 5 norm 6 norm 7 are successful; The hotel is three-star or above Then: Booking hotel
Detailed Norm	**Whenever** Norm 3 or norm 4 and norm 5 norm 6 norm 7 are successful **if** hotel is above 3A **then** user **is obliged to** input bank card password in order to reserve hotels

4.3 Mapping from Norm to Situation Calculus

1) The Situation Calculus Formally Describes the Atomic Services in Example

a) Weather atomic service: Before the atomic service of Weather is implemented, agency must know the input parameters value which is formalized as the following formula with the situation calculus.

$$Poss(InquireWeather(City, Date), s) \rightarrow$$

$$Kref\,(City, s) \wedge Kref\,(Date, s)$$

$InquireWeather(City, Date)$ is a weather query action, the action on the premise that you need to know the $City$ and $Date$, which can be represented by flow $Kref\,(City, s)$ and $Kref\,(Date, s)$.

Users want to travel as long as the weather was not dust storms and heavy rain. It is formalized as the following formula with the situation calculus.

$$\neg Constraint(Tranvel, s) \equiv$$

$$\neg(Weather(Equal, Storm, s) \wedge$$

$$Weather(Equal, Sadstorm, s));$$

The output of the Weather Service is formalized as the following formula with the situation calculus.

$$Poss(InquireWeather, s) \wedge \gamma(x, InquireWeather, s)$$

$$\rightarrow Knows(Sun/Other, do(InquireWeather, s))$$

b) The atomic service of distance query: Before the atomic service of Distance Query is implemented, agency must know the input parameters value which is formalized as the following formula with the situation calculus.

$$Poss(Distance(Start, Destination), s) \rightarrow$$

$$Kref\,(Start, s) \wedge Kref\,(Destination, s)$$

The output of the Distance Query Service is formalized as the following formula with the situation calculus.

$$Poss(Distance, s) \wedge \gamma(x, Distance, s) \rightarrow$$

$$Knows(\ Figure, do(Distance, s))$$

c) Ticket booking atomic services: Before the atomic service of Airline Reservation is implemented, agency must know the input parameters value which is formalized as the following formula with the situation calculus.

$$Poss(BuyAirplaneTicket(City, Date), s) \rightarrow$$

$$Kref\,(City, s) \wedge Kref\,(Date, s)$$

If user wants to buy air tickets must be that the distance between two places is greater than or equal to 1,000 kilometers. It is formalized as the following formula with the situation calculus.

$$Desirable(BuyAirplaneTicket(City, Date), s) \equiv$$

$$Distance((Start, Destinatin), Longer1000);$$

The effectiveness and output parameters of atomic service buy ticket are formalized as the following Positive Effect Axiom and Negative Effect Axiom.

$$\gamma_F^+ : a = BuyAirplaneTicket(City, Date) \wedge$$
$$InstockAirplaneTicket(City, Date)$$
$$Poss(a, s) \wedge a = BuyAirplaneTicket(City, Date) \wedge$$
$$InstockAirplaneTicket(City, Date) \rightarrow$$
$$own(AirplaneTicket, do(a, s))$$

2) The Situation Calculus Formally describes the Composition Services in Example

The Example involve several combination services: *BookTicket* is composed of the Distance Query Service, ticket booking service and ticket reservation service, which implement are controlled mainly by *if* control operator; *Journey* is composed of the Attractions Reservation Service; Insurance Booking Service and Car Rental Reservation Service, which implement are controlled mainly by *Anyordered* control operator; The composition service *Tranvel* is composed of the composition service *BookTicket* ; composition service *Journey* and Hotel Reservation Service, which implement are controlled mainly by *Condition* control operator.

a) Composition service *BookTicket* : Distance Query Service, ticket booking service and ticket reservation service is composed *BookTicket* , which implement are controlled mainly by *if* control operator.

The prerequisite was formalized as the following formula in the situation calculus:

$$(Knows(Over1000km,$$
$$do(InquireDis\tan ce(Start, Destination), s)) \wedge$$
$$Poss(BuyAirplaneTicket(City, Date), s) \rightarrow$$
$$Knows(Sun, do(InquireWeather(City, Date), s))) \vee$$
$$(Knows(Below1000km,$$
$$do(InquireDis\tan ce(Start, Destination), s)) \wedge$$
$$Poss(BuyTrainTicket(City, Date), s) \rightarrow$$
$$Knows(Sun, do(InquireWeather(City, Date), s)));$$

Before the composition service of *BookTicket* is implemented, agency must know the input parameters value which is formalized as the following formula with the situation calculus.

$$(Knows(Over1000km,$$
$$do(InquireDis\tan ce(Start, Destination), s)) \wedge$$
$$Poss(BuyAirplaneTicket(City, Date), s) \rightarrow$$
$$Kref(City, s) \wedge Kref(Date, s)) \vee$$
$$(Knows(Below1000km,$$
$$do(InquireDis\tan ce(Start, Destination), s)) \wedge$$
$$Poss(BuyTrainTicket(City, Date), s) \rightarrow$$
$$Kref(City, s) \wedge Kref(Date, s))$$

Based on the above two formulas, we can get the Action Precondition Axiom.

$$(Knows(Over1000km,$$
$$do(InquireDis\tan ce(Start, Destination), s)) \wedge$$
$$Poss(BuyAirplaneTicket(City, Date), s) \rightarrow$$
$$Knows(Sun, do(InquireWeather(City, Date), s)) \wedge$$
$$Kref(City, s) \wedge Kref(Date, s)) \vee$$
$$(Knows(Below1000km,$$
$$do(InquireDis\tan ce(Start, Destination), s)) \wedge$$
$$Poss(BuyTrainTicket(City, Date), s) \rightarrow$$
$$Knows(Sun, do(InquireWeather(City, Date), s)) \wedge$$
$$Kref(City, s) \wedge Kref(Date, s))$$

The effectiveness and output parameters of composition service are formalized as the following Positive Effect Axiom and Negative Effect Axiom.

$$\gamma_F^+ : a_1 = BuyAirplaneTicket(City, Date) \wedge$$
$$InstockAirplaneTicket(City, Date)$$
$$a_2 = BuyTrainTicket(City, Date) \wedge$$
$$InstockTrainTicket(City, Date)$$
$$(Knows(Over1000km,$$
$$do(InquireDistance(Start, Destination), s)) \wedge$$
$$Poss(a_1, s) \wedge \gamma_F^+(x, a_1, s) \rightarrow$$
$$Own(AirplaneTicket, do(a_1, s))) \vee$$
$$(Knows(Below1000km,$$
$$do(InquireDistance(Start, Destination), s)) \wedge$$
$$Poss(a_2, s) \wedge \gamma_F^+(x, a_2, s) \rightarrow$$
$$Own(TrainTicket, do(a_2, s)))$$
$$\gamma_F^- : a_1 = \neg BuyAirplaneTicket(City, Date) \vee$$
$$NoInstockAirplaneTicket(City, Date)$$
$$a_2 = \neg BuyTrainTicket(City, Date) \vee$$
$$NoInstockTrainTicket(City, Date)$$
$$(Knows(Over1000km,$$
$$do(InquireDis\tan ce(Start, Destination), s)) \wedge$$
$$Poss(a_1, s) \wedge \gamma_F^-(x, a_1, s) \rightarrow$$
$$\neg Own(AirplaneTicket, do(a_1, s))) \vee$$
$$(Knows(Below1000km,$$
$$do(InquireDis\tan ce(Start, Destination), s)) \wedge$$
$$Poss(a_2, s) \wedge \gamma_F^-(x, a_2, s) \rightarrow$$
$$\neg Own(TrainTicket, do(a_2, s)))$$

The output of the Composition Service *BookTicket* was formalized as the following formula with the situation calculus.

$$(Knows(Over1000km,$$

$do(InquireDis\tan ce(Start, Destination), s)) \wedge$

$Poss(a_1, s) \wedge \gamma(x, a_1, s) \rightarrow$

$Knows(AirplaneTicketMessage, do(a_1, s))) \vee$

$(Knows(Below1000km,$

$do(InquireDis\tan ce(Start, Destination), s)) \wedge$

$Poss(a_2, s) \wedge \gamma(x, a_2, s) \rightarrow$

$Knows(TrainTicketMessage, do(a_2, s)))$

b) Composition service $Journey$: It is composed with Attractions Reservation Service, Insurance Booking Service and Car Rental Reservation Service. Implement of the service is controlled by $Anyordered$ control operator.

The implementation order of atomic services which are involved in the composition service is not required, so the composition service can be controlled implemented $Anyordered$ control operator.

The prerequisite was formalized as the following formula with the situation calculus.

$Poss(BookScenicSpots(City), s) \wedge$

$Poss(BookInsurance(TypesOfInsurance),$

$do(BookScenicSpots, s)) \wedge$

$Poss(\text{Re}\, ntCar(CarModel),$

$do(BookInsurance, do(BookScenicSpots, s))) \rightarrow$

$Knows(GreaterFourA,\ do(InquireScenicSpots, s)) \wedge$

$Knows(TypesOfInsurance,$

$do(InquireInsurance(TypesOfInsurance, s))) \wedge$

$Knows(CarModel, do(Inquire\,\text{Re}\, ntCar, s)) \vee$

$Poss(\text{Re}\, ntCar(CarModel), s) \wedge$

$Poss(BookInsurance(TypesOfInsurance),$

$do(\text{Re}\, ntCar, s)) \wedge\ Poss(BookScenicSpots(City),$

$do(BookInsurance, do(\text{Re}\, ntCar, s))) \rightarrow$

$Knows(CarModel, do(Inquire\,\text{Re}\, ntCar, s)) \wedge$

$Knows(TypesOfInsurance,$

$do(InquireInsurance(TypesOfInsurance, s))) \wedge$

$Knows(GreaterFourA, do(InquireScenicSpots, s))$

Before the composition service of $Journey$ is implemented, the agency must know the input parameters value which is formalized as the following formula with the situation calculus.

$Poss(BookScenicSpots(City), s) \wedge$

$Poss(BookInsurance(TypesOfInsurance),$

$do(BookScenicSpots, s)) \wedge$

$Poss(\text{Re}\, ntCar(CarModel), do(BookInsurance,$

$do(BookScenicSpots, s))) \rightarrow$

$Kref\,(City, s) \wedge Kref\,(TypesOfInsurance, s) \wedge$

$Kref\,(CarModel, s) \vee\ Poss(\text{Re}\, ntCar(CarModel), s) \wedge$

$Poss(BookInsurance(TypesOfInsurance),$

$do(\text{Re}\, ntCar, s)) \wedge\ Poss(BookScenicSpots(City),$

$do(BookInsurance, do(\text{Re}\, ntCar, s))) \rightarrow$

$Kref\,(CarModel, s) \wedge Kref\,(TypesOfInsurance, s) \wedge$

$Kref\,(City, s)$

The above two formulas, we can get the Action Precondition Axiom.

$Poss(BookScenicSpots(City), s) \wedge$

$Poss(BookInsurance(TypesOfInsurance),$

$do(BookScenicSpots, s)) \wedge$

$Poss(\text{Re}\, ntCar(CarModel), do(BookInsurance,$

$do(BookScenicSpots, s))) \rightarrow$

$Knows(GreaterFourA, do(InquireScenicSpots, s)) \wedge$

$Knows(TypesOfInsurance,$

$do(InquireInsurance(TypesOfInsurance, s))) \wedge$

$Knows(CarModel, do(Inquire\,\text{Re}\, ntCar, s)) \wedge$

$Kref\,(City, s) \wedge Kref\,(TypesOfInsurance, s) \wedge$

$Kref\,(CarModel, s) \vee\ Poss(\text{Re}\, ntCar(CarModel), s) \wedge$

$Poss(BookInsurance(TypesOfInsurance),$

$do(\text{Re}\, ntCar, s)) \wedge\ Poss(BookScenicSpots(City),$

$do(BookInsurance, do(\text{Re}\, ntCar, s))) \rightarrow$

$Knows(CarModel, do(Inquire\,\text{Re}\, ntCar, s)) \wedge$

$Knows(TypesOfInsurance,$

$do(InquireInsurance(TypesOfInsurance, s))) \wedge$

$Knows(GreaterFourA, do(InquireScenicSpots, s)) \wedge$

$Kref\,(CarModel, s) \wedge$

$Kref\,(TypesOfInsurance, s) \wedge Kref\,(City, s)$

In this combination of services, users require the attractions level to be 4A or above and the rental car is full-time on behalf of driving, the above can be mapped to the Desirable Axiom with the situation calculus.

$Desirable(BookScenicSpots(City),\ s) \equiv$

$\text{Grade}(GreaterFourA, s)$

$Desirable(\text{Re}\, ntCar(CarModel),\ s) \equiv$

$\text{Re}\, ntCarModel(WholeDay, s)$

The composition service $Tranvel$ is composed of the composition service $BookTicket$. Composition service $Journey$ and Hotel Reservation Service are controlled by $Condition$ control operator.

Here the $BookTicket$, $Journey$ and Hotel Reservation service are seen as the three actions: $a_1 = BookTicket$, $a_2 = Journey$, $a_3 = BookHotel$.

a_1 and a_2 are the implementation precondition of

a_3, the implementation of composition service *Travel* can be controlled by the *Condition* control operator.

The prerequisite of composition service *Travel* was formalized as the following formula in the situation calculus.

$$Knows(Success, do(BookTicket, s)) \wedge$$
$$Knows(Success, do(Journey, s)) \wedge Poss(a_3, s) \to$$
$$Knows(Greater3Star, do(InquireHotel, s))$$

Before the composition service of *Travel* is implemented, the agency must know the input parameters value which is formalized as the following formula with the situation calculus.

$$Knows(Success, do(BookTicket, s)) \wedge$$
$$Knows(Success, do(Journey, s)) \wedge Poss(a_3, s) \to$$
$$Kref(City, s)$$

The above two formulas, we can get the Action Precondition Axiom.

$$Knows(Success, do(BookTicket, s)) \wedge$$
$$Knows(Success, do(Journey, s)) \wedge Poss(a_3, s) \to$$
$$Kref(City, s) \wedge Knows(Greater3Star, do(InquireHotel, s))$$

The effectiveness and output parameters of composition service are formalized as the following Positive Effect Axiom and Negative Effect Axiom.

$$\gamma_F^+ : a_3 = BookHotel(City) \wedge Instock(Room)$$
$$Knows(Success, do(BookTicket, s)) \wedge$$
$$Knows(Success, do(Journey, s)) \wedge$$
$$Poss(a_3, s) \wedge \gamma_F^+(x, a_3, s) \to$$
$$Own(TanvelMessage, do(a_3, s))$$

The output of the Composition Service was formalized as the following formula with the situation calculus.

$$Knows(Success, do(BookTicket, s)) \wedge$$
$$Knows(Success, do(Journey, s)) \wedge$$
$$Poss(a_3, s) \wedge \gamma(x, a_3, s) \to$$
$$Knows(TanvelMessage, do(a_3, s))$$

5. Summary

This paper introduces the norm to strengthen the semantics from user needs perspective based on the defects of lack of semantics described in OWL-S semantic service of the semantic web service composition. A new way is provided to increase semantic. However, the norm is non-formal and ambiguous, the situation calculus is applied to realize formal norm in order to ensure the accuracy of semantic Web service composition.

REFERENCES

[1] S. N. Sheila and S. Mcilraith, "Analysis and Simulation of Web Services," *Computer Networks*, Vol. 42, May 2003, pp. 675-693.

[2] A. Ferrara, "Web Services:A Process Algebra Approach," *Proceedings of the International Conference on Service Oriented Computing*, New York, 2004, pp.1-18

[3] Y. S. Cheng and Z. J. Wang, "Research on Formal Description and Verification of Semantic Service Composition," *Computer Science*, Vol. 35, No. 4, April 2008, pp. 40-43.

[4] J. Liao, H. Tan and J. D. Liu, "Describing and Verifying Web Service Using Pi-Calculus," *Chinese Journal of Computers*, Vol. 28, No. 4, April 2005, pp. 635-643.

[5] R. Stamper and K. Liu, "Organisational Dynamics, Social Norms and Information Norms," *Proceedings of Hawaii International Conference on System Sciences*, Hawaii, 1994, pp. (IV)645-654.

[6] H. Q. Yang, S. Z. Li, J. P. Zhao and W. Pu, "Research on Architecture Design and Requirement Modeling Based on Semiotics of Adaptive Healthcare Information System," *Journal of Computer Engineering and Applications*, Vol. 45, No. 9, September 2009, pp. 244-248.

[7] G. Wang, "Research and Implement on the Behaviors of Multiple Intelligent Virtual Human Based on the Situation Calculus," JiangSu University, Zhenjiang, 2006

[8] D. M. Xu and H. Q. Wang, "Muti-Agent Collabo-Ration for B2B Workflow Monitoring," *Knowledge-Based System*, Vol. 15, No. 8, November 2002, pp. 485-491.

[9] S. Liang, "Research on Technology of Automatic Services Composition Based on Semantic Web," Chinese Academy of Sciences, Beijing, 2003.

[10] S. McIlraith and T. C. Son, "Adapting Golog for Programming the Semantic Web," *International Conference on Knowledge Representation and Reasoning*, Toulouse, 2002, pp. 482-493.

[11] Z. S. Gao, "A Multi-Agent Based Approach for Automated Semantic Web Services Composition," XiHua University, Chengdu, 2006.