ISSN : 0974 - 7435 Volume 10 Issue 8

2014

BioTechnology
An Indian Journal

FULL PAPER
BTAIJ, 10(8), 2014 [2785-2791]

The formalization description and validation studies about semantic web services based on pi calculus combination

Qinglin Xu
Xianyang Normal University, Xianyang 712000, Shaanxi, (CHINA)

ABSTRACT

In today's well-developed information society, internet has become a necessity in our daily life. But many people may not know the wisdom and secret behind the internet. In order to further understand the internet, this thesis will first introduces the nature of the Web services as well as the practical application value, and talks about the relatively insufficient problem of the dynamic system in Web service composition causes by some standardized defects. To improve the description and validation problem above, this thesis studies the concept of Web service definition, syntax, rules and operational semantics, and compared it with Pi, finds out the corresponding relation of elements between them. Then, modeling the BPEL4WS service combination through Pi calculus system and test it through examples.

KEYWORDS

Pi calculus; Web services; Combination of formalization; BPEL4WS.

INTRODUCTION

As now we living in an information age, internet has become of vital important part in people's daily life. It offers varies kinds of services for us and provides us convenience and knowledge. Among all the services, the Web counts for most proportion. The system of Web is consists of large number of text links, and each one of them can be seen as resource. Users can access to resources through the links and the Web will send the message to users through transport protocol. However, what the users wonder about is how the Web operates. The description of Web services composition has two kinds of specifications, and through the combination of the two specifications, the implementation of service portfolio can be done. But the flaws in these two specifications leads to the relatively insufficient problem of the dynamic system in Web service composition. To improve the above description and validation issues, people often use CSS, PI calculus, Petri net and other way to authenticate the service portfolio. This thesis will gives a description about the Pi calculus web service composition model, and carries on the research and analysis.

THE MODELING AND SOLVING OF THE WEB SERVICE COMPOSITION FORMALIZATION

The Pi calculus we talk about in this thesis can define the concept, syntax rules and operational semantics of Web service composition formalization, and figures out the service order through the syntax then through the definition of operational semantics, this thesis describe the message interaction in combinations semantics.

The Survey of Pi Calculus

In Pi calculus, the mane is the basic elements, and all the channels and variables without distinction can be distinguished by their names. Since the description of its service portfolio is a dynamic structure, that is to say the channel can always migrate and change, so it is more suitable for describing the systems with fast changing structures.

If N is one of the name set in Pi calculus, the lowercase letter in it means the channel, value and variables while the capital letter means the calculate process and the expression, like:

$$P ::= 0 \left| \bar{x}y.P \right| x(y).P \left| \tau.P \right| (vx)P \left| [x=y]P \right|$$
$$[x \neq y]P \left| P \right| Q \left| P+Q \right| !P \left| A(y_1,...,y_n) \right.$$

The following paragraphs detail the concept, syntax and operational semantics that the Pi calculus provides in the Web service portfolio formalization.

The concept of Web service composition formalization

This thesis will set the service as a four tuples, that is $S = \langle I, O, V, B \rangle$. S as name; I as the input set for the channels; O as the output set from the channels; V as the internal variables set; B as the set of action in service. Apart from these, since the description of Web service composition formalization have to use varies kinds of rules that involving all kinds of proposition, so if P_S means the set of related propositions of S, the $[P_S : P]$ means any reference proposition in the set. And for the other references, I_S, O_S, V_S, and B_S are represent the set of input channel, output channel, variables and behavior. $[I_S : i]$, $[O_S : o]$, $[V_S : v]$ and $[B_S : b]$ represent any reference of one element.

Introduce the syntax of Web service composition formalization

If B_S is S 's behavior set, and $\forall b \in B_S$, then

1. When $[B_S:b]$ represent the output action, $[B_S:b]::=\overline{x}(y)$, y is the name of the input, x is the message and channel of input, they are private variables of S.
2. When $[B_S:b]$ represent the input action, $[B_S:b]::=x(y)$, y is the name of the output, x is the message and channel of output, they are private variables of S.
3. When $[B_S:b]$ represent the internal action, $[B_S:b]=\tau$, represent the internal action of S that cannot be seen by outsiders.
4. When $[B_S:b]$ represent the invoke action of S, $[B_S:b]=S(y_1,...,y_n)$ and $S(x_1,...,x_n)\overset{def}{=}S$, $y_1,...,y_n$ are the actual argument for the invoke action of service, $x_1,...,x_n$ as the formal parameter that defined S.

Since the description of S's structure is based on the operator that can control the order and structure of action, so the following part will detail the difference between the service structure of Web and the calculus structure of Pi. The definition of Web service structure is:

$$
\begin{aligned}
S ::= 0 &\,\|\, \overline{[O_S:o]}([V_S:v]).S' \,\|\, \overline{[O_S:o]}(C_S:c).S' \\
&\|\, \overline{[I_S:i]}([V_S:v]).S' \,\|\, \overline{[I_S:i]}(C_S:c).S' \,|\, \tau_S.S' \\
&\|\, [P_S:p]S' \,|\, (9)[P_S:p]S' \,|\, S_1+S_2 \,|\, S_1 \,|\, S_2 \\
&|\, !S \,|\, ![P_S:p]S \,|\, S(y_1,...,y_n) \\
&S(x_1,...,x_n) \overset{def}{=} S
\end{aligned}
$$

$[V_S:v]$, $[C_S:c]$ are the variables and message channel of S, since the name is limited in S, in the expression prefix with S, the input and output object are also limited. τ is the message channel of S and creates a group with it through interaction.

$\|[P_S:p]S'$ is when $[P_S:p]$ satisfied the result will be S', if not the result will be 0, $(9)[P_S:p]S'$ is similar to this condition. $![P_S:p]S$ is when the boundary condition $[P_S:p]$ satisfied, the service will consistently operate S until it fails. $![P_S:p]S$ can also express as $(\|[P_S:p]S'+(9)[P_S:p]0)$

The rules of web service composition formalization

The rule of Web service composition formalization is, if S, S_1, S_2 and S_3 are all services, when S_1, S_2 present different kinds of services, variable and message channels, $S_1\equiv S_2$. Then

$$
\begin{aligned}
&S_1|S_1 \equiv S_2|S_1 \\
&(S_1|S_2)|S_3 \equiv S_1|(S_2|S_3) \qquad S|0\equiv S \\
&S_1+S_2\equiv S_2+S_1 \\
&(S_1+S_2)+S_3\equiv S_1+(S_2+S_3) \qquad S+0\equiv S \\
&S(y)\equiv S\{y/x\}\ if S(y)::=S
\end{aligned}
$$

The operation semantic of the web service composition

The operation semantic is defined after the interaction and migrates of Web service group, just the same as in Pi:

$$AS-STRUCT:\frac{S_1 \equiv S_1^{'} \wedge S_1 \xrightarrow{\alpha} S_2 \wedge S_2 \equiv S_2^{'}}{S_1^{'} \xrightarrow{\alpha} S_2^{'}}$$

$$AS-ACT:\frac{-}{\tau.S \xrightarrow{\tau} S}$$

$$AS-NPUT:\frac{-}{[I_S:i]([C_S:c]).S \xrightarrow{[I_S:i](w)} S\{w/[C_S:c]\}}$$

$$\frac{-}{[I_S:i]([V_S:v]).S \xrightarrow{[I_S:i](w)} S\{w/[V_S:v]\}}$$

$$AS-OUTPUT:\frac{-}{\overline{[O_S:o]}([C_S:c]).S \xrightarrow{\overline{[O_S:o]}([C_S:c])} S}$$

$$\frac{-}{\overline{[O_S:o]}([V_S:v]).S \xrightarrow{\overline{[O_S:o]}([V_S:v])} S}$$

$$AS-CHOICE:\frac{S_1 \xrightarrow{\alpha} S_1^{'}}{S_1+S_2 \xrightarrow{\alpha} S_1^{'}}$$

$$AS-PAR:\frac{S_1 \xrightarrow{\alpha} S_1^{'} \wedge bn(\alpha)\cap(C_{S_2}\cup V_{S_2})=\approx}{S_1|S_2 \xrightarrow{\alpha} S_1^{'}|S_2}$$

$$AS-COM:\frac{S_1 \xrightarrow{\overline{[O_{S_1}:x]}([V_{S_1}:y])} S_1^{'} \quad S_2 \xrightarrow{[I_{S_2}:x]([V_{S_2}:z])} S_2^{'}}{S_1|S_2 \xrightarrow{\tau} S_1^{'}|S_2^{'}\{y/z\}}$$

$$\frac{S_1 \xrightarrow{\overline{[O_{S_1}:x]}([C_{S_1}:y])} S_1^{'} \quad S_2 \xrightarrow{[I_{S_2}:x]([C_{S_2}:z])} S_2^{'}}{S_1|S_2 \xrightarrow{\tau} S_1^{'}|S_2^{'}\{y/z\}}$$

The following semantic is the modify vision by using Web service composition formalization:

$$AS-MATCH:\frac{S_1 \xrightarrow{\alpha} S_1^{'}}{[P_{S_1}:P]S_1 \xrightarrow{\alpha} S_1^{'}}$$

$$AS-REP:\frac{S \xrightarrow{\alpha} S^{'}}{![P_S:P]S \xrightarrow{\alpha} S^{'}|![P_S:P]S}$$

In the above semantic, $AS-MATCH$ represents if S turn into $S^{'}$ after operate action α, then, if $[P_S:P]$ is satisfied, $[P_S:P]!S$ will turn into $S^{'}$. $AS-REP$ represents if S turn into $S^{'}$ after operate action α, then, if $[P_S:P]$ is satisfied, $[P_S:P]!S$ will turn into $S!|[P_S:P]!S$, that is to say this service is used to represent loop execution.

The corresponding relation of Pi calculus' and Web's elements

TABLE1 shows the corresponding relation of Pi calculus and Web's element

TABLE 1 : The corresponding relation of element in Pi calculus and Web

Pi calculus	Web service
Message	Message
Action	Operation
Sort	PortType
Port	Port
Interaction	Binding
Process	Service

The following part will discuss the interaction between Web service and Pi calculus through flow chart. In order to use Pi calculus to represent Web service, first we should let the Web service represent the process of Pi, so the interaction relationship between them is as showed in Figure 1.

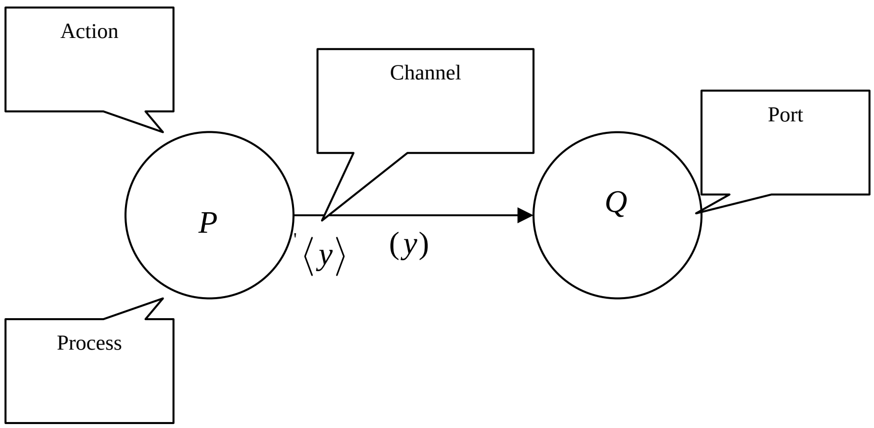

Figure 1 : The corresponding relation between Pi calculus and Web

Modeling BPEL4WS through Pi calculus

BPEL4WS is used to describe the language that is used during the interaction and dealing between commercial and business, the formal is XML. The process is mainly divided into executable and abstract. Executable process can operate on BPEL4WS's engine while the abstract one cannot, it emphasis more on description of the business agreement, rather than the corresponding process service for partners. So the next part will mainly introduces the principle and application of the abstract process.

The service provider involved in the executable process is defined as cooperator. During the period of business, BPEL will send message to involved cooperators and build up links to show their interactive relationship. BPEL clarified them as partnerLinkType, partnerLink and partner. Their relationship is showed as Figure 2.

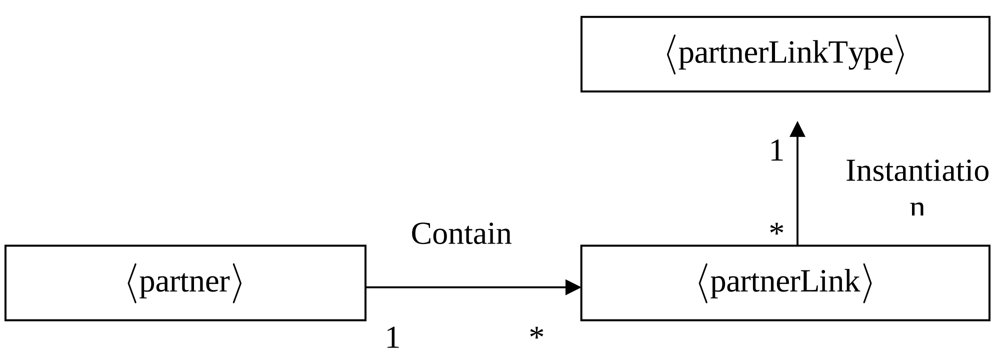

Figure 2 : Relationship between partnerLinkType partnerLink and partner

In order to describe a service portfolio of business or business processes, BPEL4WS has defines varies kinds of branch action and structure action, named the basic activities and structure activity, and provide the events and exception handler to solve the abnormal and events.

The basic activities include : receive, reply, assign, wait, empty, invoke, throw and compensate. These are the basic ones and don't contain in other activities. $\langle \text{receive} \rangle$ represents the process message send by client; $\langle \text{invoke} \rangle$ represents other services; $\langle \text{reply} \rangle$ represents synchronous operation; $\langle \text{assign} \rangle$ represents variables; $\langle \text{throw} \rangle$ represent the appearance of indicate mistake;

$\langle \text{wait} \rangle$ represents wait; $\langle \text{terminate} \rangle$ represent done. And they share some common points and elements. The attributes include: name, execution and the reminding of mistake caused by the execution; elements: source and target used to define the relationship of the activities of sync.

The structure activity is similar the control statements in traditional programming language. There are in totally 5 structure activity, $sequence$, if, $while$, $flow$ and $pick$. Like Figure 3 shows.

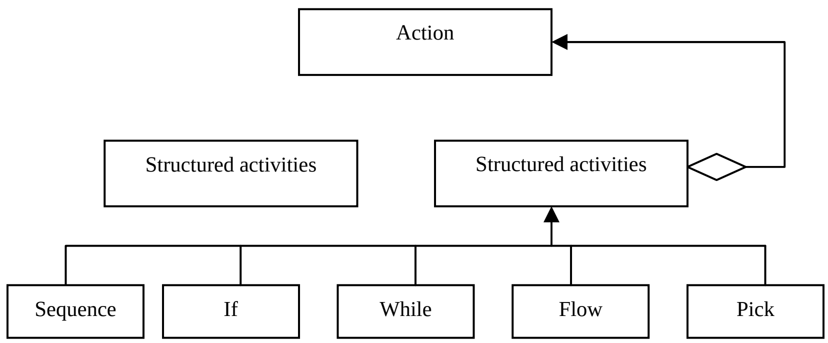

Figure 3 : The structure image of BPEL4WS

Among them, the sequence represents the order relation and order activity between program statements; if represents select activities through determines condition; flow represents parallel activities, that is all activities are completed when parallel activities in order to be executed; pick represents the incident's selective activities, that is when an event is completed, the associated activities will begin, when all these completed the activity can be called done.

- Before the activity and processor defined by BPEL4WS are going to be formalized through Pi calculus, we must firstly, find out the correspondence relationship between their concept.
  The process that BPEL4WS describe is correspondent to one process in Pi.
- The standardized action in BPEL4WS is correspondent to one process in Pi.
- The handler defined in BPEL4WS correspondent to one process in Pi.
- The interaction in Web service is described through communicates between processes in Pi.
- The variables in BPEL4WS are correspondent to the information exchanged through the communication between processes in Pi.
- The interaction access point of Web service can described by partner Link, partner Type and operation in BPEL4WS and is correspondent to the channels in Pi process.

In order to further simplify the description of BPEL4WS by Pi calculus, this thesis decides the four groups, $\text{ChannelName=(partnerLink,poerType,operation,variableName)}$ should correspond to the unique channel name. Due to space reason, here are only some example of the molding of basic action.

The Receive and Reply action can provide appropriate services for business partners in the process. First accept the request through Receive and then give the result back to the partner through Reply. Below are the syntax code for Receive and Reply action.

$$
\begin{array}{l}
\langle receivepartnerL\,\mathrm{in}\,k = "NCName" \rangle \\
\quad portType = "QName"? \qquad operation = "NCNme" \\
\quad \mathrm{var}\,iable = "BPELVariableName"? \\
\quad createIns\,\mathrm{tan}\,ce = "yes|no"? \\
\quad s\,\mathrm{tan}\,dard - attributes > \\
\quad s\,\mathrm{tan}\,dard - elements \\
\quad < correlations > ? \\
\qquad < correlation\ set = "NCName" initiate = "yes|join|no"? / > + \\
\quad < /correlations > \\
\quad < /receive > \\
\quad < replypartnerLink = "NCName" \\
\qquad portType = "QName"? operation = "NCName" \\
\qquad \mathrm{var}\,iable = "BPELVariableName"? \quad faultName = "QName"? \\
\qquad s\,\mathrm{tan}\,dard - attributes > \\
\qquad s\,\mathrm{tan}\,dard - elements \\
\qquad < correlations > ? \\
\qquad\quad < correlation\ set = "NCName" initiate = "yes|join|no"? / > + \\
\qquad < /correlations > \\
\quad < /reply >
\end{array}
$$

CONCLUSION

In this thesis, the author first introduces the nature of the Web services as well as the practical application value, and talks about the relatively insufficient problem of the dynamic system in Web service composition causes by some standardized defects. To improve the description and validation problem above, this thesis studies the concept of Web service definition, syntax, rules and operational semantics, and compared it with Pi, finds out the corresponding relation of elements between them. Then, modeling the BPEL4WS service combination through Pi calculus system and test it through examples.

REFERENCES

[1] Yue Kun, Wang Xiaoling, Zhou Aoying; The Core Support Technology of Web Service: Research Overview [J] Journal of Software, **15(3)**, 428-442 **(2004)**.

[2] Liao Jun, Tan Hao, Liu Jingde; The description and Verify of Web Service based on Pi calculus. [J] Journal of Computers, **28(4)**, 635-643 **(2005)**.

[3] Yuan Ming, Huang Zhiqiu, Cao Zining; An expansion calculus of Pi and the Transactional Equivalence Relation Research [J] Computer Research and Development, **47(3)**, 541-548 **(2010)**.

[4] Gu Xiwu, Lu Zhengding; Choreography Description Language of Web Service, Formal Modeling Framework ofWS-CDL[J] Computer Science, **34(9)**, 5-11 **(2007)**.

[5] J.Camara, C.Canal, J.Cu Bo; Issues in the formalization of Web service or chestrations[C]//Proc of the 2^{nd} International Work shop on Coordination and Application Techniques for Software Entities, **(2005)**.

[6] I.A.Brog, C.Canny, VPimentel; Form aliz ingWeb service choreographies [J]. Electronic Notes in TheoreticalC omputer Sc-ience, **105**, 73-94 **(2004)**.