Formal Modeling and Verification for Web Service Composition

Baojun Tian
College of Information Engineering,Inner Mongolia University of technology,Huhhot,China
ngdtbj@126.com

Yanlin Gu
Vocational College,Inner Mongolia University of Finance and Economics,Huhhot,China
guyanlin@126.com

***Abstract*—Web service composition is an important reuse way of service resources, through which large scale and complex applications based on Internet can get the newer and more reliable resources among the distributed nodes. An important challenge for web services composition is how to ensure the correctness and reliability of the composition process. CPN(Colored Petri Net)as a formal modeling and verification method not only provides formal semantics and a number of analysis techniques, but also is independent of the concrete flow description language(e.g. BPEL, WSCDL).This paper presents an approach of formal modeling and verifying for web service composition and uses an instance to illustrate it.**

***Index Terms*—service composition, SOC, CPN, formal**

I. INTRODUCTION

With the rapid development of Internet technique,service oriented compute(SOC) has been one of main contents of software engineering research field.Web service is a distributed, self- contained, self-discribing module.Web service provides a way for developing systems based on components that is independent on any platform.But the functions provided by a single web service is limited,and cannot meet the needs of users,web service composition is becoming an important issue.

Web service composition has the ability to obtain exiting services and combine them to form new value-added services.A variety of methods have been proposed for modeling and verfying web service composition,and they are parted into three different categories[7]:

(1)Web service composition based on Workflow

This method includes WSFL,WSCDL,WSBPEL and etc.It provides a static composition of pre-defined process,in which web services involved have usually been known. This method is lack of flexibility and automation.And it currently lacks the ability to test or simulate the correctness of workflow sequences.For example web service description language such as WSBPEL,WSCDL only gives syntax expression of process definition based on XML[12][13].

(2)Web service composition based on artificial intelligence.

This method takes web service as artificial intelligence action.It uses input,output,precondition and effect(IOPE) to describe functions and behaviour of web service and when composing maps them to formal description.Web service composition satisfying requirement can be obtained by using formal axiom and inference.It is parted into three different categories based composition algorithm: planning approach, calculus approach and rules approach[10][[11].

(3)Web service composition based on formalization

This method utilizes maths and formal tools to describe,analyze and verify web service composition.It mainly includes state calculus(e.g. Petri nets) and process algebra(e.g. Pi calculus,CCS)[9].

The process of web service composition is usually an error-prone pratice.If the process definition without being verified becomes operational,it often results in runtime errors, which causes software maintenance costly.To solve this problem,many researchers propose a lot of solutions.For example,using SPIN modeling tools verify web service composition whose language is WSFL.Using PAC and CWB-IN tools model and verify web service composition whose language is BPEL4WS based on BPE calculus. However these solutions are usually dependent on the concrete flow description language[5][15].

CPN as a formal method and an extended version of Petri nets is a graphical formal method and has precise mathematic semantics and automated verification tools, which is suitable for modeling communication, concurrency and synchronization systems.Moreover it is independent of any concrete flow description language and can be more specific description of web composition by data flow and control flow[2][3][14].

This paper proposes a CPN-based method that can detect the bugs at design stage and assure the correctness of service composition by static structural analysis and dynamic behavioural analysis.

Project number: science research project of Inner Mongolia University of Technology(ZD201317)

© 2013 ACADEMY PUBLISHER
doi:10.4304/jsw.8.11.2733-2737

II. Related Theory Knowledge

A. Formal Definition of CPN

A CPN is a tuple CPN=(∑,P,T,A,N,C,G,E,I) satisfying the requirements below[1][4]:

(i) ∑is a finite set of non-empty types, called colour sets.

(ii) P is a finite set of places.

(iii) T is a finite set of transitions.

(iv) A is a finite set of arcs such that:

- P∩T=P∩A=T∩A=Φ.

(v) N is a node function. It is defined from A into (P x T)∪(T x P).

(vi) C is a colour function. It is defined from P into ∑.

(vii) G is a guard function. It is defined from T into expression such that:

- ∀ t∈T:[Type(G(t))=$\mathbb{B}$∧Type(Var(G(t)))⊆∑].

(viii) E is an arc expression function.It is defined from A into expressions such that:

- ∀ a∈A: [Type(E(a))=C(p(a))$_{MS}$∧Type(Var (E(a)))⊆∑] where p(a) is the place of N(a).

(ix) I is an initialization function.It is defined from P into closed expressions such that:

- ∀ p∈P: [Type(I(p))=C(p)$_{MS}$].

B. Formal Definition of WS_CPN

To verify web service composition,we must map the CPN element to web service element,then create model for web service composition.

A WS_CPN is a tuple WS_CPN=(∑,P,T,A,C,G,E, I,In,Out) satisfying the requirements below[6]:

(i) ∑is a non-empty colour set,and represents the data type of web service.

(ii) P is a finite set of place,and represents the state of web service.

(iii) T is a finite set of transition,and represents the operation of web service.

(iv) A is a finite set of arc,and P∩T=P∩A=T∩A=Φ.

(v) C is a colour function,every token value is data typeC(p)
(p∈P).

(vi) G is a guard function,it makes a mapping from transition t∈T to boolean expression.Except for input parameters,the called operation of web service must fulfill the G function.

(vii) E is an arc expression function.Arc expression that calls input and output parameters of operation of web service may be null.

(viii) I is an initialization function which is defined from p∈P to a closed expresssion that doesn't contain any variable expression.It is used to specify the initial input parameters of the web service.

(ix) In is a input place,In={x∈P∪T | (x,in)∈A}=Φ, which is the initial place-set of the web service.

(x) Out is a output place,Out={ x∈P∪T | (out,x)∈A}=Φ, which is the ending place-set of the web service.

III. Verification of Web Service Composition

A. Structure of Web Service Composition

As the capability of an individual web service is limited,it is necessary to create new functions with existing web services.Web service composition is able to create a new value-added service and solve more complex problem by incorporating some existing web services together.

TABLE I.
MAPPING BETWEEN ELEMENTS

Web service composition elements	Coloured Petri net elements
Services process	Transition
Control dependence	Directed arc
Data dependence	Token
Data type	Token colour
Length of message	Token value
Delay	Time stamp

Web service composition includes four basic structures:

1) Sequence structure

A service execution is after another.

```
<sequence>
  S1
  S2
</sequence>
```

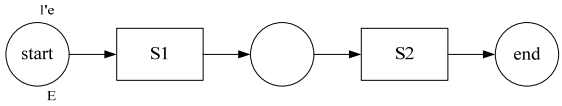

Figure 1. Sequence structure

2) Choice structure

According to the condition,a web service branch that meet the condition is executed.

```
<switch>
   <case condition="C1">
       S1
   </case>
   <case condition="C2">
       S2
   </case>
</switch>
```

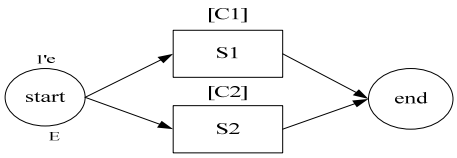

Figure 2. Choice structure

3) Parallel structure

Each web service is executed concurrently.

```
<flow>
    S1
    S2
</flow>
```

© 2013 ACADEMY PUBLISHER

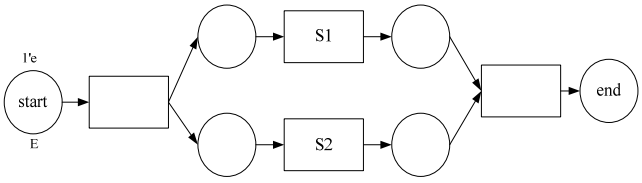

Figure 3. Parallel structure

4) Iteration structure

One or more web services are iteratively executed until a boolean condition is evaluated to be false.

```
<process>
  <while condition="C">
     S1
  </while>
</process>
```

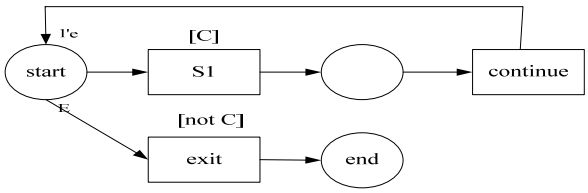

Figure 4. Iteration structure

B. Dynamic Properties of Coloured Petri Net

When it is represented by a CPN, web service composition using the properties of coloured petri net can be analyzed and verified.

1) Boundedness properties

Definition 1:Let a set of token elements X⊆TE (we use TE(p) to denote the set of token elements that correspond to a given place p)and a non-negative integer n∈N be given[1].

(i) n is an upper bound for X iff:

$\forall M\in[M_0>: |(M|X)|\leq n$.

(ii) n is a lower bound for X iif:

$\forall M\in[M_0>: |(M|X)|\geq n$.

The set X is bounded iff it has an upper bound.

2) Liveness Properties

Definition 2:Let a marking M∈M and a set of binding elements X⊆BE(we use BE(t) to denote the set containing all those binding elements which correspond to a given transition t) be given[1].

(i) M is dead iff no binding element is enabled,i.e.,iff:

$\forall x\in BE:\neg M[x>$.

(ii) X is dead in M iff no element of X can become enabled,i.e.,iff

$\forall M' \in[M> \forall x\in X: \neg M'[x>$.

(iii) X is live iff there is no reachable marking in which it is dead,i.e.,iff:

$\forall M' \in[M_0> \exists M'' \in[M'> \exists x\in X: M''[x>$.

3) Fairness Properties

We use $ENx(\sigma)=\sum_{i\in N+} ENx,i(\sigma)$, $OCx(\sigma)=\sum_{i\in N+} OCx,i(\sigma)$

Definition 3:Let X⊆BE be a set of binding elements and σ∈OSI an infinite occurrence sequence[1].

(i)X is impartial for σ iff it has infinitely many occurrences,i.e.,iff: $OCx(\sigma)=\infty$.

(ii)X is fair for σ iff an infinite number of enablings implies an infinite number of occurrences,i.e.,iff:$ENx(\sigma)=\infty \Rightarrow OCx(\sigma)=\infty$.

(iii)X is just for σ iff a persistent enabling implies an occurrence,i.e.,iff:

$\forall i\in\mathbb{N}_+$: $[Enx,i(\sigma)\neq 0 \Rightarrow \exists\ k\geq i: [ENx,k(\sigma)=0 \vee OCx,k(\sigma)\neq 0]]$.

4) Home Properties

Definition 4:Let a marking M∈𝕄 and s set of marking X⊆𝕄 be given[1]:

(i) M is a home marking iff:

$\forall M' \in[M_0>:M\in[M'>$.

(ii)X is a home space iff:

$\forall M' \in[M_0>:X\cap[M'>\neq\Phi$.

C. Formal Analysis of Coloured Petri Net

For formal analysis and validation,a full occurrence graphy (O-Graph) need to be defined.The basic idea behind occurrence graphs is to construct a graph containing a node for each reachable marking and an arc for each occurring binding element..

OG=(V,A,N) satisfying the requirements below:

(i) $V=[M_0>$,V is a set of all reachable markings from M_0.

(ii) $A=\{(M_1,b,M_2) \in V*BE*V \mid M_1[b>M_2\}$.

BE is a set of all binding elements.

(iii) $\forall a=(M_1,b,M_2) \in A: N(a)=(M_1,M_2)$.

Arc a satisfies N(a)=(M1,M2) which means the destination node M_2 can reach from M_1.

D. A Modling Instance Analysis

Below,I give an example that is service composition of providing commodity in warehouse management system.It includes three services of checking commodity amount, selecting supplier(by the price of commodity) and updating commodity list.The figure 5 shows CPN model of web service composition.

The colour sets and variables are described by CPN ML.

```
CPN Declarations:
colset E=with e;
colset AMOUNT=int;
colset ID=string;
colset NAME=string;
colset PRICE=int;
colset Lack_Comm=product ID*AMOUNT;
var data:Lack_Comm;
var am1,am2:AMOUNT;
var pri1,pr2,pri:PRICE;
var id :ID;
val cv=10;
```

© 2013 ACADEMY PUBLISHER

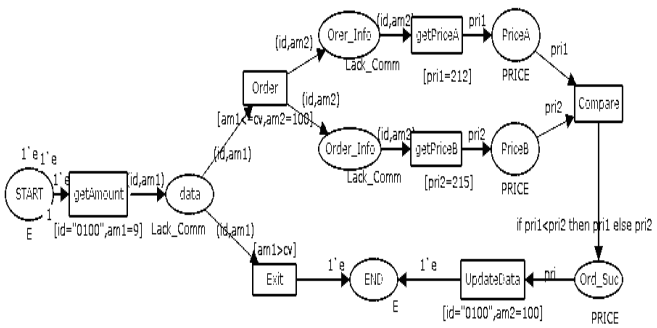

Figure 5. CPN model for web service composition

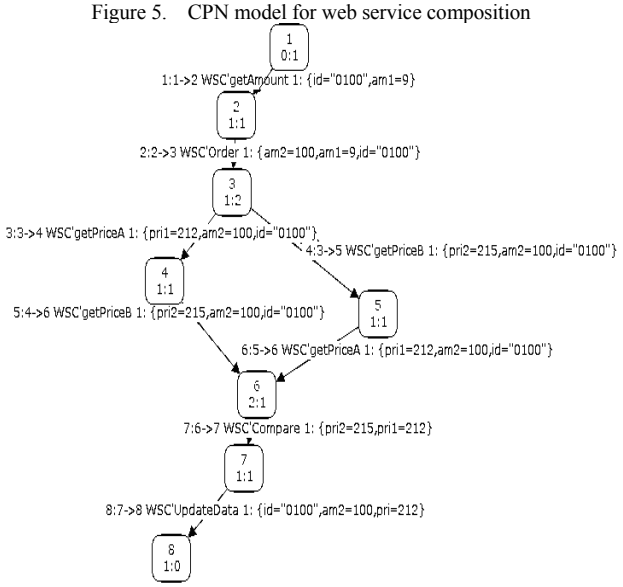

Figure 6. CPN state space fragment(boolean expression is true)

Boundedness Properties

Best Integer Bounds

	Upper	Lower
WSC'END 1	1	0
WSC'Ord_Suc 1	1	0
WSC'Order_Info 1	1	0
WSC'Orer_Info 1	1	0
WSC'PriceA 1	1	0
WSC'PriceB 1	1	0
WSC'START 1	1	0
WSC'data 1	1	0

Liveness Properties

Dead Markings

[8]

Dead Transition Instances

WSC'Exit 1

Live Transition Instances

None

Fairness Properties

No infinite occurrence sequences.

Home Properties

Home Markings

[8]

Figure 7. Report of CPN state space(boolean expression is true)

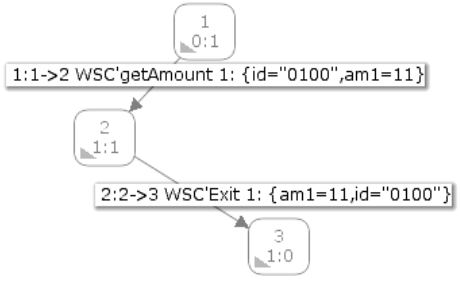

Figure 8. CPN state space fragment(boolean expression is false)

Boundedness Properties

Best Integer Bounds

	Upper	Lower
WSC'END 1	1	0
WSC'Ord_Suc 1	0	0
WSC'Order_Info 1	0	0
WSC'Orer_ Info 1	0	0
WSC'PriceA 1	0	0
WSC'PriceB 1	0	0
WSC'START 1	1	0
WSC'data 1	1	0

Liveness Properties

Dead Markings

[3]

Dead Transition Instances

WSC'Compare 1

WSC'Order 1

WSC'UpdateDase 1

WSC'getPriceA 1

WSC'getPriceB 1

Live Transition Instances

None

Fairness Properties

No infinite occurrence sequences.

Home Properties

Home Markings

[3]

Figure 9. Report of CPN state space(boolean expression is false)

IV. CONCULSION

Before we construct a key system based on web service composition, formal modeling and validation are essential for it.CPN model as a formal specification or presentation is an executable model. By means of formal analysis methods, it is possible to investigate the behaviour,verify properties,which can obtain increased confidence in the correctness of the model,and thereby the system[8].

REFERENCES

[1] Kurt Jensen,Coloured Petri Nets - Basic Concepts, Analysis Methods and Practical Use,vol. 1, pp.89-155,1992.

[2] Kurt Jensen, Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems,International Journal on Software Tools for Technology Transfer, Vol. 9, pp.213-251, 2007.

© 2013 ACADEMY PUBLISHER

[3] Robert Lukaszewski, Petri Nets in Measuring Systems Design,IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASURE-MENT,vol.57,NO.5,pp. 951-962,May 2008.
[4] Christine Choppy,A Modelling Approach with Coloured Petri Nets, F.Kordon and T.Vardanega(Eds.):Ada-Europe 2008,LNCS 5026,pp.73-86,2008.
[5] LI Jing-xia,Hierarchical Colored Petri Net Description Model for Web Service Composition,Computer Engineering,vol. 35,pp.39-40,December 2009.
[6] Yabei Wang,CPN-Based Verification of Web Service Composition Model, 2010 International Conference on Educational and Information Technology,pp.453-457.
[7] NI Wan-cheng,Survey on Web Swevices Composition Methods, Computer Engineering,vol.34,pp.79-81,February 2008.
[8] Tian Baojun,Formal Modelling And Validation for Software Process Based on CPN,Proceedings 2012 IEEE International conference on Software Engineering and Service Science,pp.89-93,2012.
[9] Zhang,Yuanyuan.Modeling web services composition with timed Pi calculus.Information Technology Journal,v10,n6,2011.
[10] Wu,Tiezhou.Research on semantic Web services composite model based on the OWL-S.Advanced Materials Research,v225-226,2011.
[11] Xia, Hong.Verification Web services composition based on OWL-S.2nd International Symposium on Knowledge Acquisition and Modeling, KAM 2009.
[12] Arias Fisteus,Jesús.Verification of web services compositions: Applying model checking to BPEL4WS[J],IEEE Latin America Transactions,v3,n1,2005.
[13] Dai,Guilan.A framework for model checking web service compositions based on BPEL4WS.IEEE International Conference on e-Business Engineering Workshops:SOAIC 2007.
[14] Enrique Caliz, Karthikeyan Umapathy, Analyzing web service choreography specifications using colored Petri nets. 2011 Proceedings of the 6th international conference on Service-oriented perspectives in design science research,pp. 412–426,2011.
[15] Deepak Chenthati, Supriya VaddiVerification of Web Services Modeled as Finite State Machines, Proceedings of the 2010 FourthAsia International Conference on Mathematical /Analytical Modelling and Computer Simulation, pp. 526-531,2010.

Associate professor Baojun Tian. 2000-2003, Master Degree in Computer Application Technology, Department of Computer Science, Inner Mongolia University,China. 1992-1996, Bachelor Degree in Mathematics, Department of Mathematics Science,Inner Mongolia University,China.His major fields of study are software engineering,formal approach and web service composition.

© 2013 ACADEMY PUBLISHER