A Petri Net based Method to Analyze the Logic Correction of Structure for WS-BPEL

Biao Wang [1, a], Qiang Hu [2, b]

[1]Library of Rizhao Campus, Qufu Normal University, Rizhao 276826, China;

[2]School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China.

[a]wangbiao2002@126.com, [b]huqiang200280@163.com

Abstract

WS-BPEL is frequently utilized to describe the flow structure of the service processes. As a kind of on-formalized description language, it is difficult to validate the structure logic correction after modeling the service process.To reduce the difficulity of checking the logic correction, the service processes described by WS-BPEL is firt modeled as service net. The mapping method between the main parts of WS-BPEL and service net is proposed. The linking rules of service net is also presented. Finally, a method to determine the logic correctness of WS-BPEL structure based on the reachable tree of logical Petri net is putforward.

Keywords

Logic, Web service, Petri net, Linking rule.

1. Introduction

Web service is the most commonly used way to provide network information service [1]. By providing a standardized interface, enterprises can encapsulate their business process as Web services and publish them in the specified service registration center. Users can query the service registry according to their own needs, and invoke some Web service to meet their serive requirment [2].

Web service is often developed with a simple function. To construct service process is the main form to realize complex business processes. A group of Web services is bulit as a service process in accordance with certain business logic process. WS-BPEL is a kind of business description language for Web service process. It provides a lot of modeling grammar to describe process logic structure. The business logic of a service process can be effectively modeled by WS-BPEL.

Since WS-BPEL is not a fommal service composition description language, the service process described by WS-BPEL need to be converted as a formal type model and then it can be analyze the property of structure[3]. Currently, a series of results have been achieved on the research of the formal model of WS-BPEL. For example, the Petri net based formal model of WS-BPEL[4,5], the process algebra and π calculus based model[6,7], the automaton based formal model of WS-BPEL[8].

Aiming at easily checking the structure logic correction of service process described based on the WS-BPEL，a logic Petri net based method is proposed in this study. The service process based on WS-BPEL was converted into service net which was modeled by logic Petri net. The structure logic correction of the service process is verified by testing the accessibility of logic Petri net. The linking rules of service net is also investigated in this paper.

2. Mapping Rules of Service Net

2.1 Service Net[9]

SN=(P, T, F, δ, , W) is a logic service net if the follows hold:

(1)W is a set of Web services that construct the process model of services flow;

(2)P= Pc∪Pd is a finite nonempty set of places, where

Pd is data place, Pd is used to describe the data dependency between services;

Pc is control place, Pc is used to describe the flow dependency between services, two special control places are defined, the initial place i and the terminal place o, let •i=o•=∅;

(3)$T=T_S \cup T_I \cup T_O$, T is a set of transitions, where Ts denotes a set of Web services, T_I denotes a set of logic input transitions; T_O denotes a set of logic output transitions;

(4)δ is a mapping function between Ts and W, i.e. $\forall t \in T_S$: δ (t)=s, s∈W;

2.2 Mapping Rules between Service Net and Logic Petri Net

The mapping rules between service net and WS-BPEL should be established before using the service net to model service processes. The main labels used to describe the structures of service processes in WS-BPEL are <reply>, <receive>, <invoke>, <sequence>, <if>, <while> and <repeat-until>. The following sections present the mapping rules of above labels and service nets.

1. <receive> and <reply>

The label <receive> is used to accept a piece of information from other services. <reply> is used to give an answer information. The variable means the name of interactive information. The modeling rules of <receive> and <reply> are shown in Fig.1.

```
<receive partnerLink="NCName"
  portType="QName"?
  operation="NCName"
  variable="BPELVariableName1"?
</receive>
```

```
<reply partnerLink="NCName"
  portType="QName"?
  operation="NCName"
  variable="BPELVariableName2"?
</reply>
```

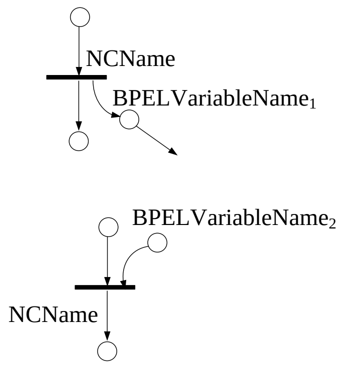

Fig. 1 The mapping rules of the < receive/reply> and service net

2. <invoke>

The label <invoke> is used to call the Web service published by the service provider. The operation of <invoke> utilize "partnerLink" to call the partner Web service. The "portType" and "operation" are employed to assign the interface and operation of WSDL. Similarly , The symbol of "inputVariable" and "outputVariable" are adopted to represent the input and output information. The modeling rules of <invoke> is shown in Fig.2.

```
<invoke partnerLink="NCName"
  portType="QName"?
  operation="NCName"
  inputVariable="BPELVariableName1"?
  outputVariable="BPELVariableName2"?
</invoke>
```

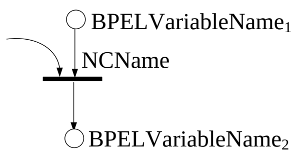

Fig. 2 The mapping rules of the <invoke> and service net

3. <sequence>

The label <sequence> is adopted to define a service process with a sequence structure. There may be several Web services in the structure of <sequence></sequence>. These services can be the atomic

Web services. They are carried out one by one according with the order in <sequence> </sequence>. Fig. 3 shows the mapping rules of <sequence> and service net.

```
<sequence standard-attributes>
  standard-elements
  activity+
</sequence>
```

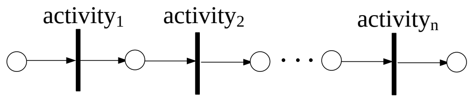

Fig.3 The mapping rules of the <sequence> and service net

4. <flow>

The label <flow> is used to define a group of Web services that perform with a parallel structure. Similar with <sequence>, there may be several Web services in the structure of <flow> </flow>. These services can be carried out simultaneously. A pair of logic input and output transitions need importing when the structure of <flow></flow> is used to model service process. Meanwhile, the logic expression need also be set by the number of transitions. Fig. 4 shows the mapping rules of <flow> and service net.

```
<flow standard-attributes>
  standard-elements
  <links>?
    <link name="NCName" />+
  </links>
  activity+
</flow>
```

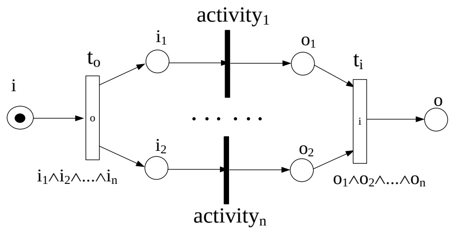

Fig. 4 The mapping rules of the <flow> and service net

5.<if>

The label <if> is used to define a group of Web services with an alternative structure. Normally, there are two Web service in the structure of <if></if>. The two services may be atomic Web services. Only one service can be performed according with the execute condition. Similar with the label <flow>, a pair of logic input and output transitions need importing while a logic expression "bool-exp" is also presented to describe the alternative structure. Fig. 5 shows the mapping rules of <if> and service net.

```
<if standard-attributes>
  standard-elements
  <condition expressionLanguage="anyURI"?>
  bool-expr</condition>
  activity1
    <else>?
     activity2
  </else>
</if>
```

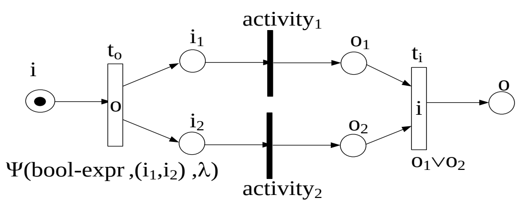

Fig. 5 The mapping rules of the <if> and service net

6. <while > and <repeatUntil>

The labels of <while > and <repeatUntil> are utilized to define the loop structures in service process. A logic expression "bool-expr" is added on the logic transition. The loop number can be set in the logic expression "bool-expr". The modeling rules of <while > and <repeatUntil> are shown in Fig.6.

```
<repeatUntil standard-attributes>
  standard-elements
  activity
  <condition expressionLanguage="anyURI"?>
   bool-expr</condition>
</repeatUntil>
```

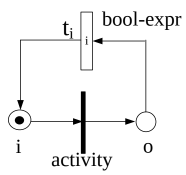

Fig.6 The mapping rules of the <repeatUntil\while> and service net

3. Link Fules of Service Nets

Two service nets with the relation of data interaction can be linked together. Three linking patterns are provided according the structure of WS-BPEL in this study. They are sequence linking pattern, branch linking pattern and join linking pattern.

If all the needed information can be totally provided by the prior service, the two Web service can be assembled by the sequence linking pattern. The output parameters of one Web service can be utilized as the collection of input parameters of two or more than two Web service, then these service can be organized by the branch linking pattern. Similar, if the input parameter need be responded by composition of output parameters in two or more than two Web service, they should be composed by the join linking pattern.

4. Logic Correction of Structure for WS-BPEL

The reachable tree is the most frequently used tool to analyze the accessibility of Petri net. Since the service net is modeled from the logic Petri net, the reachable tree is also an effective method to detect whether the structure of a service net is logic corrective.

An algorithm is provided to judge the logic correction of structure for WS-BPEL. A service process described by WS-BPEL firstly is transformed into a service net by the proposed mapping rules in section 2. The reachable tree is built to check the accessibility of the service net. If there is no embodied node, the logic correction of the service process is good. Algorithm 1 shows how to apply the detection of logic correction.

Algorithm 1. Logic correction of Structure S decribed by WS-BPEL

Input: a service process S described by WS-BPEL;

Output: logic correction of structure of S

(1)LC_WS(S)

(2){ LogCorrection=true, Boundflag=true;

(3) Map the service process S into a service net SN;

(4) Set M_0 as the initial node of reachable tree of SN;

(5)While (there exists a node with the status of “New”)

(6) Choose a node with the status of “New” and mark it as M；

(7) If there exist a node t marked as M then tag M with “old” and jump to (4);

(8) If $\forall t \in T : \neg M[t>$ then LogCorrection=false;

(9) For each $t \in T \wedge M[t>$ Do

(10) { Get M' : $M[t>M'$;

(11) If $\exists M'' \wedge M'' < M'$ and $\exists j \wedge M''(s_j) < M'(s_j)$ then Boundflag=false;

(12) Remove the status of “New” from M and jump to (4);}

(13) If (LogCorrection &&Boundflag ==True) then *S* is logic correction. }

5. Conclusion

As a popular description language for service process, WS-BPEL is often used to express the service composition. However, WS-BPEL is not the formal language. It is difficult to check whether a service process described by WS-BPEL is logic correct in its structure. A logic Petri net based method to verify the logic correction of service process is proposed in this paper. By analyzing the structure of service net, we can easily check the logic correction of service process described by WS-BPEL. The logic correction of interactive service processes will be studied in the future work.

Acknowledgements

This work is supported by the key research program of Shandong Province under Grant 2016GGX101031, the promotive research fund for young and middle-aged scientisits of Shandong Province under Grant BS2015DX010 and BS2015ZZ006.

References

[1] Jula A, Sundararajan E, Othman Z. Cloud computing service composition: A systematic literature review. Expert Systems with Applications, Vol. 41 (2014), No.8, pp.3809-3824.

[2]Lemos A L, Daniel F, Benatallah B. Web service composition: a survey of techniques and tools. ACM Computing Surveys, Vol. 48, (2016), Vol. 3, pp. 33-42.

[3]Sheng Q Z, Qiao X, Vasilakos A V, et al. Web services composition: A decade's overview. Information Sciences, Vol. 280, (2014), pp. 218-238.

[4] Ouyang C, Verbeek E, Van Der Aalst W M P, et al. Formal semantics and analysis of control flow in WS-BPEL. Science of Computer Programming, Vol. 67, (2007), No. 2, pp.162-198.

[5] Lohmann N, Verbeek E, Ouyang C, et al. Comparing and evaluating Petri net semantics for BPEL. International Journal of Business Process Integration and Management, Vol. 4,(2009), No.1, pp. 60-73.

[6]Lucchi R, Mazzara M. A pi-calculus based semantics for WS-BPEL. The Journal of logic and algebraic programming, Vol.70, (2007), No.1, pp.96-118.

[7]Ripon S, Uddin M S, Barua A. Web Service Composition--BPEL vs cCSP Process Algebra International Conference on Advanced Computer Science Applications and Technologies, 2012, pp. 150-155.

[8]Pu G, Zhao X, Wang S, et al. Towards the Semantics and Verification of BPEL4WS. Electronic Notes in Theoretical Computer Science,Vol. 151, (2006), No.2, pp. 33-52.

[9] Hu Q, Du Y, Yu S X. Service net algebra based on logic Petri nets. Information Sciences, Vol. 268, (2014), pp. 271-289.