I.J. Education and Management Engineering **2011, 3, 39-46**
Published Online September 2011 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijeme.2011.03.06

Available online at http://www.mecs-press.net/ijeme

Research of BPEL Modeling Technique Based on Colored Petri Net

Guo Feng[a], Wang Yi[b]

[a]*College of Information Engineering, North China University of Technology, Beijing, China*
[b]*College of Information Engineering, North China University of Technology, Beijing, China*

Abstract

Web Services Business Process Execution Language (BPEL) modeling technique based on colored Petri net is proposed in this paper. Introduce a BPEL modeling principle based on colored Petri net, and a method of generating BPEL code from the colored Petri net. CPN Tools is used to build the colored Petri net model. At last, the case study of booking ticket was provided to illustrate the applicability of the transformation algorithm. The generated BPEL can run on Oracle BPEL server.

Index Terms: colored Petri Net; WS-BPEL; CPN Tools; web service

1. Introduction

WS-BPEL is an acronym for Web Services Business Process Execution Language (or BPEL for short).It is a language for business process based on web services. The WS-BPEL provides new web service by reusing existing web services.

There are two kinds of processes, executable process and abstract process. Abstract processes describe process behavior partially without covering every detail of execution. An Abstract Process can be "implemented" by a set of executable processes. The "implementation" is also known as executable completion. Elements and attributes of an executable process may be hidden in an Abstract Process, by either just omitting them or by replacing them with opaque constructs. The common language constructs of abstract and executable processes have the same semantics. [1]The executable processes include error handling and exception handling.

With the advent of service-oriented architecture, BPEL is used more and more widely. BPEL may invoke web services provided by different partners, and compose them in a complex order. So how to generate correct BPEL is an important question. In [5], the author builds abstract BPEL process from object Petri net. In [8], the author translates colored Petri net to work flow net, then generate BPEL according the work flow net.

Most of the integrated development environments could validate the process, but it may miss some errors, which lead running error on the server.

Identify applicable sponsor/s here. If no sponsors, delete this text box. (sponsors)

In order to build BPEL without errors, it could make a Petri net which describes the model first; translate the Petri net to BPEL after validated. It will introduce how to build Petri net model with CPN Tools, and does not concern with the BPEL error handling and exception handling. How to validate colored Petri net has studied a lot, so it will not be involved neither.

2. Build Colored Petri Net model

2.1 Colored Petri Net

Definition 1: The colored Petri net record is as follows, it has six-tuple CPN= (P, T, C, I_-, I_+, M_0): satisfied [2]

1) P and T are the sets of place and transformation.
2) $P \cup T \neq \emptyset$, $P \cap T = \emptyset$.
3) C is color function which is defined in.
4) I_- and I_+ are negative connected functions and positive connected functions.
5) $\forall p \in P, \exists t \in T: I- (p, t) \neq 0 \vee I_+ (p, t) \neq 0$,
   $\forall t \in T, \exists p \in P: I- (p, t) \neq 0 \vee I_+ (p, t) \neq 0$.
6) M_0 is the initial characteristic which is defined in P.

The colored Petri net model described both the system static structure, and the system dynamic change process. Colored Petri net vicissitude occurs with the request of the flow the complete surface perspicuity system of dynamic movement process.

CPN Tools which is developed by Aarhus University was used to build colored Petri net models. It is one of the best colored Petri net modeling and emulation tools. The CPN Tools use ML language to build and analysis colored Petri net. After building the model, the CPN Tools also could validate it later. It will provide a model without error.

2.2 Building rules

The BPEL has two kinds of activities: structured activities which can contain other activities and basic activities which only perform intended purpose. The structured activities define the business logic. In order to simplify the translation, we make some rules to build the colored Petri net model.

BPEL has several basic activities, including invoke, receive, assign, reply, wait and so on. There is an example which illuminates how to build the basic activity. It explains with assign activity. The code below defines an assign activity in BPEL.

```
<assign name="Assign_1">
        <copy>
          <from source…/>
          <to destination…/>
        </copy>
</assign>
```

Fig.1 is the assign activity defined in colored Petri net. In CPN Tools, the comment at the right bottom of palaces stands for place color, which could be treated as input or output variable type; the place name express variable name and translation name express activity name. The assign activity in Fig.1 assigns a “SourceType” type variable to a “DestinationType” type variable.

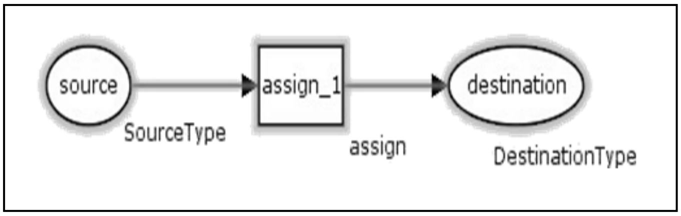


Fig 1. Assign activity

After building basics activities, they should be made up to structured activities.

Create two places with the name start and end which help to find where the process begins and where the process ends as shown in Fig.2.

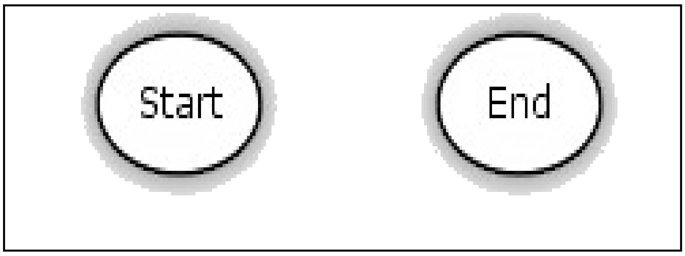


Fig 2. start place and end place

If it was a sequence activity, connect start place, end place and work process.Fig.3 shows a process only has one basic activity.

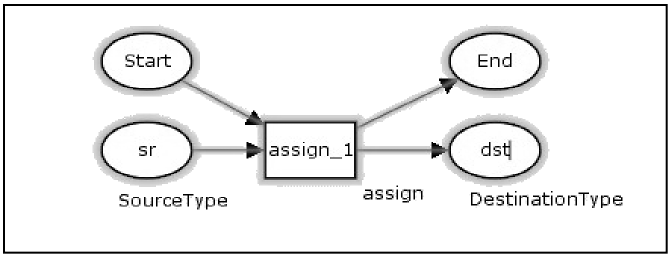


Fig 3. Sequence activity

Complex structured activities including switch activities, parallel processing and Repetitive activities are difficult to identify with program. In order to simplify the identify process, we build them in subpage. And replace the whole structure with the subpag. Add the subpage type at the right bottom of the subpage as shown in Fig.4.In subpage, it should add structure type switch also as shown in Fig.5.

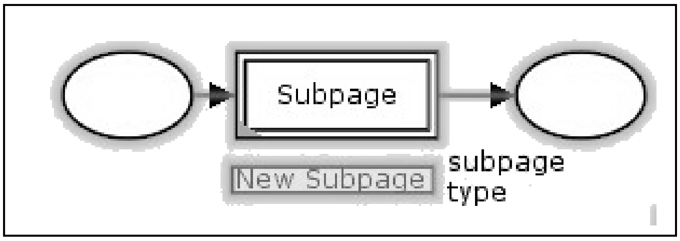


Fig 4. Complex structure activity

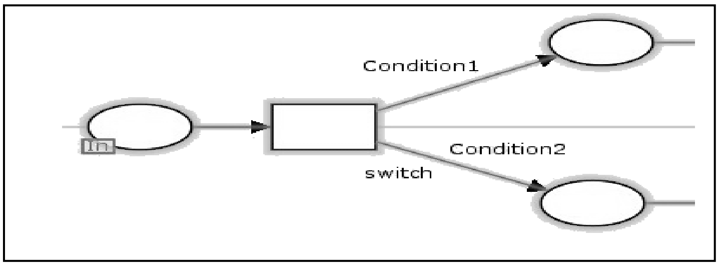


Fig 5. Switch activity

Parallel processing subpage should add type flow as shown in Fig.6.

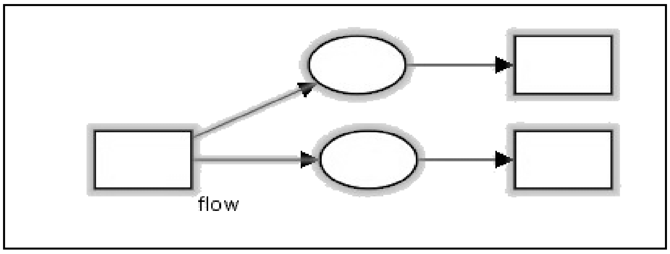


Fig 6. flow activity

BPEL has more than one repetitive activities, we only support key word while.Fig.7 is a repetitive activity. Beside add type loop to the first place, add a place to store loop conditions.

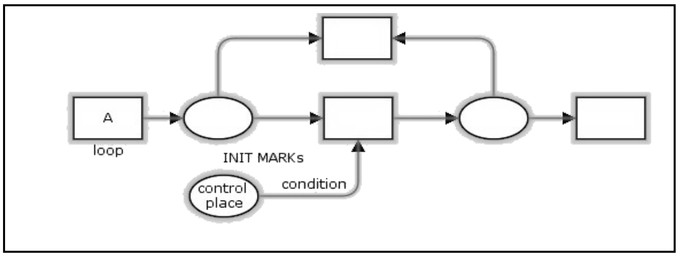


Fig 7. Loop activity

In order to generate BPEL with a program, we need to add extra information which the program needs to build BPEL code including partner links and variables. We treat them as color set. This is a partner link below, connect its attribute to a string, and divide them with semicolon.

< partnerLink name= "plName" partnerLinkType= "pltype" myRole= "QurryFareProvider" partnerRole = "QurryFareRequester"/>

The color set would be:

plName;pltype;QurryFareProvider;QurryFareRequester=STRING

The extra information is useless when validate the Petri net model, so it should be add after the validation.

3. Translate Petri net model to BPEL

The CPN Tools document and BPEL document are both in Extensible Markup Language (XML) format. It could be operate with Dom model. The arithmetic will generate the BPEL document.

```
1. LoadFile（filePath）；
2. ParseDocument（document）；；
3.SetStartPlacs();
  SetEndPlace();
4.root = CreateBPELRoot();
SetCurrentNode(root)
5.tran = FindNextTrans()，
if  tran == NULL && isMainNet()
goto 11
    if  tran == NULL && isSubpage()
              goto father node
if isBPELActivity()，
   goto 6
else goto 7
6. Build activity with the information collect from the Petri net model; insert the activity to the father node；
7.type = GetSubpageType(page)
if  type == flow
goto 8
if type == switch
goto 9
If type == loop
goto 10
8.flowNode = CreateFlow(Vertex)
foreach next child
childBPEL = SetCurrentNode(child)
goto 5
9.switchNode = CreateSwitch（Vertex），
foreach direct next child
childBPEL = SetCurrentNode(child)
goto 5
10. loopNode = CreateLoop（Vertex），
foreach direct next child
childBPEL = SetCurrentNode(child)
goto 5
11. End.
```

4. Example

Fig.8 is a Petri net model of quarrying ticket price. It queries the ticket class according the employee number. Then quarry fare from two airlines, return the lower price. In Fig.8, subpage ChooseTicket and ChooseTicket illustrate flow and switch activities, the detail is showed in Fig.9 and Fig.10.

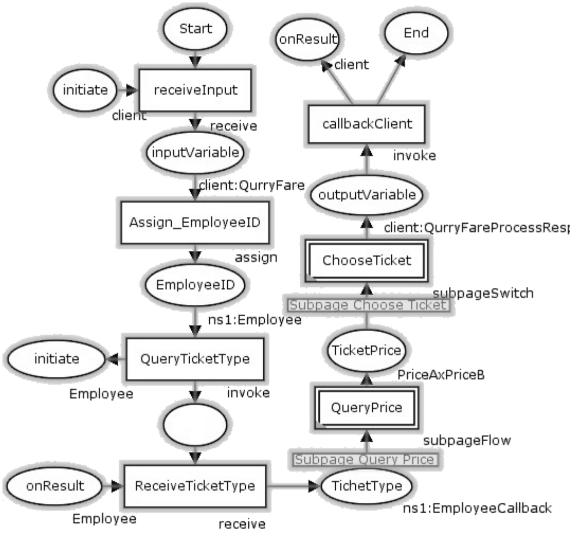


Fig 8. Example of auaring price

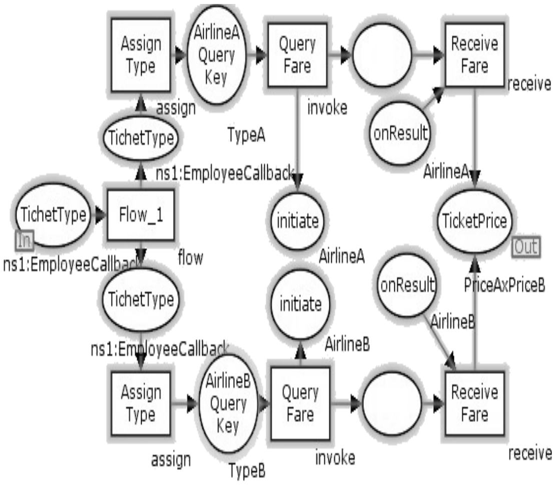


Fig 9. Subpage Query Price

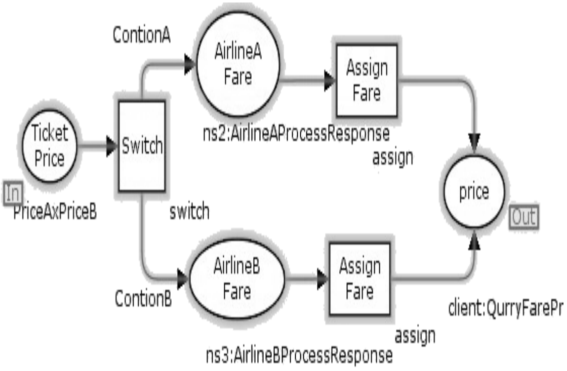
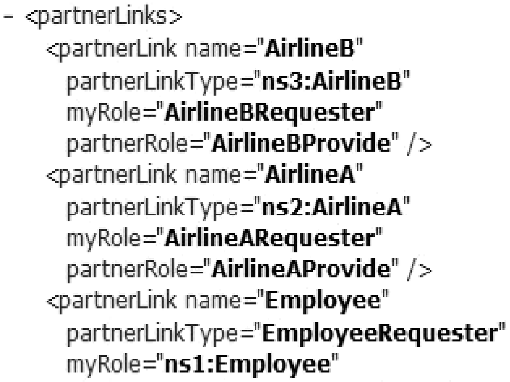


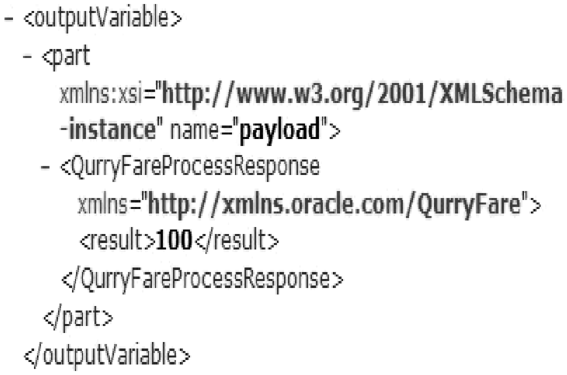
Fig 10. Subpage Choose Ticket

Part of the generated BPEL document was shown in Fig.10, and the result running on the server was shown in Fig.11.



```
- <partnerLinks>
    <partnerLink name="AirlineB"
      partnerLinkType="ns3:AirlineB"
      myRole="AirlineBRequester"
      partnerRole="AirlineBProvide" />
    <partnerLink name="AirlineA"
      partnerLinkType="ns2:AirlineA"
      myRole="AirlineARequester"
      partnerRole="AirlineAProvide" />
    <partnerLink name="Employee"
      partnerLinkType="EmployeeRequester"
      myRole="ns1:Employee"
```

Fig 11. Part of generated BPEL document



```
- <outputVariable>
  - <part
      xmlns:xsi="http://www.w3.org/2001/XMLSchema
      -instance" name="payload">
    - <QurryFareProcessResponse
        xmlns="http://xmlns.oracle.com/QurryFare">
        <result>100</result>
      </QurryFareProcessResponse>
    </part>
  </outputVariable>
```

Fig 12. Running result

5. Conclusion

The article introduces a theory of generating BPEL document from Petri net model. It includes the Petri net model building rules and translating arithmetic.

For further research, we will add error handling and exception handling to the generated BPEL.

References

[1] Business process execution language for web services. http://www.ibm.com/developerworks/webservices/library/specification/ws-bpel4people

[2] Yanfeng Ge, An HV transmission line overhaul plan optimization method based on the color Petri net, Control and Decision Conference, 2008, pp.38738–3876.

[3] Kurt Jensen. An introduction to the theoretical aspects of colored Petri nets. http://www.springerlink.com/content/p66k27518p64243x/fulltext.pdf.

[4] Lin Qiang, Hu Hao, LV Jian, BPEL modeling technique based on object petri net, Computer Engineering. .2009，35(2):74-75.

[5] Deng Xin-guo, Translating Colored Petri Nets Collaboration Model to BPEL,Computer science, 2008，135(111):262-265.

[6] Qian Zhong-zhu,Automatic composition of Petri net based on web service,Chinese journal of computer science, 2006，29（7）：1057-1065.

[7] Li Qing, Modeling of software development process based on CPN.China university of petroleum.,