



Journal of  
**Software  
Engineering**

ISSN 1819-4311



Academic  
Journals Inc.

[www.academicjournals.com](http://www.academicjournals.com)

## **A Logic Petri Net Method to Analyze the Well-structured Property of WS-BPEL**

Hu Qiang

School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, 266061, China

### **ABSTRACT**

A Web service process is well-structured, means its branch flows are splited and converged correctly in the logical structure. Well-structure is an import property to guarantee a service process to work normally. To detect the well-structured property of Web service process described by WS-BPEL, a logical Petri net method is proposed. The service process based on WS-BPEL is modeled as a service net form logical petri nets and the well-structured property is mapped into the features of logic expressions in the logical transitions of the service net. The formal definition and decision method of well-structure is proposed and the examples are also provided to show how to detect whether a service net is well-structured or not. From the proposed method in this study, the well-structured property of WS-BPEL can be easily analyzed.

**Key words:** Web service, well-structure, logic petri net, WS-BPEL

### **INTRODUCTION**

Web service is a mature technology to achieve the network-based software oriented service computing. A lot of Web services with small grain and simple functions were developed and deployed in the Internet. The software developer can invoke them to build new application system or compose them into Web service processes to cater complicated service requests.

To realize service composition effectively, many composition languages and interface specifications are proposed, such as WS-BPEL, WS-CDL and WSCI. WS-BPEL is the mainstream language of service composition in the industry. However, WS-BPEL is not a strict formal composition language and it adopts the XML format codes to describe the service process. The structural defects of the Web service process, such as interactive deadlock, structure redundancy, or non-well structure cannot be discovered from the code of WS-BPEL directly. These structural defects are the fatal errors for the Web service processes and should be detected and eliminated.

A Web service process is well-structured means that its branch processes are splited and converged correctly in the logical structure. The concept of well-structure is frequently used in the research of workflow (Eder *et al.*, 2013; Ha and Suh, 2008; Van der Aalst *et al.*, 2011). Since the Web service process is similar with workflow, many researchers introduced well-structure into the study of Web service to express whether the logical structure is correct and soundness in the service process. However, most of the proposed methods (Chae *et al.*, 2008; Li *et al.*, 2011; Xiong *et al.*, 2010) are with a high complexity in detect the well-structured property.

How to easily analyze the well-structured property of WS-BPEL is focused in this study. A method to analyze whether a service process based on WS-BPEL is well-structured or is proposed which based on logic petri net.

## SERVICE NET BASED ON LOGIC PETRI NET

A logic petri net is generated by appending the logic expressions on some of the transitions in petri nets. It had been successfully used to the model and analyze the real-time cooperative systems (Du *et al.*, 2007), E-commerce workflows (Du *et al.*, 2009) and the Web service processes (Hu *et al.*, 2014). Now the definitions of logical petri net and service net are provided.

### Definition 1

**Logic petri net:** LPN = (P, T, F, I, O, M) is a logic petri net if the following hold:

- P is a finite set of places
- T includes three subsets of transitions, i.e.,  $T = T_D \cup T_I \cup T_O$ , where,  $T_D$  denotes a set of traditional transitions,  $T_I$  denotes a set of logic input transitions and  $T_O$  denotes a set of logic output transitions
- F is a flow relation, i.e., a set of directed arcs  $F \subseteq (P \times T) \cup (T \times P)$
- I is a mapping function from a logic input transition to a logic input expression, i.e.,  $\forall t \in T_I, I(t) = f_i$
- O is a mapping function from a logic output transition to a logic output expression, i.e.,  $\forall t \in T_O, O(t) = f_o$
- M:  $P \rightarrow \{0, 1\}$  is a marking function,  $\forall p \in P, M(p)$  denotes the token count in p
- The firing rules of the transitions in LPN are defined as follows:
  - $\forall t \in T_D$ , if  $\forall p \in \bullet t: M(p) = 1$ , t is enabled. If t is enabled, it can fire and a new marking M' is generated from current marking M, denoted by  $M[t > M']$ , where:

$$M'(p) = \begin{cases} M(p) - 1, & p \in \bullet t - t \bullet \\ M(p) + 1, & p \in t \bullet - \bullet t \\ M(p), & \text{otherwise} \end{cases}$$

- $\forall t \in T_I, I(t) = f_i$ , if  $f_i | M = \bullet T \bullet$ , i.e., all input places of t satisfy the input expression  $f_i$  at M, t is enabled. If t is enabled, it can fire and generate a new marking M', where:

$$M'(p) = \begin{cases} M(p) - 1, & p \in \bullet t \\ M(p) + 1, & p \in t \bullet \\ M(p), & \text{otherwise} \end{cases}$$

- $\forall t \in T_O, O(t) = f_o$ , if  $\forall p \in \bullet t: M(p) = 1$ , t is enabled. If t is enabled, it can fire and a new marking M' is generated and  $\exists S \subseteq t \bullet, S$  must satisfy  $f_o | M' = \bullet T \bullet$ :

$$M'(p) = \begin{cases} M(p) - 1, & p \in \bullet t \\ M(p) + 1, & p \in S \\ M(p), & \text{otherwise} \end{cases}$$

### Definition 2

**Service net:** A service net is a labeled logic petri net and defined as  $SN = (LPN, I, O, \Gamma, \varphi)$  where:

- LPN is the net model of the service net with the following rules:
  - $P = P_c \cup P_d$  is a set of places, where  $P_d$  is a set of data places interacting with the exterior Web services and  $P_c$  is a set of control places which represent the states of the Web service
  - $T = T_s \cup T_I \cup T_O$  is a set of transitions, where,  $T_s$  represents the operations in a service process;  $T_I$  and  $T_O$  denote the set of logic transitions used to control the business flow in the service process
- $F$  represents the transformational relation between service operations and service states
- $i$  and  $o$  are two special control places, where  $i$  is the initial place and  $o$  is the terminal place of a service process, with  $\bullet i = \bullet o = \emptyset$  and the initial marking  $M_0(i) = 1$ ,  $M_0(p) = 0$  for any other place  $p$  and
- $\varphi: T \leftrightarrow \Gamma$  is a bidirectional mapping, where  $\Gamma$  is a alphabet

**MAPPING RULES BETWEEN SERVICE NET AND WS-BPEL**

Many literatures had provided the methods to convert the service process described by WS-BPEL into petri net. There are some open-source engines, which can analyze and run WS-BPEL, such as RiftSaw and ActiveBPEL2.0. With the help of engine tools, the model transformation can be easily achieved. Since focusing on the well-structured property of WS-BPEL, the mapping rules between the main control structures of WS-BPEL and service net are proposed in this section.

The control structure of WS-BPEL includes the labels of <receive>, <reply>, <invoke>, <sequence>, <if>, <pick>, <flow>, <while> and <repeatUntil> (Tan *et al.*, 2009). To simplify the structure of converted model from WS-BPEL to service net, control structures with similar functions are classified into the same category. The mapping rules of control structure between WS-BPEL and service net are presented in Fig. 1. The label <variables> in the WS-BPEL is mapped into a place with a variable name.

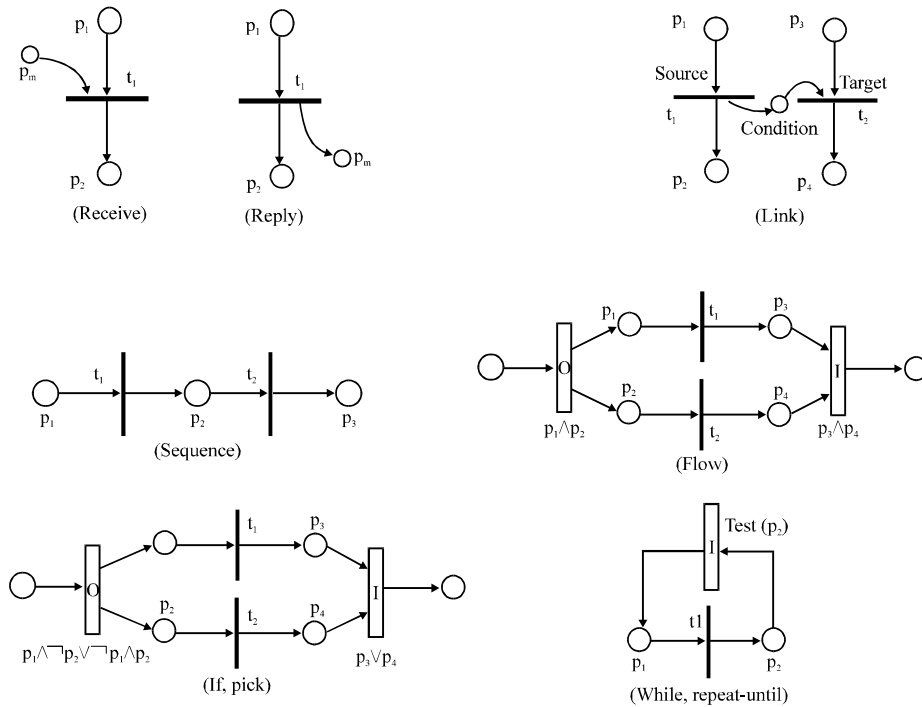


Fig. 1: Mapping rules between WS-BPEL and service nets

## DEFINITION OF WELL-STRUCTURE

The definition of well-structured in WS-BPEL is based on the concept of well-structured workflow nets. A well-structured service process is that which parallel branches initiated by an AND split should be synchronized by an AND joint. And the alternative branches initiated by OR split should be synchronized by an OR joint.

Let  $F = \{f_1, f_2, \dots, f_n\}$  be a set of service processes,  $t_o$  and  $t_i$  be two logic transitions linking  $F$  with parallel or alternative structure.

### Definition 1

**Logic expression on parallel or alternative structure:**  $O(t_o)$  and  $I(t_i)$  represent the logic expressions on  $t_o$  and  $t_i$ , respectively. To distinctly differentiate the type of logic transitions, the logic expressions  $O(t_o)$  and  $I(t_i)$  should be formulated as the following standard forms.

- $O(t_o) = f_{1,i} \wedge \neg f_{2,i} \wedge \dots \wedge \neg f_{n,i} \vee \neg f_{1,i} \wedge f_{2,i} \wedge \dots \wedge \neg f_{n,i} \vee \dots \vee \neg f_{1,i} \wedge \neg f_{2,i} \wedge \dots \wedge f_{n,i}$ , denoted as  $O_v$
- $O(t_o) = f_{1,i} \wedge f_{2,i} \wedge \dots \wedge f_{n,i}$ , denoted as  $O_\wedge$
- $I(t_i) = f_{1,o} \vee f_{2,o} \vee \dots \vee f_{n,o}$ , denoted as  $I_v$  and
- $I(t_i) = f_{1,o} \wedge f_{2,o} \wedge \dots \wedge f_{n,o}$ , denoted as  $I_\wedge$

A symbol  $\partial$  is introduced to represent the type of logical expression of logical transition.

### Definition 2

**Matching logical transitions:** Let  $F = \{f_1, f_2, \dots, f_n\}$  be a service net,  $t_1$  and  $t_2$  are the logical transition to link  $F$ ,  $\forall f_j: \pi_1(f_j, i) = \{t_o\}$ ,  $\forall f_j: \pi_2(f_j, o) = \{t_i\}$ , ( $1 \leq j \leq n$ ). If  $t_1$  and  $t_2$  are logical input transition and logical output transition or in converse,  $t_1$  and  $t_2$  are called matching logical transitions.

### Definition 3

**Synergic expression:** Let  $F = \{f_1, f_2, \dots, f_n\}$  be a service net,  $t_o$  and  $t_i$  are the logical transition to link  $F$ ,  $\forall f_j: \pi_1(f_j, i) = \{t_o\}$ ,  $\forall f_j: \pi_2(f_j, o) = \{t_i\}$ , ( $1 \leq j \leq n$ ). If  $\partial(t_o) = O_v \wedge \partial(t_i) = I_v$  or  $\partial(t_o) = O_\wedge \wedge \partial(t_i) = I_\wedge$ ,  $O(t_o)$  and  $I(t_i)$  are called the synergic expression, they are denoted as  $O(t_o) \diamond I(t_i)$ .

### Definition 4

**Well-structured service net:** SN is a service net, if  $\forall t_o \in SN, \exists t_i: O(t_o) \diamond I(t_i) \wedge \forall t_i \in SN, \exists t_o: I(t_i) \diamond O(t_o)$  then SN is a well-structured service net.

## DETECT WELL-STRUCTURE IN WS-BPEL

**Decision method:** From definition 4, two steps need to be checked in analyze whether a Web service process is well-structured. One is whether its logical transitions are occurred dually, i.e., a logical input transition with a corresponding logical output transition, the other is whether the dual

---

Algorithm 1: Detect well-structure in the service process based on WS-BPEL

---

**Input:** A service process SP described by WS-BPEL

**Output:** Whether the service process SP is well-structured or not

- Construct the service net SN for SP from the mapping rules
  - Traverse the SN to find the dual logical transitions and check whether they are matching logical transitions
  - For each matching logical transitions, check their logical expression is synergic expression or not
  - If all the dual logical transitions are matching logical transitions with synergic expression, then the SP is well-structured, otherwise, the SP is not well-structured
-

logical transitions are with a matching expression. A concise algorithm to detect whether a WS-BPEL based service process is well-structured is proposed in algorithm 1.

**Examples:** Due to the algorithm 1, there are two types of service net which are not well-structured. One is with non-matching logical transition, the other is with non-synergic expression.

Figure 2 shows two service nets while Fig. 2a is well-structured and Fig. 2b is not well-structured. The logical transition  $t_{o2}$  is with a non-matching logical transition, i.e., there is no

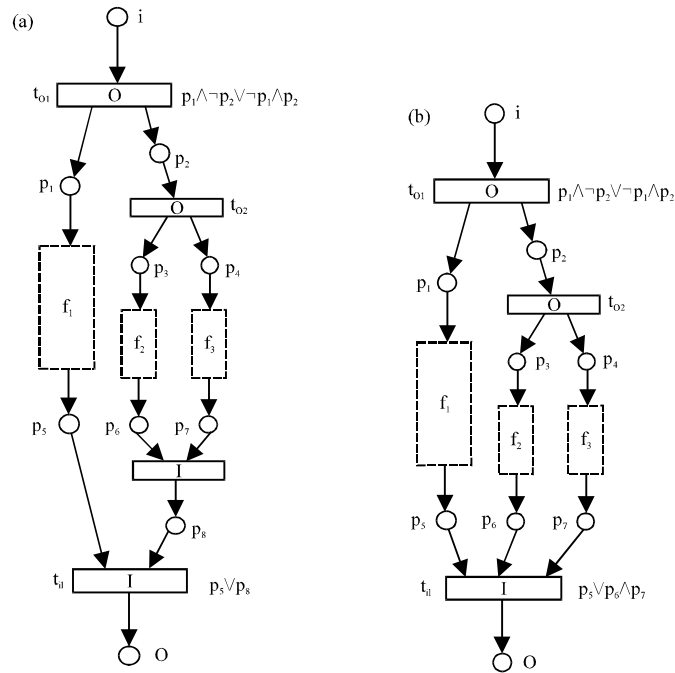


Fig. 2(a-b): Service net with non-matching logical transition

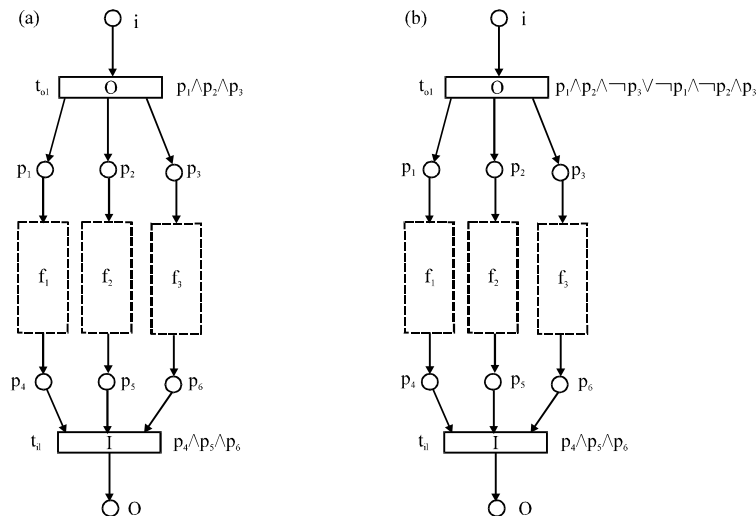


Fig. 3(a-b): Service net with non-synergic expression

Table 1: Comparison of the logic petri net method and other methods based on petri nets

Method name	Whether to build the coverability tree of petri net	Problem of space explosion
CRG method	Yes	Yes
Siphon based method	Yes	Yes
Logic petri net method	No	No

logical transition matching  $t_{o2}$ . The branch flow  $f_2$  and  $f_3$  should be converged first by a logical input transition and then synchronized by the logical transition  $t_{i1}$ , i.e., the well-structured service process is shown as the service net in Fig. 2a.

Similarly, Fig. 3 shows two service nets while Fig. 3a is well-structured and Fig. 3b is not well-structured. The logical transition  $t_{o1}$  and  $t_{i1}$  are with non-synergic expression. The logical expression of logical transition  $t_{o1}$  is  $p_1 \wedge p_2 \wedge \neg p_3 \vee \neg p_1 \wedge \neg p_2 \wedge p_3$ , it means that the branch flow  $f_1$  and  $f_2$  should be executed simultaneously or only the branch flow  $f_3$  is executed. The branch flow  $f_1$ ,  $f_2$  and  $f_3$  are not permitted to be executed simultaneously. However, the logical expression of logical transition  $t_{i1}$  is labeled a logical expression  $p_4 \wedge p_5 \wedge p_6$ , all the branch flow  $f_1$ ,  $f_2$  and  $f_3$  should be executed simultaneously while the logical transition  $t_{i1}$  is triggered. Thus, the expression labeled on the logical transition  $t_{o1}$  and  $t_{i1}$  is not a synergic expression and the service net is not well-structured.

The service net shown in the Fig. 3a is well-structured, the logical expression of logical transition  $t_{o1}$  is updated as  $p_1 \wedge p_2 \wedge p_3$  and it is the synergic expression of  $p_4 \wedge p_5 \wedge p_6$ .

**Comparison:** The method based on Communicating Reachability Graph (CRG) (Li *et al.*, 2011) and the siphon based method (Xiong *et al.*, 2010) to analyze the well-structured property were provided based on the petri net. The communicating reachability graph is a variant of the coverability tree, thus it cannot avoid the space explosion when the service process with a complicated business flow. The siphon based method was also faced the same problem since the coverability tree need building when the siphon of the petri net was computed. Since the two petri net based methods were with the problem of space explosion they could only be used to detect the well-structured property of simple service processes. An efficient result cannot be achieved when the above method used to analyze the well-structured property of complicated service processes. The comparison of our proposed method and the above methods (Li *et al.*, 2011; Xiong *et al.*, 2010) is shown in Table 1. The logic petri net method is not with the problem of space explosion and it is also with a high efficiency since the coverability tree is not built when analyzing the well-structured property.

## CONCLUSION

Well-structure is a vital property of service processes. Since WS-BPEL is not with a formal format, it is difficult to detect the well-structured property in the service process described by WS-BPEL. A logical petri net method to analyze the well-structured property of WS-BPEL is proposed in this study. By mapping the WS-BPEL into a service net modeling by logic petri net, the well-structured property is converted by the restriction of logical expression in the service net. The well-structured property is easily analyzed by detecting whether the logical transitions are with the synergic expressions. The examples are also given to shown the efficiency of the proposed method in this study.

## **ACKNOWLEDGMENTS**

This study is supported by the National Basic Research Program of China under Grant No. 2010CB328101; the National Natural Science Foundation of China under Grant No. 61170078 and 61273180; the Doctoral Program of Higher Education of the Specialized Research Fund of China under Grant No. 20113718110004, the Basic Research Program of Qingdao City of China under Grant No. 13-1-4-116-jch.

## **REFERENCES**

- Chae, H.S., J.S. Lee and J. Bae, 2008. An approach to checking behavioral compatibility between Web services. *Int. J. Software Eng. Knowledge Eng.*, 18: 223-241.
- Du, Y.Y., C.J. Jiang and M.C. Zhou, 2007. Modeling and analysis of real-time cooperative systems using petri nets. *IEEE Trans. Syst. Man Cybern. Part A: Part A: Syst. Hum.*, 37: 643-654.
- Du, Y.Y., C.J. Jiang, M.C. Zhou and Y. Fu, 2009. Modeling and monitoring of E-commerce workflows. *Inform. Sci.*, 179: 995-1006.
- Eder, J., E. Panagos and M. Rabinovich, 2013. Time Constraints in Workflow Systems. In: *Information Systems Engineering*, Bubenko, J., J. Krogstie, O. Pastor, B. Pernici, C. Rolland and A. Solvberg (Eds.). Springer, Berlin Heidelberg, ISBN: 978-3-642-36925-4, pp: 191-205.
- Ha, S. and H.W. Suh, 2008. A timed colored petri nets modeling for dynamic workflow in product development process. *Comput. Ind.*, 59: 193-209.
- Hu, Q., Y.Y. Du and S.X. Yu, 2014. Service net algebra based on logic petri nets. *Inform. Sci* 268: 271-289.
- Li, X., Y. Fan, Q.Z. Sheng, Z. Maamar and H. Zhu, 2011. A petri net approach to analyzing behavioral compatibility and similarity of web services. *IEEE Trans. Syst. Man Cybern.*, 4: 510-521.
- Tan, W., Y.S. Fan and M.C. Zhou, 2009. A petri net-based method for compatibility analysis and composition of web services in business process execution language. *IEEE Trans. Automat. Sci. Eng.*, 6: 94-106.
- Van der Aalst, W.M.P., K.M. van Hee, A.H.M. ter Hofstede, N. Sidorova, H.M.W. Verbeek, M. Voorhoeve and M.T. Wynn, 2011. Soundness of workflow nets: Classification, decidability and analysis. *Formal Aspects Comput.*, 23: 333-363.
- Xiong, P.C., C. Pu and M. Zhou, 2010. Protocol-level service composition mismatches: A petri net siphon based solution. *Int. J. Web Serv. Res.*, 7: 1-20.