

# Web Service Automatic Composition Model Based on Colored Petri Nets

Kai Nie<sup>1</sup>, Houxiang Wang<sup>1</sup>, Xiaopei Jing<sup>1</sup>, Zhihao Xie<sup>2</sup>

<sup>1</sup>College of Electronic Engineering, Naval University of Engineering, Wuhan, China

<sup>2</sup>College of Electronic and Information Engineering, Beijing University of Aeronautics and Astronautics, Beijing, China  
Email: 1999104133@163.com

Received 2012

## ABSTRACT

As the capability of an individual Web service is limited, it's necessary to create new functionalities with existing Web services. Web services composition is the ability to create a new value-added service by incorporating some existing web services together. A model based colored Petri net (CPN) to provide semantic support for web service composition is proposed. The basic composite constructs in the model are sequence, concurrent, choice and loop. A closed composing algebra is defined to obtain a framework which enables declarative composition of web services. Finally modeling composite processes of Web services based on CPN is applied to a case of naval vessel command and control system.

**Keywords:** Web Service; Colored Petri Net (CPN); Web Service Composition; WS-BPEL

## 1. Introduction

Web services have become an emerging and promising technology for designing and building complex inter-enterprise business applications out of single web-based software components. To establish the existence of a global component market in order to enforce extensive software reuse, service composition has received increasing research efforts. Current technologies based on universal description, discovery, and integration (UDDI), web service description language(WSDL),and simple object access protocol (SOAP) do not realize complex web service combinations, hence providing limited support in service composition [1,2].Web services should be based on open standards, platform independent, application independent, and enable to share data and resources. Web service composition is a task of combining and linking existing web services to create new web processes in order to add value to the collection of services.

In the research related to web services, several initiatives have been conducted with the intention to provide platforms and languages that will allow easy integration of heterogeneous systems. In particular, such languages as UDDI,WSDL,SOAP and part of DAML-S ontology (Service Profile and Service Grounding),define standard ways for service discovery, description and invocation (message passing).Some other initiatives such as WS-BPEL and DAML-S Service Model, are focused on representing service compositions where flow of a process and bindings between services are known apriori [3-5].

Ontology-driven web services composition is used to discover and assemble services into processes for easier and better quality workflow executions given increasing number and complexity of web services [6]. Besides that, current solutions for web service composition include web components,  $\pi$ -calculus, Model checking/FSM and Petri nets [7]. In 2003, Hamadi proposed a Petri net-based model for web service composition [8], in which the data types cannot be distinguishable because an elementary Petri net model is used. In a recent research, a CPN model for web service composition is proposed [9]. However, the rules and procedures of composition must be defined previously, and the services composition chain cannot be generated automatically without pre-defined conditions. In the message oriented activity based Petri net model [10], the web service composition relies on messages, which increase complexity of composition.

For the sake of fast computation, many researchers prefer Petri nets [7-13], since they are well suited for capturing flows in web services, modeling the distributed nature of web services, representing methods in a web service and reasoning about the correctness of the flows. A web service behavior is basically a partially ordered set of operations. Therefore; it is straightforward to map it into a Petri net. Operations are modeled by transitions and the state of the service is modeled by places. The arrows between places and transitions are used to specify causal relations. Therefore, information is modeled by tokens and the types of information are modeled by the

colors of the tokens.

It is assumed that a Petri net, which represents the behavior of a service, contains one input place (i.e., a place with no incoming arcs) and one output place (i.e., a place with no outgoing arcs)[12]. A Petri net with one input place for absorbing information, and one output place for emitting information, will facilitate the definition of the composition operators and the analysis as well as the verification of certain properties (e.g., reach ability, availability, and security). At any given time, a web service can be in one of the following states: Not instantiated, Ready, Running, Suspended, or Completed [14]. When a web service is in the Ready state, it means that tokens in their corresponding input place enable post set (set of transitions) of input place to fire. Whereas the Completed state means that preset (set of transitions) of output place has fired and has generated tokens in corresponding output place.

In this paper, a colored Petri net [15-17](CPN)based algebra for modeling web services is proposed. The model is expressive enough to capture semantics of complex service combinations and their respective specificities. The web service is formally defined and the obtained framework enables declarative composition of web services. Modeling composite processes of Web services based on CPN is applied to a case of naval vessel command and control system.

## 2. Modeling Composite Web Services Based on Colored Petri Nets

CPN is a graphical oriented language for design, specification, simulation and verification of systems. CPN combines the strength of Petri nets with the strength of programming languages. Petri nets provide the primitives of the description of the synchronization of concurrent processes, while programming languages provide the primitives for the definition of data types and the manipulation of data values. The formal definition of CPN is shown below:

Definition 1 CPN (colored Petri net)

A CPN is a tuple  $CPN = (\Sigma, P, T, F, C, G, E, I)$  satisfying the requirements below:

- i)  $\Sigma$  is a finite set of non-empty types, called color sets.
- ii)  $P$  is a finite set of places.
- iii)  $T$  is a finite set of transitions.
- iv)  $F$  is a finite set of arcs. It is defined from  $A$  into  $P \times T \cup T \times P$ .
- v)  $C$  is a color function. It is defined from  $P$  into  $\Sigma$ .
- vi)  $G$  is a guard function. It is defined from  $T$  into expressions such that:

$[Type(G(t)) = B \wedge Type(Var(G(t))) \subseteq \Sigma]$ .

vii)  $E$  is an arc expression function. It is defined from  $A$  into expressions such that:  $[Type(E(a)) = C(p)MS \wedge$

$Type(Var(E(a))) \subseteq \Sigma]$ , where  $p(a)$  is the place which  $F$  connects.

viii)  $I$  is an initialization function. It is defined from  $P$  into closed expressions (without variables) such that:  $\forall p \in P: [Type(I(p)) = C(p)MS]$ .

Definition 2 Service net  $SN = (CPN, i, o)$  is called a service net if and only if:

- i) CPN is a colored Petri net;
- ii)  $In \in P$  is the input place with  $i = \emptyset$ ;
- iii)  $Out \in P$  is the output place with  $o = \emptyset$ ;
- iv) If we add a transition  $t$  to CPN which connects  $i$  and  $o$  (i.e.,  $i = o, i = i$ ), then the resulting Petri net is strongly connected.

Based on CPN and WS-BPEL, the model is defined as follows:

- A Web service state is represented by a CPN place. An input-place represents the input of the corresponding activity. The out-place represents the output of the corresponding activity.
- Messages and process variables are represented by tokens.
- A Web service activity is represented by a CPN transition. A <receive> activity is represented by a transition which has an in-place. A <reply> activity is represented by a transition which has out-place. An <invoke> activity is represented by a pair of transitions. A structured activity is represented by a substitution transition.
- Data types involved in Web services composition are represented by color sets.
- Web services Conditions except for input parameters that must be met are represented by CPN guard expressions.
- The input and output parameters of a Web service activity are represented by CPN arcs. The control flow between activities is captured by connecting transitions with arcs.

• The whole process of the composite service is represented by a CPN Net  $C$  (net composition). Each partner service is represented by a CPN Service net. Net  $C$  interacts with Service net through arcs connecting the in-and out-places of Service net. Each arc must be labeled with a token variable that matches the colored set declared for the in-place/out-place.

Each WS-BPEL process can be translated to a CPN model, and then a formal model of WS-BPEL can be obtained, which allows the analysis and verification techniques and tools developed for CPN can be exploited in the context of WS-BPEL.

## 3. Composing Web Services

Researchers have discussed various composite constructs [12-14]. In this paper, we take sequence, concurrent, choice and loop constructs as basic constructs specified

in the control flow. We also give a formal semantics to the proposed algebra in terms of CPN.

### 3.1. Composite Constructs

Below we describe syntax and informal semantics of the service algebra operators. The constructs are chosen to allow basic and advanced web service composition. The set of services can be defined as:

Definition 3

$$S ::= |X|Seq(S,S)|Conc(S,S)|Choice(S,S)|Loop(S)$$

X represents a service constant, used as an atomic or basic service in this context.

$Seq = (S_1, S_2)$  represents a composite service that performs the service  $S_1$  followed by the service  $S_2$ .  $Seq(\cdot)$  is an operator of sequence. If a composite service that performs either the service  $S_1$  followed by the service  $S_2$ , or  $S_2$  followed by  $S_1$ , it is called unordered sequence. In practice, we can decide the order by any condition since the order is not important, then unordered sequence could be treated as sequence.

$Conc = (S_1, S_2)$  represents a composite service that performs the service  $S_1$  and  $S_2$  independently. Both services are concurrently enabled and the overall composite service waits until both services are completed.  $Conc(\cdot)$  is a concurrent operator.

$Choice = (S_1, S_2)$  represents a composite service that behaves as either service  $S_1$  or service  $S_2$ . Once one of them is executed, another service is discarded.  $Choice(\cdot)$  is a choice operator. The choice is not arbitrary (in fact, there is no absolute arbitrariness), and depends on conditions. Thus, the condition construct with Boolean variants could be treated as choice.

$Loop(S)$  represents a composite service that performs a certain number of times of the service S.  $Loop(\cdot)$  is a loop operator.

The proposed algebra verifies the closure property. It guarantees that each result of an operation on services is a service to which one can again apply algebra operators. Software engineers thus are able to build more complex services by aggregating and reusing existing services through service algebra.

### 3.2. Formal Semantics

Let

$$SN_j = (\sum_j, P_j, T_j, F_j, C_j, G_j, E_j, I_j, i_j, o_j)$$

for  $j = 1, \dots, n$ , be n web services such that  $P_j \cap P_k = \emptyset$  and  $T_j \cap T_k = \emptyset$  for  $j \neq k$ . The black is the control transition and the clarity is the service transition.

Definition 4 the service  $Seq = (S_1, S_2)$  is defined as

$$Seq = (S_1, S_2) = SN = (\sum, P, T, F, C, G, E, I, i, o) \text{ where}$$

$$P = P_1 \cup P_2, T = T_1 \cup T_2 \cup \{t\},$$

$$F = F_1 \cup F_2 \cup \{(o_1, t), (t, i_2)\}, i = i_1, o = o_2,$$

$$G = G \cup G(o_1, t), E = E \cup E(i_2, t), \text{ and } I = I_1 \cup I_2.$$

Given  $S_1$  and  $S_2$ ,  $Seq = (S_1, S_2)$  is represented graphically by CPN shown in **Figure 1(a)**.

Definition 5 the service  $Conc = (S_1, S_2)$  is defined as

$$Conc = (S_1, S_2) = SN = (\sum, P, T, F, C, G, E, I, i, o) \text{ where}$$

$$P = P_1 \cup P_2 \cup \{i, o\}, T = T_1 \cup T_2 \cup \{t_i, t_o\},$$

$$F = F_1 \cup F_2 \cup \{(i, t_i), (t_i, i_1), (t_i, i_2), (o_1, t_o), (o_2, t_o), (t_o, o)\},$$

$$G = G \cup G(In, t_i) \cup G(o_1, t_o) \cup G(o_2, t_o),$$

$$E = E \cup E(i_1, t_i) \cup E(i_2, t_i) \cup E(o, t_o), \text{ and } I = I_1 \cup I_2.$$

Given  $S_1$  and  $S_2$ ,  $Conc = (S_1, S_2)$  is represented graphically by CPN shown in **Figure 1(b)**.

Definition 6 The service  $Choice = (S_1, S_2)$  is defined as  $Choice = (S_1, S_2) = SN = (\sum, P, T, F, C, G, E, I, i, o)$

$$\text{where } P = P_1 \cup P_2 \cup \{i, o\}, T = T_1 \cup T_2 \cup \{t_{i_1}, t_{i_2}, t_{o_1}, t_{o_2}\},$$

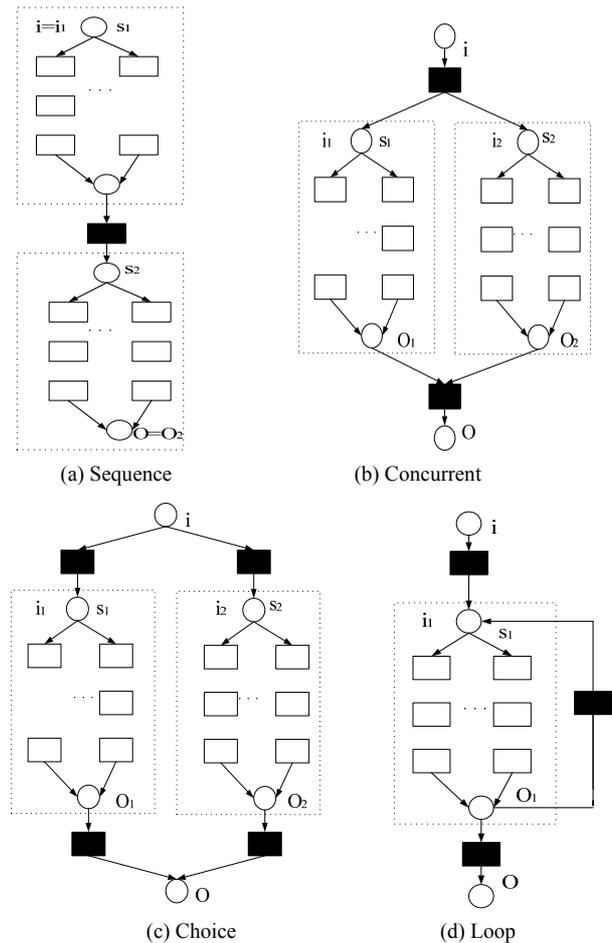


Figure 1. Colored Petri nets of basic constructs.

$$F = F_1 \cup F_2 \cup \{(i, t_{i_1}), (i, t_{i_2}), (t_{i_1}, i_1), (t_{i_2}, i_2), (o_1, t_{o_1}), (o_2, t_{o_2}), (t_{o_1}, o), (t_{o_2}, o)\},$$

$$G = G \cup G(i, t_{i_1}) \cup G(i, t_{i_2}) \cup G(o_1, t_{o_1}) \cup G(o_2, t_{o_2}),$$

$$E = E \cup E(i_1, t_{i_1}) \cup E(i_2, t_{i_2}) \cup E(o, t_{o_1}) \cup E(o, t_{o_2}),$$

and  $I = I_1 \cup I_2$ .

Given  $S_1$  and  $S_2$ ,  $Choice = (S_1, S_2)$  is represented graphically by CPN shown in **Figure 1(c)**.

Definition 7 the service  $Loop = (S_1)$  is defined as  $Loop = (S_1) = SN = (\sum, P, T, F, C, G, E, I, i, o)$  where

$$P = P_1 \cup \{i, o\}, \quad T = T_1 \cup \{t_i, t_o, t\},$$

$$F = F_1 \cup \{(i, t_i), (t_i, i_1), (o_1, t_o), (t_o, o), (o_1, t), (t, i_1)\},$$

$$G = G \cup G(i, t_i) \cup G(o_1, t_o) \cup G(o_1, t),$$

$$E = E \cup E(i_1, t_i) \cup E(o, t_o) \cup E(i_1, t), \text{ and } I = I_1.$$

Given  $S_1$ ,  $Loop = (S_1)$  is represented graphically by CPN shown in **Figure 1(d)**.

Using the four kinds of basic structures, some atomic processes and composite processes can be composed to form a new composite process. This sort of composite structure is suitable to be used to model large systems.

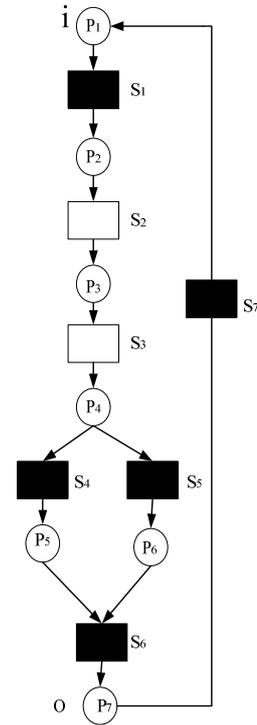
#### 4. Application

In this section, modeling composite processes of Web services based on CPN is applied to a case. The case is the process of a naval vessel command and control system shown in **Figure 2**. There are seven transitions and seven places in the CPN net. The seven transitions are: S1: information obtain; S2: information process; S3: command and control; S4: torpedo launch and control; S5: missile launch and control; S6: combat efficiency evaluation; S7: repeat attack. S1 obtains information of enemies from all sorts of sensors, then S2 process the information and sends them to the S3, S3 conduct the target motion analysis and set the weapons launching parameter, then P4 choices S4 or S5 based the enemy types, if the enemies are submarines, then S4 torpedo launch and control is conducted; if the enemies are vessels, then S5 missile launch and control is conducted; S6 evaluate the combat efficiency, if the aim is not achieved, S7 repeat attack is conducted.

Three of important verification problems are: reachability, safety and existence of deadlock. Within the model, reachability, safety and existence of deadlock of the composite service can be analyzed.

#### 5. Conclusions

A model based colored Petri net (CPN) to provide semantic support for web service composition is proposed. The formal semantics of the composition operations is



**Figure 2. Colored Petri net of command and control system.**

expressed in terms of CPNs by providing a direct mapping from each operator to a colored Petri net construction. A closed composing algebra is defined to obtain a framework which enables declarative composition of web services. Also modeling composite processes of Web services based on CPN is applied to a case of naval vessel command and control system. Further work will include the use of a formal model allows verification of reach ability, existence of deadlock and security properties and detection of inconsistencies both within and between services.

#### REFERENCES

- [1] A. Tsalgatidou and T. Pilioura, "An Overview of Standards and Related Technology in Web Services," *Distributed and Parallel Databases*, Vol. 12, No. 2-3, 2002, pp. 135-162. doi:10.1023/A:1016599017660
- [2] S. Dustdar and W. Schreiner, "A Survey on Web Services Composition," *International Journal of Web and Grid Services*, Vol. 1, No. 1, 2005, pp. 1-30. doi:10.1504/IJWGS.2005.007545
- [3] A. Bucchiarone and S. Gnesi, "A Survey on Services Composition Languages and Models," in *Proceedings of International Workshop on Web Services Modeling and Testing*, Palermo. Berlin: Springer-Verlag Press, 2006, pp. 51-63.
- [4] J. Rao and X. Su, "A Survey of Automated Web Service Composition Methods," *Lecture Notes in Computer Science*, Vol. 3387, 2005, pp. 43-54.

- [doi:10.1007/978-3-540-30581-1\\_5](https://doi.org/10.1007/978-3-540-30581-1_5)
- [5] S. Dustdar and W. Schreiner, "A Survey on Web Services Composition," *International Journal of Web and Grid Services*, Vol. 1, No. 1, 2005, pp. 1-30.  
[doi:10.1504/IJWGS.2005.007545](https://doi.org/10.1504/IJWGS.2005.007545)
- [6] A. I. Budak, B. Aleman-Meza and R. Zhang, "Ontology-driven Web Services Composition Platform," in *Proceedings of IEEE International Conference on E-commerce Technology*, San Diego, Los Alamitos: IEEE Computer Society Press, 2005, pp. 146-152.
- [7] N. Milanovic and M. Malek, "Current Solutions for Web Service Composition," *IEEE Internet Computing*, Vol. 8, No. 6, 2004, pp. 51-59. [doi:10.1109/MIC.2004.58](https://doi.org/10.1109/MIC.2004.58)
- [8] R. Hamadi and B. Benatallah, "A Petri Net-based Model for Web Service Composition," in *Proceedings of the 14th Australasian Database Conference*, Adelaide. Darlinghurst: Australian Computer Society, 2003, pp. 191-200.
- [9] Y. Guo, Y. Du and J. Xi, "A CP-net Model and Operation Properties for Web Service Composition," *Chinese Journal of Computers*, Vol. 29, No. 7, 2006, pp. 1067-1075.
- [10] Z. Qian, S. Lu, L. Xie, "Automatic Composition of Petri Net based Web Services," *Chinese Journal of Computers*, Vol. 29, No. 7, 2006, pp. 1057-1066.
- [11] J. P. Thomas, M. Tomas and G. Ghinea, "Modeling of Web Services Flow," in *Proceedings of IEEE International Conference on E-commerce*, San Diego, California. Los Alamitos: IEEE Computer Society Press, 2005, pp. 391-398.
- [12] R. Hamadi and B. Benatallah, "A Petri Net-based Model for Web Service Composition," in *Proceedings of the 14th Australasian Database Conference*, Adelaide. Darlinghurst: Australian Computer Society, 2003, pp. 191-200.
- [13] Z. Tan, C. Lin and H. Yin, "Approximate Performance Analysis of Web Services Flow using Stochastic Petri net," *Lecture Notes in Computer Science*, Vol. 3251, 2004, pp. 193-200. [doi:10.1007/978-3-540-30208-7\\_31](https://doi.org/10.1007/978-3-540-30208-7_31)
- [14] H. Schuster, D. Georgakopoulos and A. Cichocki, "Modeling and Composing Service-based and Reference Process-based Multi-enterprise Processes," *Lecture Notes in Computer Science*, Vol. 1789, 2000, pp. 247-263.
- [15] M. Lars, S. C. Kristiansen and K. Jensen, "The Practitioner's Guide to Colored Petri Nets," *International Journal on Software Tools for Technology Transfer*, Vol. 2, 1998, pp. 98-132. [doi:10.1007/s100090050021](https://doi.org/10.1007/s100090050021)
- [16] H. Kang, X. Yang and S. Yuan, "Modeling and Verification of Web Services Composition based on CPN," in *2007 IFIP International Conference on Network and Parallel Computing-Workshops*, IEEE Computer Society Press, 2007, pp. 613-617.
- [17] Z. Zhang, F. Hong and H. Xiao, "A Colored Petri Net-based Model for Web Service Composition," *Journal of Shanghai University*, Vol. 12, No. 4, 2008, pp. 323-329. [doi:10.1007/s11741-008-0409-2](https://doi.org/10.1007/s11741-008-0409-2)