Technologies Used for Petri Net-Based Web Service Composition

Azizbek Marakhimov a, Jaegeol Yim b,1 and Jaehun Joo c

a Cooperative Department of Techno Management, Dongguk University
b Department of Computer Engineering, Dongguk University
c Department of Information Management, Dongguk University
Seokjang Dong 707, Gyeongju, Gyeongbuk, 780-714, KOREA
azizbek.marakhimov@gmail.com, {yim, givej}@dongguk.ac.kr

Abstract. Web service is a software system that can be used by other software system. Therefore, we can relatively easily and economically build up a huge practical software system by using Web services. Consequently, Web service composition, building up a business process system by making use of published Web services, has been extremely popular in the field of Software Engineering. Among the technologies utilized in Web service composition, Petri net-based one is most attractive because it is both graphical and mathematical. So, we review the technologies used for Petri net-based Web service composition.

Keywords: Web service composition, Petri net, Colored Petri net, Logical Petri net

1 Introduction

Web service technology provides a way to integrate some distributed service units over the network into a coordinative system. Compared with the traditional enterprise application integration techniques, it provides better interoperability for data exchange and application invocation. Therefore, it has been widely adopted for constructing distributed applications [1].

Service Oriented Architecture (SOA) is a software design and software architecture design pattern based on discrete pieces of software providing application functionality as services to other applications. SOA makes it easy for computers connected over a network to cooperate. Every computer can run an arbitrary number of services, and each service is built in a way that ensures that the service can exchange information
with any other service in the network without human interaction and without the need
to make changes to the underlying program itself [2]. Web services are one of the key
technologies to support SOA.

In computing, aspect-oriented programming (AOP) is a programming paradigm
that aims to increase modularity by allowing the separation of cross-cutting concerns
[3]. In computer science, cross-cutting concerns are aspects of program that affect
other concerns where a concern is a particular set of information that has an effect on
the code of a computer program [4]. A concern can be as general as the details of
database interaction or as specific as performing a primitive calculation, depending on
the level of conversation between developers and the program being discussed [4].
These concerns often cannot be cleanly decomposed from the rest of the system in
both the design and implementation, and result in either scattering (code duplication),
tangling (significant dependencies between systems), or both [5].

According to the authors of [6], Aspect Oriented Web Service Composition
(AOWSC) is the most formal way of composition. Composition is a set of relevant
candidate web services, synchronously tied up with each other, delivered to the re-
questor at run time based on the service request. In AOWSC, the candidate web se-
rvices are developed as aspects, dynamically selected at run time based on the service
request, composed through a weaving mechanism and thus handle composition. The
authors of [6] claimed that they projected the formal approach of aspect oriented web
service composition by using Petri Net algebra. Therefore, we review the Petri nets
introduced by those authors.

2 Petri Net Model of Web Service Composition

The authors of [1] discussed complexity analysis for Petri net-based business process
in Web service composition. They used the Petri net defined in Table 1 [7]:

<table>
<thead>
<tr>
<th>Table 1. Formal Definition of a Petri net [7]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Definition 1:</strong> A Petri net is a 5-tuple, ( PN = (P,T,F,W,M_0) ) where:</td>
</tr>
<tr>
<td>( P = { p_1, p_2, \ldots, p_m } ) is a finite set of places,</td>
</tr>
<tr>
<td>( T = { t_1, t_2, \ldots, t_n } ) is a finite set of transitions,</td>
</tr>
<tr>
<td>( F \subseteq (P \times T) \cup (T \times P) ) is a finite set of arcs (flow relation),</td>
</tr>
<tr>
<td>( W : F \to [1,2,3,\ldots] ) is a weight function,</td>
</tr>
<tr>
<td>( M_0 : P \to {0,1,2,3,\ldots} ) is the initial marking,</td>
</tr>
<tr>
<td>( P \cap T = \Phi ) and ( P \cup T \neq \Phi ).</td>
</tr>
</tbody>
</table>
| A Petri net structure \( N = (P,T,F,W) \) without any specific initial marking is denot-
ed by \( N \). |
| A Petri net with the given initial marking is denoted by \( (N,M_0) \). |
3 Logical Petri Net

The authors of [6] used Logical Petri Nets to describe Web services. A Logical Petri Net (LPN), defined in Definition 2, is high-level abstraction of Petri nets with inhibitor arcs. In LPNs, the input and output of a transition are restricted by logical input expressions and logical output expressions.

Definition 2 (Logical Petri Nets): Suppose $LN=(P,T,F,DT,I,O)$, $LPN=(LN,M)$ is a Logical Petri Net when:

1. $P$ is a finite set of places;
2. $T=T_D \cup T_I \cup T_O$ is a finite set of transitions, $T \cup P \neq \Phi$, $\forall t \in T_I \cup T_D: t \in \Phi$, and $P,T_D,T_I,T_O$ are disjoint with each other, where:
   - (a) $T_D$ is a set of delay transitions, and the delay time is described as $\tau$;
   - (b) $T_I$ is a set of logical input transitions, and $\forall t \in T_I$, all input places of $t$ are restricted by a logical input expression named $I_f$;
   - (c) $T_O$ is a set of logical output transitions, and $\forall t \in T_O$, all output places of $t$ are restricted by a logical output expression named $O_f$;
3. $F \subseteq (P \times T) \cup (T \times P)$ is a finite set of arcs;
4. $DT$ is a real function, $\forall t \in T_D: DT(t) = \tau \in R$, $\tau$ is the delay time;
5. $I$ is a logical input function, $\forall t \in T_I: I(t) = f_I$ is a logical input expression;
6. $O$ is a logical output function, $\forall t \in T_O: O(t) = f_O$ is a logical output expression;
7. $M: P \rightarrow \{0,1\}$ is a marking function, $\forall p \in P: M(p)$ is the number of tokens in $p$;
8. The rules of transition’s trigger are:
   - (a) $\forall t \in T_D: DT(t) = \tau$, if $\forall p \in \bullet: M(p)=1$, $t$ is enabled at $M$; if the enabled time of $t$ is equal to or greater than $\tau$, then $t$ can fire, and generates a new state $M'$, $\forall p \in \bullet - \bullet: M'(p)=M(p)-1$, $\forall p \in \bullet: M'(p)=M(p)+1$, otherwise, $M'(p)=M(p)$;
   - (b) $\forall t \in T_I : I(t) = f_I$, if $f_{I|T'} = \bullet T \bullet$, in other words, $\bullet t$ satisfy a logical input expression $f_I$ at $M$, then $t$ is enabled at $M$; if $t$ is enabled, then it can fire, and generates a new state $M'$, $\forall p \in \bullet - \bullet: M'(p)=M(p)-1$, $\forall p \in \bullet: M'(p)=M(p)+1$, $\forall p \notin \bullet \cup \circ : M'(p)=M(p)$;
   - (c) $\forall t \in T_O: O(t) = f_O$, if $\forall p \in \bullet: M(p)=1$, then $t$ is enabled at $M$; if $t$ is enabled, then it can fire, and generate a new state $M'$, $\forall p \in \bullet: M'(p)=M(p)-1$, $\forall p \notin \bullet \cup \circ: M'(p)=M(p)$, while $t \bullet$ should satisfy the expression $f_{O|T'} = \bullet T \bullet$, in other words, $t \bullet$ must satisfy a logical expression $f_o$ at $M'$.
4 Colored Petri Nets

In order to ensure the correctness of Web service composition in WS-BPEL, the authors of [8] propose modeling, analysis and verification based on Colored Petri Nets (CPN) defined in Definition 3 [9].

Definition 3: A net is a tuple \( N = (P, T, A, \Sigma, C, N, E, G, I) \) where:
- \( P \) is a set of places.
- \( T \) is a set of transitions.
- \( A \) is a set of arcs.
- \( \Sigma \) is a set of color sets defined within CPN model. This set contains all possible colors, operations and functions used within CPN.
- \( C \) is a color function. It maps places in \( P \) into colors in \( \Sigma \).
- \( N \) is a node function. It maps \( A \) into \( P \times T \cup T \times P \).
- \( E \) is an arc expression function.

5 Conclusions

We have surveyed the definitions of Petri net, Colored Petri net and Logical Petri net and applications of those Petri nets on Web service composition. Referring to these Petri nets, we are working on Petri net based semantic web service composition.

References

2 http://en.wikipedia.org/wiki/Service-oriented_architecture
4 http://en.wikipedia.org/wiki/Concern_(computer_science)
5 http://en.wikipedia.org/wiki/Cross-cutting_concern