



Modeling and Analysis of Web Services Composition Based on Higher-Order Petri Net

Ali M. Meligy¹, Hani M. Ibrahim², Amal M. Aqlan³^{1, 2, 3}Dept. of Mathematics, Faculty of Science,
Menoufia University, Egypt

Abstract— *Web service composition is a most mature and effective way to realize the rapidly changing requirements of business in service-oriented solutions. Testing the compositions of web services is complex, due to their distributed nature and asynchronous behaviour. Petri Nets (PNs) provide a framework for the construction and analysis of distributed and concurrent systems. This work focuses on the modeling and analysis of web services composition using Higher-Order Petri Nets (HOPNs). The modeling emphasizes on aspects of composition, i.e. a modeling capable of verifying and analyzing the performance of composition at present. The analysis of composition can adopt other approaches, for example Reachability graph.*

Keyword— *Web services composition (WSC), Reachability graph (RG), Petri Nets (PN), Higher-Order Petri Net (HOPN), Open Workflow net (oWF-net).*

I. INTRODUCTION

With the continuous development of computer technology, constructing the loosely coupled collaborative software system, which can flexibly integrates program or legacy system distributing in heterogeneous environment, has become an important research field of software engineering. Service oriented computing (SOC) appear in order to solve above problems. And web service has become one of the most important computing resources on the Internet. Meanwhile, along with its combination standards and technology continuing to develop, web service has been the core technology of SOC [1].

According to application requirements, web service composition may consist of many autonomous web services, thus it may provide new, more powerful and value-added functions.

In order to realize this composition [2], several approaches have been proposed.

- Web service composition based on workflow.

The researchers adopted several languages to implement this approach, namely eFlow.

- Web service composition relied on XML.

The most popular languages that are based on this approach are XLANG and BPEL4WS. Unfortunately, these methods suffer from a total lack of semantic representations of services, which has led researchers to propose another approach based on ontology such as OWL-S. It enables the exchange of information in a meaningful way. However, these approaches cannot verify certain properties such as QoS, accuracy, etc.

- Web service composition based on formal method.

This approach utilizes mathematics model and formal tools to describe web service composition. Petri net is a formal model of concurrency. Process algebra is formal language, which can describe and reason process behaviour.

The authors in [2] presents a modeling the WSC based on Petri nets, they use open workflow nets to model Web services which communicate with other services. Open workflow nets (oWF-nets) result from the extension of workflow nets (WF-nets) by adding external places for the communication of the external environment of the considered Web services.

The classical Petri nets cannot model the behavior of the complex system accurately. To solve this problem, researchers proposed many extensions to the classical Petri nets. Examples are: Higher-Order Petri nets (HOPNs) [3], which use the similarities between neural networks and Petri nets to exploit the properties of higher-order neural networks.

In this paper, we propose a new Higher-Order Petri net called the Higher-Order Open workflow nets (HOWFNs) by adding some amendments to the oWF-nets Model, depending on the definition of Higher-Order Petri net and a Conversion Procedure.

The rest of this paper is organized as follows. Section II is dedicated to present background on Web services composition (WSC), Petri nets, as well as Open workflow net. The definitions proposed of Higher-Order Open Workflow Nets presents in section III. Section IV and V discusses Web Services Composition model with Petri Net and analysis using Reachability Graph, respectively. In section VI presents related work. Conclusions are drawn in Section VII.

II. BACKGROUND

A. Web Services Composition

Web services composition (WSC) consists on the construction of new Web services by composing existing ones [2]. A composite service is a new service with added value obtained by combining other existing services. It can be the allocation of basic services or composite services. Fig.1 shows the obtaining composite service.

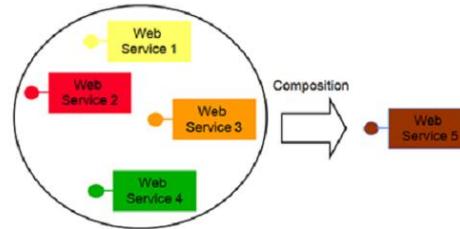


Fig.1 Schematic description of WSC

B. Petri Nets

Petri nets were originally developed to meet the need in specifying process synchronization, asynchronous events, concurrent operations, and conflicts or resource sharing for a variety of industrial automated systems at the discrete-event level. For that reasons, we propose to model WSC using Petri nets.

Definition 1 (Petri Nets): A Petri net is a 4-tuple $PN = (P, T, F, W)$ where

- P and T are two finite non-empty sets of places and transitions respectively, $P \cap T = \phi$,
- $F \subseteq (P \times T) \cup (T \times P)$ is the flow relation, and
- $W: F \rightarrow \mathbb{N}$ is the weight function of N satisfying $W(x, y) = 0 \Leftrightarrow (x, y) \notin F$.

If $W(u) = 1 \forall u \in F$ then PN is said to be ordinary net and it is denoted by $N = (P, T, F)$. For all $x \in P \cup T$, the preset of x is $\bullet x = \{y \mid (y, x) \in F\}$ and the postset of x is $x\bullet = \{y \mid (x, y) \in F\}$. A marking of a Petri net N is a function $M: P \rightarrow \mathbb{N}$. The initial marking of N is denoted by M_0 . A transition $t \in T$ is enabled in a marking M (denoted by $M[t]$) if and only if $\forall p \in \bullet t: M(p) \geq W(p, t)$. If transition t is enabled in marking M , it can be fired, leading to a new marking M' such that: $\forall p \in P: M'(p) = M(p) - W(p, t) + W(t, p)$. The firing is denoted by $M[t]M_0$. The set of all markings reachable from a marking M is denoted by $[M]$. For a place p of P , we denote by M_p the marking given by $M_p(p) = 1$ and $M_p(p') = 0 \forall p' \neq p$.

Petri nets are represented as follows: places are represented by circles, transitions by boxes, the flow relation is represented by drawing an arc between x and y whenever (x, y) is in the relation and the weight function labels the arcs whenever their weights are greater than 1. A marking M of a Petri net is represented by drawing $M(p)$ black tokens into the circle representing the place p .

Definition 2 (Higher-Order Petri Net): An $HOPN$ is formally defined in the same way as the classical Petri nets, $HOPN = (P, T, F, W, M_0)$, where [3]

- $F \subseteq (P \times T) \cup (P^2 \times T) \cup \dots \cup (P^m \times T) \cup (T \times P)$ is a set of arcs;
- $W: F \rightarrow \mathbb{N}$ is a weight function;
- $M_0: P \rightarrow \mathbb{N}$ is the initial token distribution, called the initial marking;

The main difference is the definition of the set of arcs.

Definition 3 (Firing Rule of HOPN): A transition t is said to be enabled or firable if there exist at least one of its k th-order input arcs such that each of this arc's places have at least as many tokens as the weight of this k th-order arc. Such an arc is defined as an enabled arc.

C. Open Workflow Petri Nets (oWF-net)

Open workflow nets [2] (oWF -nets) result from the extension of workflow nets (WF -nets) by adding external places for the communication of the external environment of the considered Web services. Now for WF -nets, they result from the application of Petri nets to workflow management.

Definition 4 (Workflow net, WF-net): A Petri net $PN = (P, T, F, W, M_0)$ is a workflow net if and only if [4]:

- PN has two special places: i and o ; place i is a source place, i.e., $\bullet i = \emptyset$; place o is a sink place, i.e., $o\bullet = \emptyset$,
- If transition t^* would be added to the set of transitions T ($t^* \notin T$) and the arcs (o, t^*) and (t^*, i) would be added to the flow relation F of PN , the resulting net is strongly connected.

Definition 5 (Open Workflow net, oWF-net): An oWF -net is a 6-tuple (P, T, F, IP, OP, F') , where [5]

- $\{P, T, F\}$ is a WFN ;
- IP is the set of input message places, and $\forall z \in IP, \bullet z = \phi$;
- OP is the set of output message places, and $\forall z \in OP, z\bullet = \phi, IP \cap OP = \phi$;
- $F' \subseteq (T \times OP) \cup (IP \times T)$ is the set of interface arcs.

III. THE PROPOSED DEFINITIONS OF HOTPN

Definition 6 (Higher-Order Open Workflow Net, HOWFN): An $HOWFN = (P, T, F, IP, OP, F')$ is called a *Higher-order Open Workflow Net (HOWFN)*, if

- $F \subseteq (P \times T) \cup (P^2 \times T) \cup \dots \cup (P^m \times T) \cup (T \times P)$ is a set of arcs;
- $\{P, T, F, IP, OP, F'\}$ is an oWF-Net,

Definition 7 (Firing Rule of HOWFN): A transition t is said to be enabled or firable if there exist at least one of its k th-order input arcs such that each of this arc's places have at least as many tokens as the weight of this k th-order arc. Such an arc is defined as an enabled arc.

Definition 8 (Conversion Procedure): This procedure consists of two phases:

First phase: we must ensure the validity of all of the following conditions:

- First: If there exist at least two of transitions that have the same output place(s).
- Second: If there exist one of the previous transitions that have at least two of the incoming arcs.
- Third: If all previous transitions have the same type (timed or immediate transitions).

Second phase: we apply the following steps:

1. For each transition contain at least two of incoming arcs. An arcs become a single arc, as in the follow figure:



Fig.2 The process of integrating arcs

2. Merge all transitions to a single transition, as in Fig.3.

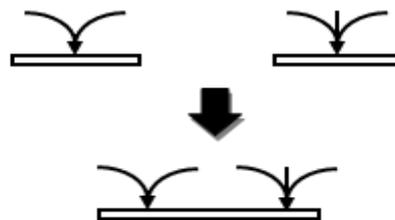


Fig.3 The process of integrating transitions

After modeled by a $HOWFN$, web service composition is analyzed and verified by behaviour properties (or called dynamic properties).

Definition 9 (Reachability Property): Let $(HOWFN, M_0)$ Higher-Order Open Workflow Net, fire sequence $\sigma = t_1, t_2, \dots, t_n$, the sequence of states and transitions $M_0 t_1 M_1 t_2 \dots t_n M_n$ and from $(HOWFN, M_0)$ a set of all possible fire sequence $L(HOWFN, M_0)$ or $L(M_0)$ be given. The $HOWFN$ model is reachable iff:

$$\exists \sigma_i \in L(M_0) \Rightarrow M_0 [\sigma > M_n,$$

Definition 10 (safeness Property): A place of a $HOWFN$ is safe if the number of tokens in it never exceeds one for any reachable marking of the $HOWFN$. The $HOWFN$ is safe if all its places are safe. i.e., iff:

$$\forall p \in P, M_i \in R(M_0) \Rightarrow M(p) \leq 1.$$

Definition 11 (Boundedness Property): A place of a $HOWFN$ is k -bounded if the number of tokens in it never exceeds $k \geq 1$ ($k \in N$). The $HOWFN$ is k -bounded if all its places are k -bounded. i. e., iff:

$$\forall p \in P, M_i \in R(M_0) \Rightarrow M(p) \leq k.$$

Then the bound k for the $HOWFN$ can be defined as the maximum bound over all its places.

Definition 12 (Liveness Property): $(HOWFN, M_0)$ is live; there is no transition that cannot fire by proper transition sequence. i.e., iff:

$$\forall M_i \in R(M_0) \Rightarrow M_i(p) \neq \omega, M_x [t > M_y, M_y [t > M_x.$$

ω is the number of tokens presented at certain place is infinite.

IV. MODELING WEB SERVICES COMPOSITION WITH PETRI NET

Petri Net is one of tools to model discrete event systems [6]. It contains transition related events. In order that an event takes place, some conditions are to be first satisfied. Each of the information about the events and conditions are represented in transitions and places, respectively. The workflow process modeling with Petri Net approach is designed in three ways, i.e. 1) the processes are modeled with transitions, 2) the conditions are modeled with places and 3) cases are modeled with tokens.

- Model is Boundedness and safe; marking count is always limited under 1.
 $\forall p \in P, M_i \in R(M_0) \Rightarrow M(p) \leq k, k=1$
- Model is Reachability, i.e., all places are passable to transit from the initial place.
 $\exists \sigma_i \in L(M_0) \Rightarrow M_0 [\sigma > M_n,$
- Model is not liveness; the initial place never comes back, i. e., with a deadlock.
 $\neg[\forall M_i \in R(M_0) \Rightarrow M_i(p) \neq \omega, M_x [t > M_y, M_y [t > M_x]$

TABLE I
TRANSITIONS FOR WS₁ (CLIENT SERVICE)

Transitions	Meaning
t _{1,1}	Enter id of the client
t _{1,2}	Request credit
t _{1,3}	Enter guarantee
t _{1,4}	Enter balance
t _{1,5}	Accept the request after validation
t _{1,6}	Receive update Message
t _{1,7}	Receive offer
t _{1,8}	Accept the request
t _{1,9}	Accept the request
t _{HO}	Receive Reject Message

TABLE II
TRANSITIONS FOR WS₂ (PORTAL SERVICE)

Transitions	Meaning
t _{2,1}	Valid authentication
t _{2,2}	Generate session id
t _{2,3}	Ask the client to re-enter id

TABLE III
TRANSITIONS FOR WS₃ (VALIDATION SERVICE)

Transitions	Meaning
t _{3,1}	Check pay
t _{3,2}	Put away Requests
t _{3,3}	Ask the client to re-enter

TABLE IV
TRANSITIONS FOR WS₄ (REQUESTS PROCESSING SERVICE)

Transitions	Meaning
t _{4,1}	Put Request task list
t _{4,2}	Withdraw request
t _{4,3}	Evaluate (Public part)
t _{4,4}	Evaluate (Private part)
t _{4,5}	Update the application
t _{4,6}	Reject the request
t _{4,7}	Send update message

t _{4,8}	Send reject message
t _{4,9}	Accept the request
t _{4,10}	Save client information
t _{4,11}	Save insurance information
t _{4,12}	Update contract
t _{4,13}	evaluate contract
t _{4,14}	Send offer to the client
t _{4,15}	Request treated

TABLE V
INTERFACE PLACES BETWEEN WEB SERVICERS

Interface Places	Meaning
ip ₁	id entered
ip ₂	id OK
ip ₃	id not OK
ip ₄	Pay balance
ip ₅	Valid request
ip ₆	Add request
ip ₇	Message sent to update
ip ₈	Message sent to reject
ip ₉	Offer sent
ip ₁₀	Message sent to reject

VI. RELATED WORK

In order to fast computation, many researchers prefer Petri nets, 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.

In [2], Guerfel, et. al. develop a tool for formal verification of Web services composition. These services are modeled by oWF-nets composed of a set of workflow nets interconnected via interface places. These places ensure the communication between the different services. This model is generated to an SMV code that is sent to NuSMV Model Checker. NuSMV verifies certain properties formulated in CTL such as soundness properties.

In [5], Yanhua, Xitong and PengCheng present a Petri net approach to mediation-aided composition of Web services. First, services are modeled as open Workflow Nets (oWFNs) and are composed using mediation transitions (MTs). Second, the modular reachability graph (MRG) of composition is automatically constructed and the compatibility is analyzed.

In [8-10] Lohmann et al. adopt open workflow nets (oWFN) for modeling BPEL processes and automatically analyze the interactional behavior of a given oWFN.

In [11], P. Massuthe, W. Reisig, and K. Schmidt propose oWFN as a formal model for services that use workflows as their internal control structure. They characterize well-behaving pairs of requester's and provider's services and suggest operating guidelines as a convenient and intuitive artifact to realize publish. Then, the find operation reduces to a matching problem between the requester's service and the operating guideline. Binding of a requester's and a provider's service are therefore guaranteed to result in a well-behaving cooperating service.

In [12] K. Schmidt propose adequate model of Open workflow nets for services or parts of inter-organizational business processes. He investigates the problem of controllability, a natural counterpart of soundness in classical workflow nets (as studied by van der Aalst). He distinguishes centralized, distributed, and local controllability and provides solutions to all problems.

VII. CONCLUSION

We proposed in this paper a tool for modeling and analysis of Web services composition. First, these services are modeled by oWF-nets composed of a set of workflow nets interconnected via interface places. Then, web services composition is modeled by HOWFN.

This model reduces the number of transitions to one transition using Conversion Procedure, while maintaining the work of Model as is.

Finally, we analyze the behavioral properties of the model using reachability graph. Where model is Boundedness, reachability and not-liveness (The lack of repetition), i. e., the model is deadlock.

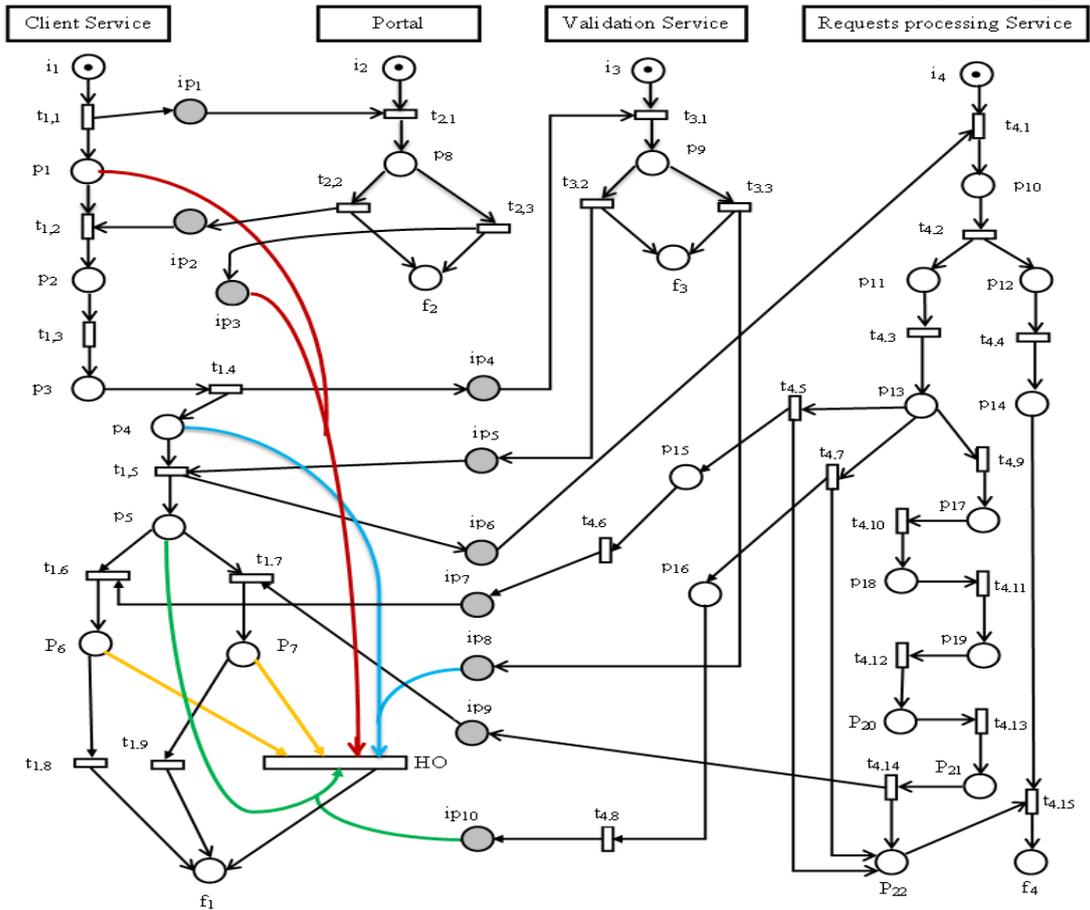


Fig.4 A Higher-Order Workflow Net model of the credit management

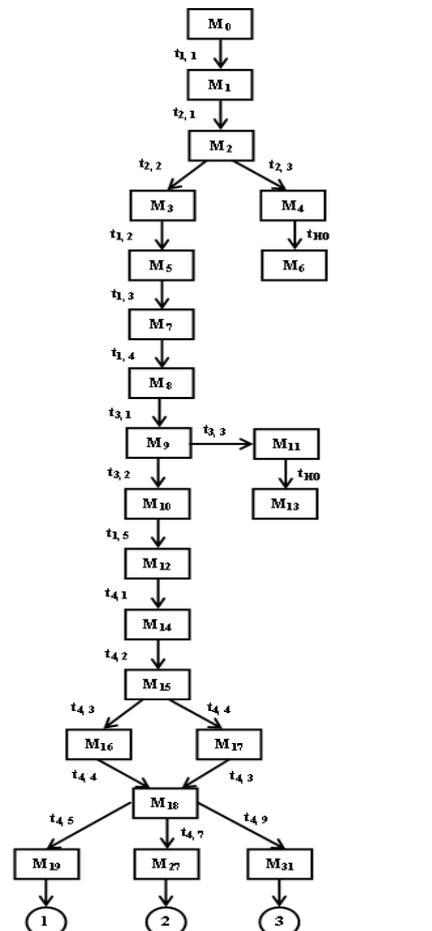


Fig.5 the Reachability Graph of HOWFN

REFERENCE

- [1] B. Tian, "The Performance Analysis of Web Service Composition Based on Timed Colored Petri Net", *Journal of Software*, Vol. 8, No. 12, pp. 3152- 3157, Dec. 2013.
- [2] R. Guerfelm, Z. Sba, and K. Barkaoui, "Modeling and formal verification framework of Web services composition", *International Conference on Control, engineering & information technology (CEIT'13), proceedings engineering & technology*, vol.2, pp. 140-145, 2013.
- [3] T. Chow, and J-Y Li, "Higher-Order Petri Net Models Based on Artificial Neural Networks", *Artificial Intelligence*, vol. 92, pp. 289-300, 1997.
- [4] H. A. Reijers, "Design and Control of Workflow Processes", *Springer-Verlag Berlin Heidelberg*, 2003.
- [5] Y. Du, X. Li, and P.C. Xiong, "A Petri Net Approach to Mediation-Aided Composition of Web Services", *IEEE Transactions On Automation Science and Engineering*, vol. 9, No. 2, pp. 429-435, Apr. 2012.
- [6] R. Sarno, B. A. Sanjoyo, I. Mukhlash, and H. M. Astuti, "Petri Net Model of ERP Business Process Variation for Small and Medium Enterprises", *Journal of Theoretical and Applied Information Technology*, vol. 54 No.1, pp. 31-38, Aug. 2013.
- [7] N. Luo, J. Yan, M. Liu, and S. Yang, "Towards Context-Aware Composition of Web Services", *IEEE Computer Society, International Conference on Grid and Cooperative Computing (GCC'06)*, 2006.
- [8] N. Lohmann, "A feature-complete Petri net semantics for WS-BPEL 2.0", *Proc. of the 4th International Workshop on Web Services and Formal Methods, Brisbane, Australia*, pp. 77-91, Sept. 2007.
- [9] N. Lohmann, P. Massuthe, C. Stahl, and D. einberg, "Analyzing Interacting BPEL Processes", *Proc. of the 4th International Conference on Business Process Management, Vienna, Austria*, pp. 17-32, Sept. 2006.
- [10] D. König, N. Lohmann, S. Moser, C. Stahl, and K. Wolf, "Extending the Compatibility Notion for Abstract WS-BPEL Processes", *Proc. of the 17th International World Wide Web Conference, Beijing, China*, pp. 785-794, Apr. 2008.
- [11] P. Massuthe, W. Reisig, and K. Schmidt, "An Operating Guideline Approach to the SOA", *Annals of Mathematics, Computing & Teleinformatics*, vol. 1(3):pp.35-43, 2005.
- [12] K. Schmidt, "Controllability of open workflow nets", In: *Desel, J., Frank, U. (eds.) Enterprise Modelling and Information Systems Architectures, Proceedings of the Workshop in Klagenfurt, October 24-25, 2005. Lecture Notes in Informatics (LNI), GI*, vol. 75, pp. 236-249 2005.

