

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/224108192>

A Petri Net Approach to Analysis and Composition of Web Services

Article in IEEE Transactions on Systems Man and Cybernetics - Part A Systems and Humans · April 2010

Impact Factor: 2.18 · DOI: 10.1109/TSMCA.2009.2037018 · Source: IEEE Xplore

CITATIONS

79

READS

43

3 authors, including:



[Pengcheng Xiong](#)

NEC Corporation

34 PUBLICATIONS 534 CITATIONS

[SEE PROFILE](#)



[Mengchu Zhou](#)

New Jersey Institute of Technology

688 PUBLICATIONS 15,895 CITATIONS

[SEE PROFILE](#)

A Petri Net Approach to Analysis and Composition of Web Services

PengCheng Xiong, YuShun Fan, and MengChu Zhou, *Fellow, IEEE*

Abstract—Business Process Execution Language for Web Services (BPEL) is becoming the industrial standard for modeling web service-based business processes. Behavioral compatibility for web service composition is one of the most important topics. The commonly used reachability exploration method focuses on verifying deadlock-freeness. When this property is violated, the states and traces in the reachability graph only give clues to re-design the composition. The process must then repeat itself until no deadlock is found. In this paper, multiple web services interaction is modeled with a Petri net called Composition net (C-net for short). The problem of behavioral compatibility among web services is hence transformed into the deadlock structure problem of a C-net. If services are incompatible, a policy based on appending additional information channels is proposed. It is proved that it can offer a good solution that can be mapped back into the BPEL models automatically.

Index Terms— Business Process Execution Language for Web Services, business process, discrete event systems, Petri nets, Web service, siphons

I. INTRODUCTION

SERVICE oriented architecture (SOA) [1-2] has evolved to become a promising technology for the integration of disparate software components using Internet protocols. Web Services Description Language (WSDL), Universal Description, Discovery, and Integration (UDDI) and Simple Object Access Protocol (SOAP) are three core standards for web service description, registration and binding respectively. After a provider offers a service by publishing its description at UDDI, a requestor queries UDDI in order to discover a suitable service to perform a certain task.

Most of the time, multiple web services need to invoke each

other dynamically to accomplish service requestor's manifold requirements. Orchestration and choreography [3] are two different viewpoints for web service interaction. The former considers one particular service that directs the logical order of all other services while the latter considers the case where individual services work together in a loosely coupled network.

Hull and Su [4] give an overview of the current work on service composition. Business Process Execution Language for Web Services (BPEL) [5] represents a merge of the web services flow language of IBM and the extensible language of Microsoft. It is the most widely used language for web service orchestration and choreography. A BPEL model has three different kinds of activities [6-8]. Basic activities perform simple operations. Structured ones consist of basic activities that impose a certain execution sequence. Others deal with exceptions and alarms. The web service interactions are specified by BPEL. Although the specification, composition, and execution of BPEL processes are well supported by various vendors, the analysis of those models, especially compatibility is still an open problem.

There are mainly two compatibility considerations [9-10]. First, syntactically, web services can be composed only if the provided interfaces specified by WSDL with port types, operations, and message types match the required interfaces of the other web services. The second one is behavioral including equivalence and usability. Two web services are equivalent if one service can be replaced by another while the remaining components stay unchanged. Web services are usable if the interaction among them guarantees compatibility. Since the interaction of BPEL processes mirrors the interaction of web services, we do not distinguish them in the rest of the paper.

This paper proposes a Petri net-based analysis technique and corresponding policy to deal with behavioral incompatibility for the implementation purpose. The rest of the paper is organized as follows: Section II reviews the current research work on BPEL processes from both syntactic and behavioral perspectives. Section III proposes a Petri net called C-net to model multiple web service interaction. Section IV performs the structural analysis of C-nets and proposes a policy based on appending additional information channels for an incompatible C-net. Section V describes and compares different implementations for the compatibility of a C-net. Section VI validates the analysis results and solution by a real example. Finally, Section VII concludes the paper.

Manuscript received Jan. 2008. This work was supported in part by the National High Technology Research and Development (863) Program of China under Grant 2006AA04Z151, the China National Science Foundation under Grant 60674080, the National Basic Research Program of China (973 Program) under Grant 2006CB705407, and Chang Jiang Scholars Program of PRC Ministry of Education.

P.C. Xiong is with College of Computing, Georgia Institute of Technology Atlanta, GA 30332 USA (e-mail: xiong@gatech.edu). Most of the work was done when this author was with Tsinghua University.

Y.S. Fan is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: fanyus@tsinghua.edu.cn)

M.C. Zhou is with Department of ECE, New Jersey Institute of Technology Newark, NJ 07102-1982 USA and is also with School of Electro-Mechanical Engineering, Xidian University, Xi'An, Shanxi 710071, China (e-mail: zhou@njit.edu).

II. RELATED WORK

A. Syntactic requirement

The syntactical compatibility refers to the conformance of signatures between the two web services. For instance, if one web service invokes an operation of the second one, it is necessary that the parameters, and their number and type of the first and second ones match. Data-driven based methods are often adopted for the syntactic consideration. Data mapping is to relate equivalent data elements of two messages so that two interfaces that belong to different services can be linked. Two services can be composed by specifying the output message of one interface as the mirror image of the input message of another one. In order to solve the problems of integrating data models and message formats, Extensible Stylesheet Transformations (XSLT) can be used. Through constructing an adapter or mediator on either request-side or service-side based on XSLT, after message transformation is performed on either side, the gap between cooperating data structures in different web services is reduced or eliminated. Nezhad *et al.* propose a solution for mismatches between service interfaces based on schema matching [11]. They first identify the relationships between messages in partner web services and then specify mapping functions. Our previous work [12] proposes a method to automatically check and mediate the messages sent and received by two web service parties during the interaction. Moreover, web service choreography interface and web service choreography description language also provide simple mechanisms at the global model level for connection and adaptation. Although the above studies consider whether web services can understand each other in terms of message syntax and form, the assumption that the interaction logic among web services can coincide may not hold in practice.

B. Behavioral requirement

Merely satisfying syntactic requirement is not enough for successful interaction. The following motivating scenario comes from the composition of a customer and an online shop service [13]. Although it is a simple excerpt of a real business scenario, it is illustrative enough to explain the proposed concept, algorithm and result.

In Fig. 1, a customer waits for a product before paying the bill while the online shop waits for payment before sending it. Suppose that two parties are compatible in invoking interfaces for delivery and payment information and the message structure for the interfaces is also the same, e.g., delivery date is of a time type and product number is of an integer type. However, the interaction between them leads to a deadlock. In Fig. 2, although the two parties agree on Pay-on-Delivery, web service for the online shop cannot terminate appropriately when a customer chooses to pay by VISA while the online shop waits to be paid by eCash. This kind of problems is also classified as a non-local choice problem [14]. Thus the behavior of web services must be taken into account when analyzing the compatibility of web services.

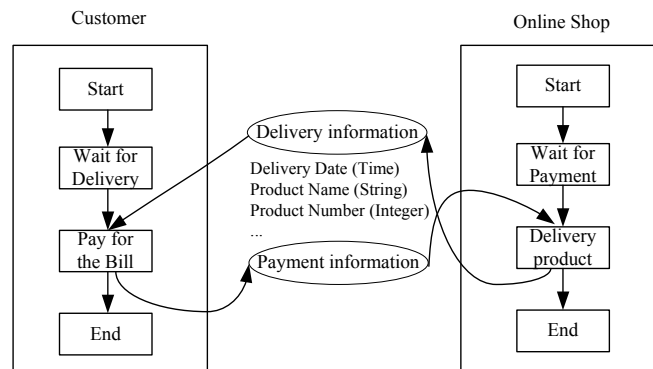


Fig. 1. Illustration of an incompatible case

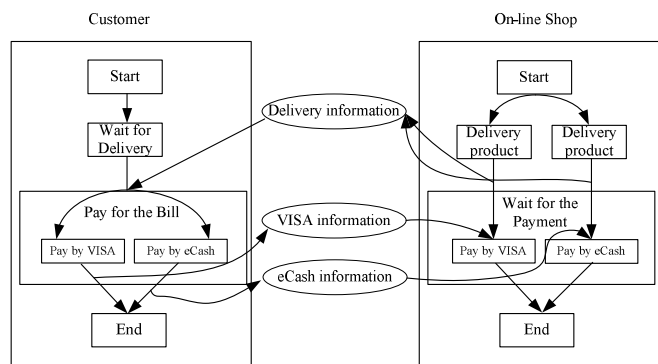


Fig. 2. Illustration of an incompatible case due to a non-local choice problem

Since the textual specification of BPEL is not suitable for computer aided verification for behavior incompatibility, current methods mainly follow two steps, i.e., modeling and analysis. There is a plethora of modeling methods, e.g., abstract state machines [19], finite state machines [17, 20], process algebra and pi-calculus based approaches [21]. Bordeaux *et al.* [22] use labelled transition system to formalize the behavior of web services. Wohed *et al.* [15] propose an analysis of BPEL in terms of a set of workflow patterns. By extracting the interaction message, Foster *et al.* [16] propose a model-based approach to verify web service composition by using message sequence charts. Heckel and Mariani [18] propose a test-based method to check whether a web service with respect to a set of graph transformation rules can satisfy the interaction operations.

As an appropriate method for modeling and analyzing distributed business processes [23], Petri nets are also an adequate modeling tool for web service behavior. As shown in [24], they are able to define and verify usability, compatibility and equivalence of web services. In particular, their semantics for BPEL are proposed [24-32]. Since BPEL is becoming the industrial standard for modeling web service based business processes, a Petri net based method is thus directly applicable to real world examples. Almost all of the above formalisms cover basic and structured activities of BPEL, while only some of them, e.g., [24, 30] cover event and exception handling. A brief comparison is as follows.

Ouyang *et al.* [25] transform BPEL into Petri nets represented in the Petri nets markup language and perform

WofBPEL to support three types of analysis, i.e., reachability analysis, competing message-consuming activities and garbage collection of queued messages by generating the full state space. They adopt Petri net reduction rules to reduce a model before generating its state space [26]. Lohmann *et al.* adopt open workflow nets (oWFN)[27-29] for modeling BPEL processes and automatically analyze the interactional behavior of a given oWFN. Hinz *et al.* propose a parser to automatically transform a given BPEL process into a Petri net [30] and perform model checking.

Martens [31-32] proposes a BPEL annotated Petri nets (BPN) and presents a decision algorithm for the controllability of a BPN model based on a communication graph (c-graph). A c-graph is a directed, bipartite graph in which nodes denote reachable states of the BPN and edges denote messages that the BPN is able to send or receive. Martens transforms the check of interaction between the composed BPEL processes into the verification of deadlock-freeness of a BPN. After all parts that yield a deadlock are cut off, the remaining part of BPN is proven to be controllable. As shown in Fig. 3, Martens' system starts from a given set of BPEL processes. The next steps are: (1) Transform each BPEL process into a BPN. (2) Compose all BPN models and reduce them to speed up analysis. (3) Generate and analyze the c-graph of the composed BPN. (4) Based on the related states and traces in case of incompatibility, re-design the composition.

König *et al.* [29] point out that the BPEL specification distinguishes two kinds of business processes, i.e., executable and abstract. Most of the previous work [24-32] focuses on abstract ones. Their work permits the occurrence of incompatible cases. When it happens, it is detected by the analysis method based on the Petri net-based model, and then, two BPEL profiles for composition re-design can be applied, i.e., the abstract process profile for observable behavior from bottom-up viewpoint and the abstract process one for templates from top-down viewpoint. Their main idea is to substitute an erroneous service with a correct one. The efficiency of this "substitute" based approach depends upon how to find out the exactly right web service from thousands of candidate web services with the exact behavior that conforms to the other web services. In general, these methods may become complex, considering that a single error of a certain web service may collapse the compatibility of the whole BPEL process.

The detail of our proposed system flow is shown in Fig. 4. We address the web service compatibility issues by concentrating on a) modeling and definition of the compatibility, b) compatibility analysis, and c) a policy for resolving incompatibility. We model the BPEL in terms of Petri nets called workflow module net (WMN) and Composition net (C-net). These models are different from the existing Petri net ones since we can characterize the compatibility in terms of structural Petri net objects, i.e., siphons. Finally, we formulate a policy by appending additional possible information channels in terms of Petri net elements.

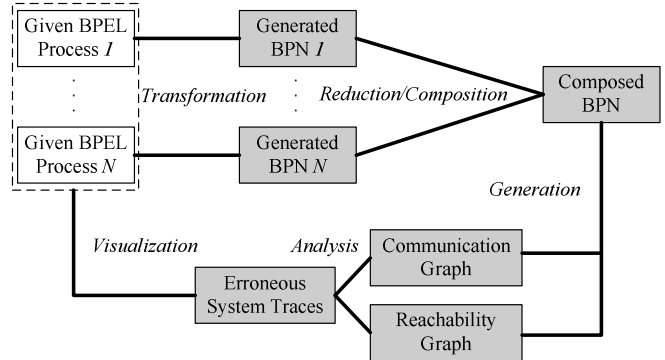


Fig. 3. Illustration of the flow through Martens' system [31]

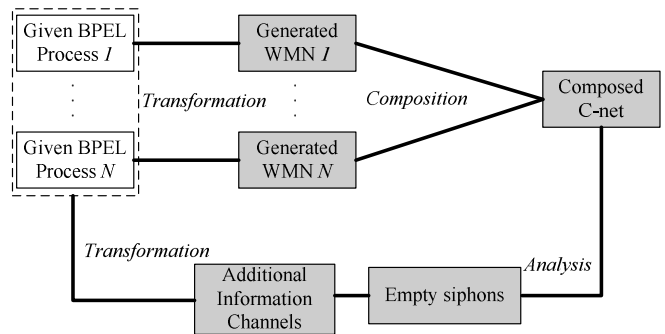


Fig. 4. Illustration of the flow through the proposed system

The main contribution of our paper is that as we analyze the model through structural properties instead of the reachable states, we can impose constraints on the model to prevent it from reaching incompatible states. Since we do not need to search other replaceable web services and the new model can be transformed back to code in BPEL in order to guarantee the compatibility automatically, this modification based approach is more efficient than the existing "substitute" based ones.

Although the compatibility problem for web service interaction is similar in some way to the traditional deadlock problem in flexible manufacturing systems (FMS) [33-34], they are not totally the same. Firstly, the pre-conditions that must be held for a deadlock to occur are totally different. Four conditions [35], i.e., mutual exclusion, no preemption, wait for condition and circular wait, must be held for a deadlock in FMS to occur. However, the web service incompatibility is mainly attributed to the erroneous web service behaviors. Secondly, since deadlock in FMS is caused by an inappropriate allocation of resources to concurrently executing processes, the deadlock problem can be solved in many cases by increasing the number of shared resources. Web services can be made to be compatible by adding information channels, or they are just incompatible at all. Thirdly, compared with the relatively fixed working processes in FMS, the routes in BPEL are much more flexible. There are a large number of choices of services with different behaviors that can provide the same syntactic interfaces.

III. MODELING WEB SERVICES INTERACTION WITH PETRI NETS

We assume that a reader is familiar with basic Petri nets [33-37]. The following definition is used.

Definition 1: A Petri net is a 3-tuple, $N = (P, T, F)$ where:

- i. $P = \{p_1, p_2, \dots, p_m\}$, $m > 0$, is a finite set of places pictured by circles;
- ii. $T = \{t_1, t_2, \dots, t_n\}$, $n > 0$, is a finite set of transitions pictured by bars, with $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$;
- iii. $F \subseteq (P \times T) \cup (T \times P)$ is the incidence relation. Based on F , we can derive the input and output functions which are $m \times n$ matrices. $I: P \times T \rightarrow \{0, 1\}$ is an input function. $I(p, t)=1$ if $(p, t) \in F$; and 0 otherwise. $O: P \times T \rightarrow \{0, 1\}$ is an output function. $O(p, t)=1$ if $(t, p) \in F$; and 0 otherwise.

Postset of t is the set of output places of t , i.e., $t^\bullet = \{p \mid O(p, t) \neq 0\}$. Preset of t is the set of input places of t , i.e., ${}^\bullet t = \{p \mid I(p, t) \neq 0\}$. Post (Pre) set of p is the set of output (input) transitions of p , denoted by p^\bullet and ${}^\bullet p$ respectively. $|{}^\bullet t|$ and $|t^\bullet|$ are defined as the number of input and output places, respectively.

Given $Q \subseteq P$, ${}^\bullet Q = \bigcup_{p \in Q} {}^\bullet p$ and $Q^\bullet = \bigcup_{p \in Q} p^\bullet$. ${}^\bullet p = \bigcup_{t \in p^\bullet} t$

and ${}^\bullet p = \bigcup_{t \in {}^\bullet p} t$.

$M: P \rightarrow \mathbb{Z}^+$, is a marking where $M(p)$ represents the number of tokens in place p and $\mathbb{Z}^+ = \{0, 1, 2, \dots\}$. An initial marking is denoted by M_0 . Tokens are pictured by dots. (N, M) is called a net system or marked net. p is marked by M iff $M(p) > 0$. A place subset $S \subseteq P$ is marked by M iff at least one place in S is marked. The sum of tokens in all places in S is denoted by $M(S) = \sum_{p \in S} M(p)$. A transition $t \in T$ is enabled under M , if and only if $\forall p \in {}^\bullet t: M(p) > 0$, denoted as $M[t >]$. If $M[t >]$ holds, t may fire, resulting in a new marking M' , denoted as $M[t > M']$, such that $M'(p) = M(p) - 1$ if $\forall p \in {}^\bullet t \setminus t^\bullet$; $M'(p) = M(p) + 1$ if $\forall p \in t^\bullet \setminus {}^\bullet t$; and otherwise $M'(p) = M(p)$.

M' is reachable from M iff there exists a firing sequence $\sigma = t_1 t_2 \dots t_n$ such that $M[t_1 > M_1][t_2 > \dots M_{n-1}][t_n > M']$ holds. The set of markings reachable from M_0 in N is denoted as $R(N, M_0)$. Given a marked net (N, M_0) , $t \in T$ is live under M_0 iff $\forall M \in R(N, M_0)$, $\exists M' \in R(N, M)$, $\ni M[t >]$ holds. N is dead under M_0 iff $\forall t \in T$, $M_0[t >]$ cannot hold. (N, M_0) is live iff $\forall t \in T: t$ is live under M_0 .

$N_X = (P_X, T_X, F_X)$ is a subnet of the Petri net $N = (P, T, F)$ iff $P_X \subseteq P$, $T_X \subseteq T$ and $F_X = F \cap \{(P_X \times T_X) \cup (P_X \times T_X)\}$. N_X is generated by P_X iff $T_X = {}^\bullet P_X \cup P_X^\bullet$. Note that. Petri net is called a state machine iff $\forall t \in T$, $|{}^\bullet t| = |t^\bullet| = 1$.

A P-vector is a column vector $X: P \rightarrow \mathbb{Z}$ indexed by P and a T-vector is a column vector $Y: T \rightarrow \mathbb{Z}$ indexed by T , where \mathbb{Z} is the set of integers. The incidence matrix of N is a matrix $[N] = O - I$. X is a P-invariant (place invariant) iff $X \neq 0$ and $X^T[N] = \mathbf{0}^T$. Y is a T-invariant (transition invariant) iff $Y \neq 0$ and $[N]Y = \mathbf{0}$. $\|X\| = \{p \in P \mid X(p) \neq 0\}$ and $\|Y\| = \{t \in T \mid Y(t) \neq 0\}$ are called the support of X and Y respectively.

A nonempty place set $S \subseteq P$ is called a siphon iff ${}^\bullet S \subseteq S^\bullet$. A siphon is minimal iff there does not exist a siphon $S' \subset S$. If a minimal siphon does not contain the support of any P-invariant of a net N , it is called strict minimal siphon (SMS). Suppose that $S \subseteq P$ is a siphon of $N = (P, T, F)$ and X is a P-invariant. Siphon S is controlled by X under M_0 iff $X^T M_0 > 0$ and $\{p \in P \mid X(p) > 0\} \subseteq S$. S is also called an invariant-controlled siphon.

Definition 2: A simple sequential workflow net (SSN) is a Petri net $N = \{P \cup \{\alpha, \beta\}, T, F\}$ iff:

- i. N has two special places: α and β where α is a source place ${}^\bullet \alpha = \emptyset$ and β is a sink place $\beta^\bullet = \emptyset$.
- ii. If we add a new transition t to N which connects α with β , i.e., ${}^\bullet t = \alpha$ and $t^\bullet = \beta$, then the resulting extended net $\bar{N} = (\bar{P}, \bar{T}, \bar{F})$ where $\bar{P} = P$, $\bar{T} = T \cup \{t\}$, $\bar{F} = F \cup \{(\beta, t), (t, \alpha)\}$, is a strongly connected state machine.
- iii. Every circuit of N contains t .
- iv. $M_0(\alpha) = 1$ and $M_0(p) = 0$, $\forall p \neq \alpha$

State α is defined as $M(\alpha) = 1$ and $M(p) = 0$, $\forall p \in P \setminus \{\alpha\}$ while state β is defined as $M(\beta) = 1$ and $M(p) = 0$, $\forall p \in P \setminus \{\beta\}$. Note that SSN is a special kind of workflow net [23] and is close to the definition of simple sequential processes in [38]. It can be verified that the conditions ii)-iv) in Definition 2 impose a sound property to all SSNs.

- i. For every state M reachable from state α , there exists a firing sequence leading from M to β .
- ii. State β is the only state reachable from α with at least one token in β .
- iii. There are no dead transitions in it.

Definition 3: A workflow module net (WMN) is an extended Petri net $N = \{P \cup P_I \cup P_O, T, F\}$, where:

- i. The subnet generated by P is a sound SSN.

- ii. P_I and P_O denote the input and output interfaces for the workflow module respectively. $P_I \neq \emptyset$, $P_O \neq \emptyset$, $P_I \cap P_O = \emptyset$ and $(P_I \cup P_O) \cap P = \emptyset$.
- iii. The following statements are true
 - a) $\forall p \in P_I, p^{\bullet} \cap P \neq \emptyset$
 - b) $\forall p \in P_O, P \cap p^{\bullet} \neq \emptyset$
 - c) $\forall p \in (P_I \cup P_O), p \cap P^{\bullet} = \emptyset$
- iv. $M_0(\alpha) = 1$ and $M_0(p) = 0, \forall p \neq \alpha$

We define P as the set of process places and $P_I \cup P_O$ as the set of interface places. The initial marking of a WMN follows that of its SSN according to condition iv.

Definition 4: Two WMN $N_j = \{P_j \cup P_{I_j} \cup P_{O_j}, T_j, F_j\}$, $j \in \{1, 2\}$ are composable, iff $P_1 \cap P_2 = \emptyset$, $T_1 \cap T_2 = \emptyset$ and $(P_{I_1} \cap P_{O_2}) \cup (P_{I_2} \cap P_{O_1}) = P_E \neq \emptyset$. For every $p \in P_E$, if $x \in p^{\bullet}$ and $y \in p^{\bullet}$, we call (x, p, y) an information channel. The 2-member Composition net (C-net) denoted as $N = N_1 \odot N_2$ is defined as follows:

- i. $P = P_1 \cup P_2$,
- ii. $P_I = \{(P_{I_1} \cup P_{I_2}) \setminus P_E\}$, $P_O = \{(P_{O_1} \cup P_{O_2}) \setminus P_E\}$,
- iii. $T = T_1 \cup T_2$, and
- iv. $F = F_1 \cup F_2$.

Two WMN N_1 and N_2 are composable when they interact through a set of common places. For example, if $P_{I_2} \cap P_{O_1} = P' \neq \emptyset$, N_1 sends information through the set of interfaces P' that is received by N_2 .

We then show how we model web service interaction with WMNs. We assume that the compatibility at the syntactic level is satisfied, i.e., the input and output interfaces have the same port types, operations, and message types. We divide the basic structures in BPEL, i.e., *receive*, *reply*, *invoke*, *assign*, *throw*, *terminate*, *wait* and *empty* into two categories. The first category is internal behavior that includes *assign*, *throw*, *terminate*, *wait* and *empty*. The second category is external behavior that includes *receive*, *reply* and *invoke*. Basic structures in the first category are not related to the interaction between different web services and we model them as a single transition in WMN. Basic structures in the second category are related to the interaction between different web services and we model them as transitions connected with interface places as shown in Fig. 5.

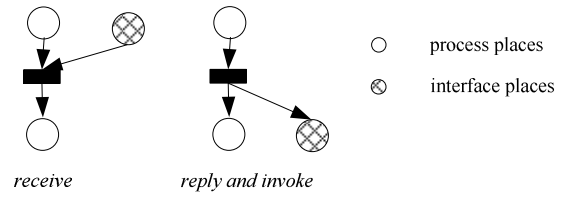


Fig. 5. Modeling of *receive*, *reply* and *invoke* basic structures in BPEL with Petri nets

There are *sequence*, *flow*, *pick*, *switch* and *while* structured activities in a BPEL process. Based on basic structures, a WMN can cover the *sequence*, *pick* and *switch* structures. The semantics of *while* structure are similar to while-loop in programming languages like Java. Here we approximate the number of loops in a finite *while* structured activity and transform the activity to a *sequence* activity by expanding cycles [38]. We can transform the processes that are executed in parallel in the *flow* structure into the same processes that are invoked simultaneously in the *invoke* structure while maintaining the business logic. For example, we can divide the processes that are executed in parallel into separate BPEL processes while maintaining the business logic as shown in Fig. 6.

Note that according to Definition 3, the interface places do not have tokens at the initial marking. According to Definition 4, the interface places can have tokens if and only if a certain transition wants to send a message through the information channel while the interface places can lose token if and only if a certain transition wants to receive a message through the information channel. A token in interface places models the situation when the required message is ready. Since we use non-colored Petri nets to model C-nets, we assume that the maximum number of tokens that an interface place can hold is one. Otherwise there should be duplicate information. Thus, if there exists situation when more than one token are in an interface place, the BPEL process is not acceptable and the developer should check the correctness of the BPEL process. Note that, if we adopt colored Petri nets to model C-nets where we use different color tokens to denote different instances, the maximum number of tokens that an interface place can hold can exceed one.

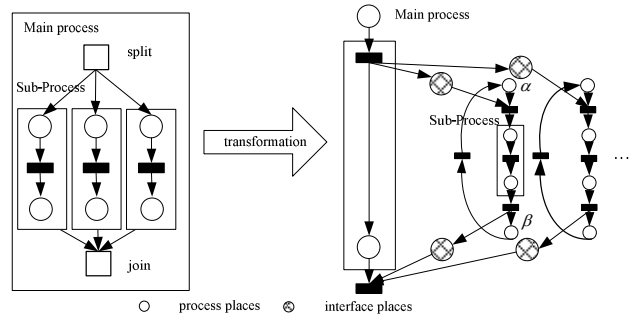


Fig. 6. Transform *flow* structure to several *invoke* structures

Definition 5: An n -member C-net denoted as $N = \odot_{i=1}^n N_i$ is defined recursively following Definition 4. A C-net is a complete net iff $P_I = \emptyset$ and $P_O = \emptyset$.

A WMN can be considered as a single-member C-net. A complete C-net does not have additional input and output interfaces for external interaction. It is easy to prove that $N_1 \odot N_2 = N_2 \odot N_1$ and $(N_1 \odot N_2) \odot N_3 = N_1 \odot (N_2 \odot N_3)$. Note that in some situations, Definitions 4 and 5 are too restrictive. Two C-nets may interact satisfactorily even when one has output interfaces while the other one does not intend to input. Bordeaux *et al.* [22] define this kind of interfaces as unspecified receptions and discuss the three kinds of relationship between the input and output interfaces, i.e., equal, unspecified receptions and unspecified emissions. In our paper, we assume that there are neither unspecified receptions nor unspecified emissions in a complete C-net.

Definition 6: Suppose that C-net $N = N_1 \odot N_2$ is complete, where N_1 and N_2 are incomplete C-net. Then N_1 is called environment of N_2 and vice versa.

According to the web service description, registration and binding mechanisms, Internet is the application environment for all the web services. Then for a single-member C-net, the other C-nets that interact with it to form a complete C-net are the environment of this C-net.

For example, the process in Fig. 2 can be described by BPEL in Appendix A. Following the definitions, the C-nets for the scenarios in Figs. 1 and 2 are described in Figs. 7 (a) and (b) respectively.

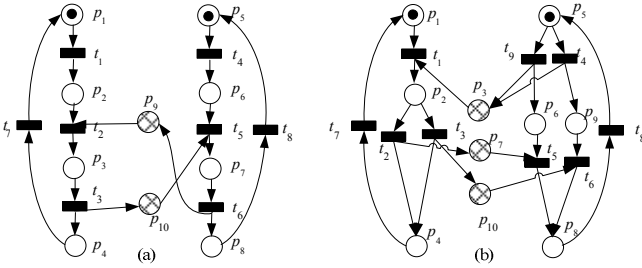


Fig. 7. The C-nets for the scenarios in Figs. 1 and 2

IV. COMPATIBILITY ANALYSIS AND POLICY OF C-NET

According to its definition, a C-net has an important property: there always exists one and only one process place that is marked with one token in each WMN.

According to Definition 3, since a WMN is built based on an SSN, whenever a web service sends or receives information following the structures in Fig. 3, there is always one and only one process place in a WMN which is marked with one token.

Definition 7: A complete C-net $N = \odot_{i=1}^n N_i$ is compatible iff the initial marking $M_0(\alpha_i) = 1$ and $M_0(p) = 0$, $\forall p \neq \alpha_i$ where $i \in \mathbb{N} = \{1, 2, \dots, n\}$ is reachable for each reachable marking M .

According to Definition 2, if $M_0(\alpha_i) = 1$ and $M_0(p) = 0$, $\forall p \neq \alpha_i$ is still reachable for each reachable marking M under the interaction situation, the complete C-net is compatible.

Lemma 1: A complete C-net is compatible iff the complete C-net is live.

Proof: (Sufficiency) From Definition 7, if a complete C-net $N = \odot_{i=1}^n N_i$ is incompatible, the initial marking $M_0(\alpha_i) = 1$ and $M_0(p) = 0$, $\forall p \neq \alpha_i$ where $i \in \mathbb{N} = \{1, 2, \dots, n\}$ is not reachable. According to the property of C-net, there must be a process place p which is marked with one token in N_i but the transition $t \in p^*$ is dead.

(Necessity) Suppose a complete C-net $N = \odot_{i=1}^n N_i$ with an initial marking M_0 and let $M \in R(N, M_0)$ be a reachable marking so that $t \in T$ is a dead transition at M , then according to Definition 7, $M_0 \notin R(N, M)$ and the complete C-net is incompatible. \square

Definition 8: A C-net N_1 is usable with environment N_2 iff the complete C-net $N = N_1 \odot N_2$ is compatible.

When complete C-net $N = N_1 \odot N_2$ is incompatible, either N_1 or its environment N_2 should be changed. This can serve as the basis to solve an incompatibility problem.

Theorem 1: Suppose a complete C-net N with an initial marking M_0 . Let $M \in R(N, M_0)$ and let $t \in T$ be a dead transition at M . Then \exists a siphon S , $\exists M(S) = 0$.

Proof: Let $S_i = \{i \in P_E | M(i) = 0\}$ and $S_p = \{p \in P | M(p) = 0\}$. Thus we have $S = S_i \cup S_p$ and we will prove that S is a nonempty unmarked siphon.

a) $S \neq \emptyset$. By contradiction, if $S = \emptyset$ then $S_i = \emptyset$. This means that all the interface places are marked with one token. By the property of C-net, we take the process place p where $M(p) = 1$.

Thus, transition p^* can fire, which contradicts the assumption.

b) $M(S) = 0$. Straight forward from the way we choose S_i and S_p .

c) S is a siphon. We consider two cases for *S .

$\forall t \in ^*q$ where $q \in S$. Following Definition 4, there are two cases. One is that $\{p^*\} = ^*t \cap P$ and $^*t \cap P_E = \emptyset$. The other is that $\{p^*\} = ^*t \cap P$ and $\{i_1, i_2, \dots, i_k\} = ^*t \cap P_E$. In the first case, we have $M(p^*) = 0$ (Otherwise the transition t can fire). Then we have $p^* \in S_p$ and $t \in p^* \in S^*$. In the second case,

$\exists p \in \{p', i_1, i_2, \dots, i_k\}$ and $M(p) = 0$ (Otherwise the transition t can fire). Then we have $t \in p^* \in S^*$. \square

Theorem 2: A complete C-net $N = \odot_{i=1}^n N_i$ is compatible iff $\forall M \in R(N, M_0)$, \forall (minimal) Siphon S , $M(S) \neq 0$.

Proof: (Sufficiency) From Theorem 1, if $\forall M \in R(N, M_0)$ we have $M(S) \neq 0$, then no transition can be dead. The complete C-net is live. From Lemma 1, we can conclude that the complete C-net is compatible.

(Necessity) From the basic property of siphons, if $\exists M \in R(N, M_0)$ we have $M(S) = 0$, then no output transition of the siphon S can be enabled. The complete C-net is not live. From Lemma 1, we can conclude that the complete C-net is incompatible. \square

Thus the problem of compatibility of web service interaction is transformed to the problem of empty minimal siphons in the C-net. Suppose that we have an incompatible complete C-net $N = \odot_{i=1}^n N_i$ where some minimal siphons can become empty. Our main goal is to introduce into the system an interface policy to guarantee that no empty minimal siphons is reachable during the evolution of the new C-net, i.e., the new C-net is compatible.

The following concepts are due to [39-43].

Definition 9: Let $S \subseteq P$ be a subset of places of N . P-vector λ_S is called the characteristic P-vector of S iff $\forall p \in S : \lambda_S(p) = 1$; otherwise $\lambda_S(p) = 0$.

Definition 10: Let $S \subseteq P$ be a subset of places of N and λ_S be the characteristic P-vector of S . η_S is called the characteristic T-vector of S if: $\eta_S^T = \lambda_S^T * [N]$.

Let S be a place subset of net N , and η_S be its characteristic T-vector. Sets $\{t \in T \mid \eta_S(t) > 0\}$, $\{t \in T \mid \eta_S(t) = 0\}$, $\{t \in T \mid \eta_S(t) < 0\}$ are the sets of transitions whose firings increase, maintain and decrease the number of tokens in S , respectively. For instance, let $S = \{p_{2-3}, p_5, p_{7-9}\}$ in Fig. 7(b). One can see that firing t_3 decreases the number of tokens by one in p_2 and maintains it in $\{p_3, p_5, p_{7-9}\}$. Hence firing t_3 decreases the number of tokens in S by one. It is so because $\eta_S(t_3) = -1$, where $\eta_S(t) = \{0, 0, -1, 0, 0, 0, 0, 0\}$. It is easy to see that the characteristic T-vector of the support of a P-invariant is 0. For another instance, let $S = \{p_{1-2}, p_4\}$ in Fig. 7(b), one can easily check that $\eta_S(t) = \{0, 0, 0, 0, 0, 0, 0, 0\}$ because $\lambda_S(p)$ is the support of P-invariant.

Definition 11: We denote the set of SMS of N as $\Omega = \{S_1, S_2, \dots, S_k\}$. $\Omega_E = \{S_1, S_2, \dots, S_i\}$ is a set of elementary siphons of N , iff $\text{Rank}[\eta_{S_1}, \eta_{S_2}, \dots, \eta_{S_i}] = \text{Rank}[\eta_{S_1}, \eta_{S_2}, \dots, \eta_{S_k}] = i$.

Definition 12: Suppose that $S \in \Omega \setminus \Omega_E$ is a siphon in N and $S_1, S_2, \dots, S_i \in \Omega_E$ be its elementary siphons. Then it is a strongly dependent siphon if $\exists A \in \Omega_E$, $\exists \eta_S = \sum_{S_i \in A} a_i \eta_{S_i}$ and a_i

> 0 . It is weakly dependent if $\exists A, B \in \Omega_E$, $\exists B \neq \emptyset$, $\eta_S = \sum_{S_i \in A} a_i \eta_{S_i} - \sum_{S_j \in B} a_j \eta_{S_j}$, $a_i, a_j > 0$.

Definition 13: An interface policy is a set of empty additional interface places and their corresponding information channels connected with the transitions in the complete C-net.

Therefore, the problem is how to build these information channels to guarantee that no empty minimal siphons are reachable in the new C-net. Suppose that S is an SMS of a net system (N_0, M_0) where $N_0 = \{P_0, P_{E0}, T_0, F_0\}$. Add an interface place V_S to N_0 to make P-vector $I = (\lambda_S \mid -1)^T$ be a P-invariant of the new net system (N_1, M_1) , where P-vector λ_S corresponds to S and $I(V_S) = -1$. We have $\forall p \in P_0 \cup P_{E0}$, $M_1(p) = M_0(p)$. According to Definition 3, $M_1(V_S) = 0$. We have $[N_1] = [[N_0]^T \mid L_{V_S}^T]^T$, where L_{V_S} is a row vector due to the addition of place V_S . Since $I = (\lambda_S \mid -1)^T$ is a P-invariant of the new net system, we have $I^T * [N_1] = 0^T$, i.e., $\lambda_S^T * [N_0] - L_{V_S} = 0^T$. Therefore, $L_{V_S} = \eta_S^T$.

Prerequisite 1: For every minimum siphon, $M_0(S) > 0$ holds.

Prerequisite 2: The characteristic T-vector η_S for every elementary SMS contains exactly one element of 1, one element of -1 and the other elements are all 0s.

Prerequisite 1 states the requirement for C-net compatibility. If \forall (minimal) siphon S , $M_0(S) = 0$, according to Theorem 2, the complete C-net is incompatible. Prerequisite 2 states the requirement for information channel. If it is not satisfied, the number of tokens in an interface place may exceed one and it is impossible to build an information channel.

Theorem 3: Application of the proposed interface policy to every elementary siphon of N under Prerequisites 1 and 2 can guarantee that the new C-net is compatible.

Proof:

1) For every minimum siphon that is not SMS, i.e., an invariant-controlled siphon, since $M_0(S) > 0$ holds, we have $M_1(S) = M_0(S) > 0$.

2) If $M_0(S) > 0$ for every elementary siphon S , according to the steps in Definition 14, we have $M_1(S) = M_0(S) > 0$. For every strongly dependent siphon S and $S_1, S_2, \dots, S_i \in \Omega_E$ if $\eta_s = a_1\eta_{s_1} + a_2\eta_{s_2} + \dots + a_i\eta_{s_i}$, we have $M_1(Vs_1) = M_1(Vs_2) = \dots = M_1(Vs_i) = 0$. Following Theorem 1 in [39], we have

$$M_0(S) > \sum_{j=1}^i a_j M_1(Vs_j) = 0 \text{ and } M_1(S) = M_0(S) > 0. \text{ Similarly,}$$

following Theorem 2 in [39], for every weakly dependent siphon S , $M_1(S) > 0$ also holds.

According to Theorem 2, the new complete C-net is compatible. \square

For instance, there are three minimum siphons in Fig. 7(a), i.e., $S_1 = \{p_{1-4}\}$, $S_2 = \{p_{5-8}\}$ and $S_3 = \{p_3, p_7, p_{9-10}\}$. But since $M_0(S_3) = 0$, Prerequisite 1 is not satisfied. The interface policy cannot be applied.

The example in Fig. 7(b) has four minimum siphons, i.e., $S_1 = \{p_{2-3}, p_5, p_{7-9}\}$, $S_2 = \{p_{2-3}, p_{5-6}, p_8, p_{10}\}$, $S_3 = \{p_1, p_2, p_4\}$ and $S_4 = \{p_{5-6}, p_{8-9}\}$. Since $M_0(S_i) = 1 > 0$, $i \in \{1, 2, 3, 4\}$, Prerequisite 1 is satisfied. S_3 and S_4 are supported by P-invariant and are always marked with one token. S_1 and S_2 are elementary SMS. We have $\eta_{s_1}(t) = \{0, 0, -1, 1, 0, 0, 0, 0\}$ and $\eta_{s_2}(t) = \{0, -1, 0, 0, 0, 0, 0, 1\}$. Prerequisite 2 is also satisfied. The interface policy can be applied.

V. IMPLEMENTATION FOR COMPATIBILITY OF C-NET

A. Changing environments

A large number of choices of services with different behavior can provide the same syntactic interfaces. According to Definition 8, when an n -member complete C-net $N = \odot_{i=1}^n N_i$ is incompatible, C-net N_i can be replaced with another C-net that provides the same syntactic interfaces but has different behavior. This “substitute” based action can be performed until we find a complete and compatible C-net. For example, after we replace the web service for the online shop in Fig. 2 with the web service in Fig. 8, i.e., after we change the C-net from Fig. 9(a) to Fig. 9(b), we find that the new C-net becomes compatible. Although the web service for the online shop in Fig. 2 is slightly different with the web service in Fig. 8, their behaviors are distinct. Both web services in Figs. 2 and 8 have the choice between the two

kinds of payment, i.e., VISA and eCash. On one hand, since the customer service makes a choice first and sends it to the online shop service, their composition is compatible in Fig. 8. On the other hand, since the online shop service makes a choice and does not notify the customer service, the customer might choose the “wrong” payment, and their composition ends up in a deadlock in Fig. 2.

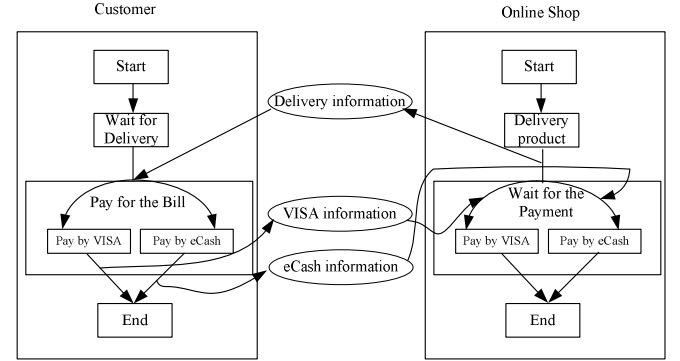


Fig. 8. Illustration of a compatible case

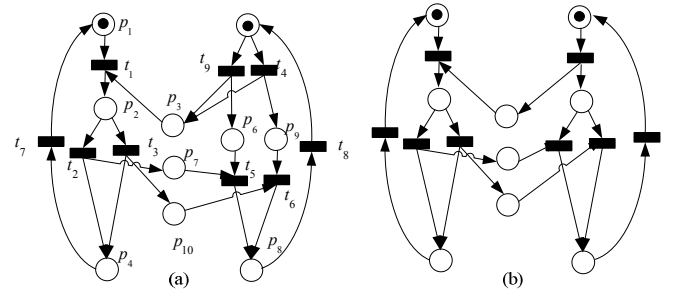


Fig. 9. Two optional C-nets

B. Appending additional information channels

This motivates us to propose a modification-based algorithm by appending additional information channels as follows:

Algorithm:

/* Obtain the n -member C-net N^* */

Step 1. Following Definition 3, transform the structures of BPEL process to WMN. Following Definition 5, obtain the n -member C-net N .

/* Add an information channel for each elementary siphon */

Step 2. Compute the set of minimum siphons Ψ in N . Compute the set of SMS Ω in N .

For every minimum siphon $S \in \Psi$, check Prerequisite 1. If it holds, then continue; and otherwise the policy is not applicable and go to *Step 9*.

Step 3. If $\Omega = \emptyset$, then go to *Step 9* otherwise continue.

Step 4. Construct characteristic T-vectors η_s of the SMS $S \in \Omega$ in N .

Step 5. Find the set of elementary siphons in a net N as Ω_E .

Suppose that $\Omega_E = \{ES_i \mid i = 1, 2, \dots, n\}$.

Step 6. For every elementary SMS $S \in \Omega_E$, check Prerequisite 2. If it holds, then continue, otherwise the policy is not

applicable and go to *Step 9*.

Step 7. For ES_i , $i = 1, 2, \dots, n$, add an information channel by the approach stated in Definition 13.

Step 8. The resultant C-net is compatible. We transform the additional information channels back to “reply” and “receive” basic structures in BPEL.

Step 9. Stop.

The idea underlying this algorithm is simple. First, model the BPEL with the n -member C-net. Second, find the set of SMS, elementary SMS in it. Third, check whether the n -member C-net is compatible and whether the method based on appending additional information channel can be adopted where Theorems 2 and 3 are successively utilized. Fourth, add additional information channel for each elementary SMS and the compatibility is guaranteed. Finally, transform the additional information channels back to “reply” and “receive” basic structures in BPEL. Its main limitations include: a) When the method is not applicable, we have to return to the method based on changing environments, i.e., the “substitute” based method, and b) The BPEL code must be allowed to change to attach more interfaces for web services.

C. Comparisons

We make detailed comparisons about the “substitute” based and modification based methods in two respects, i.e., the ability for detecting and resolving incompatibility.

In the aspect of detecting incompatibility, neither of the methods has clear computational advantage over the other. Since it is well known that the computation of a reachability graph is of exponential complexity, the “substitute” based method has exponential complexity when judging compatibility of each candidate complete C-net. The modification based method needs the complete siphon enumeration of a C-net model. The number of siphons grows fast and in the worst case grows exponentially with respect to the size of a net [44]. However, the former is sensitive to the change of initial markings while the later is not. This property can be used to reduce the computation complexity for incompatibility detection for different initial states. In this case, a new reachability graph should be obtained for each new initial state, but the same siphons should be checked.

In the aspect of resolving incompatibility, the modification based method is better than the “substitute” based one. Although the “substitute” based approach provides the solution for the models as mentioned in Section II when incompatibility occurs, the large number of web services brings in difficulty in finding out the exactly right one since a single error of a C-net N_i may collapse the usability of the whole n -member complete C-net. For example, in an n -member complete C-net, suppose that the number of candidate web services with the same syntactic interface for each member C-net is K , then in the worst-case the method has to analyze approximately K^n candidate complete C-nets.

When Prerequisites 1 and 2 are satisfied, the proposed

algorithm is to modify the C-net by adding additional information channels instead of replacing the web services. Thus, no matter how many member C-nets there are in a complete C-net or how many candidate web services with the same syntactic interface there are for each member C-net, as long as the algorithm can be implemented in a single candidate complete C-net, the compatibility of the new C-net is guaranteed.

VI. CASE STUDY

In this section we take the interaction example of customer and online shop web services as shown in Fig. 2. The whole process can be described by BPEL in *Appendix A*. Here we omit the definition of portTypes, plink, message, operation and variables by WSDL and focus on the normal interaction behavior. We obtain the n -member C-net N in Fig. 7(b). The C-net contains deadlock and it is not compatible. The interface policy can be applied after checking Prerequisites 1 and 2.

There are two elementary SMS in N , i.e., $S_1 = \{p_{2-3}, p_5, p_{7-9}\}$ and $S_2 = \{p_{2-3}, p_{5-6}, p_8, p_{10}\}$. We have $\eta_{s_1}(t) = \{0, 0, -1, 1, 0, 0, 0, 0, 0\}$ and $\eta_{s_2}(t) = \{0, -1, 0, 0, 0, 0, 0, 0, 1\}$.

We add two information places V_1 and V_2 . According to the correspondent row vector $\eta_{s_1}(t)$ and $\eta_{s_2}(t)$, we have the arcs (t_4, V_1) , (V_1, t_3) , (t_9, V_2) and (V_2, t_2) , where (t_4, V_1, t_3) and (t_9, V_2, t_2) are two additional information channels. We obtain the compatible C-net in Fig. 10. We transform the two additional information channels back to code in BPEL and obtain the new BPEL processes in *Appendix B*. According to Theorem 3, the new BPEL processes are compatible.

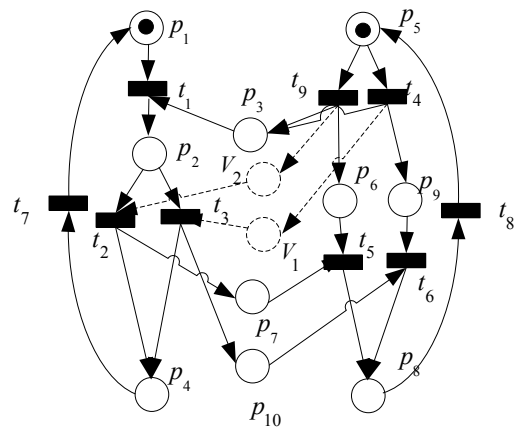


Fig. 10. Compatible C-net model for the web service interaction via adding additional information channels

From the new BPEL process in *Appendix B*, we see two additional partnerLinks that correspond to the two additional information channels, respectively. The information channel

that corresponds to (t_4, V_1, t_3) means that the online shop “tells” the customer to use eCash while the other information channel means the online shop “tells” the customer to use Visa.

We use ActiveVOS, which is the industry's first complete, integrated, standards-based visual orchestration system [45], to design and test our case. ActiveVOS integrate the tasks associated with the design, development, testing, deployment and maintenance of BPEL-based process compositions.

We first visually build the online shop service using various type of BPEL activity blocks on the canvas as shown in Fig. 11. We use the “Receive” type block to receive the request from the customer. The “Invoke” type blocks “InvokeVisaPayment” and “InvokeEcashPayment” denote the VisaPayment invocation and EcashPayment invocation respectively. The “Assign” type block denotes the successful payment by assigning a “true” value to a global variable. Finally, the “Reply” type block denotes the ending of the whole process.

We then run simulation of the service step by step. The console window shows “completed with fault: missing reply” which means that the “InvokeEcashPayment” path is blocked due to receiving a Visa message instead of an eCash message. However, after we add two additional information channels and run the simulation again, the simulation result as shown in Fig. 12 indicates that the process is completed successfully.

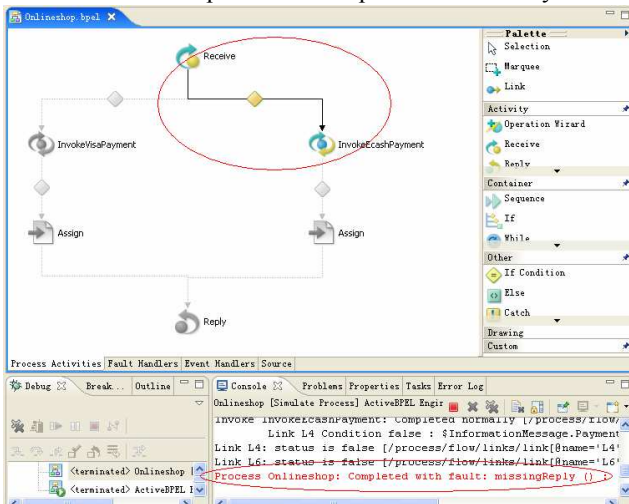


Fig. 11. The process was blocked.

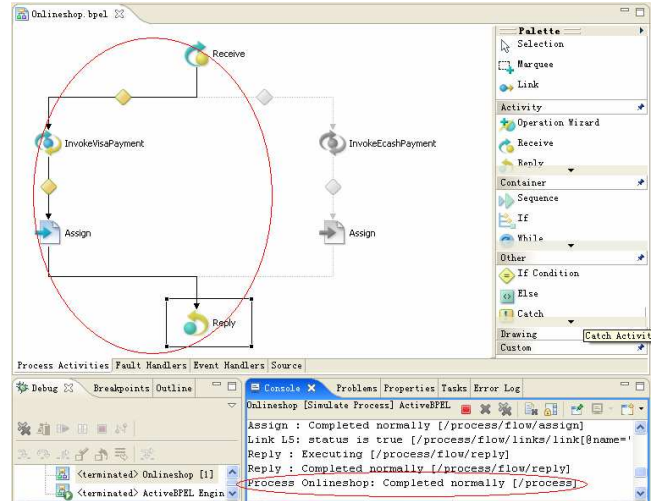


Fig. 12. The process completed normally.

VII. CONCLUSIONS

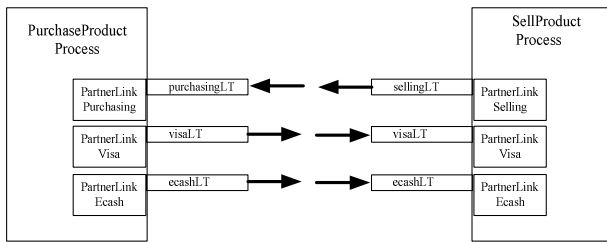
In service-oriented computing environment, BPEL is more and more widely used as a service composition language. However, it lacks compatibility analysis that renders the composition unusable. Although current studies try to solve the problem based on reachability analysis, it suffers from a space exploration problem. This paper proposes a Petri net based analysis technique and a corresponding policy based on appending additional information channels to deal with the incompatibility of web services.

The paper presents a formal model for web service interaction with Petri nets. The basic activities, structured activities and interfaces among BPEL processes are all covered by the model elements in a C-net. Next, by carefully analyzing the structure of C-net, we discover that the compatibility of web services is equal to the existence of non-empty minimal siphons, which leads to the useful method that the compatibility can be guaranteed by appending additional information channels. Then, after adding information channels in Petri net elements, we transform the channels back to code in BPEL and obtain the new compatible web services. The real life case and BPEL codes validate our approach and prove that our approach can be readily put into industrial applications.

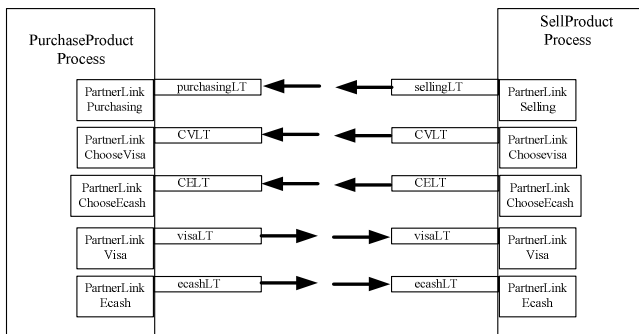
There are some limitations in our work. First, our algorithm cannot lead to a solution if the method of adding information channels is not applicable. Second, since the algorithm is based on the computation of minimal siphons, in some complex structured C-net, the computation can be expensive. Some polynomial complex algorithms to find and control siphons should be adopted for C-net [46-47]. Third, although our algorithm prevents the SMS from being empty by appending additional information channels, it does not prove that this action will not introduce more SMS. Although it does not bring in more SMS in many tested real examples, it is interesting to prove this conclusion after more investigation about the special structure of C-nets. This will be left for future exploration.

APPENDIX

A. The old BPEL processes



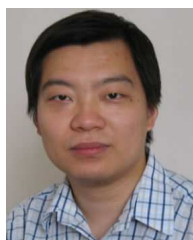
B. The new BPEL processes



REFERENCES

- [1] M.P. Papazoglou and D. Georgakopoulos, Service-Oriented Computing, *Communications of the ACM*, Vol. 46, No. 10, Oct. 2003, pp. 25-28.
- [2] J. Zhang, C.K. Chang, L.J. Zhang and P.C.K. Hung, "Toward a Service-Oriented Development Through a Case Study," *IEEE Trans. on Systems, Man, and Cybernetics: Part A*, Vol. 37, Iss. 6, Nov. 2007, pp. 955-969.
- [3] A. Barros, M. Dumas and P. Oaks, "Standards for Web Service Choreography and Orchestration: Status and Perspectives", *Proc. of 2006 Business Process Management Workshops*, Nancy, France, 2006, pp. 61-74.
- [4] R. Hull and J. Su, "Tools for Composite Web Services: A Short Overview", *ACM SIGMOD Record*, Vol 34, No. 2, June 2005, pp. 86-95.
- [5] T. Andrews, F. Curbera, H. Dholakia, Y. Golland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana, "Business Process Execution Language for Web Services", version 1.1, May 2003, <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel>
- [6] W.M.P. van der Aalst, "Don't go with the flow: Web services composition standards exposed", *IEEE Intelligent Systems*, Vol. 18, No. 1, Jan./Feb. 2003, pp. 72-76.
- [7] F. Curbera, R. Khalaf, N. Mukhi, S. Tai, and S. Weerawarana, "The next step in web services", *Communications of the ACM*, Vol. 46, No. 10, Oct. 2003, pp. 29-34.
- [8] F. Curbera, N. Mukhi, W. Nagy, S. Weerawarana, "Implementing BPEL4WS: the Architecture of a BPEL4WS Implementation", *Concurrency and Computation: Practice & Experience*, Vol. 18, Iss. 10, Aug. 2006, pp 1219-1228.
- [9] M.D. Backer, "On the Verification of Web Services Compatibility: A Petri Net Approach", *Proc. of 2004 International Workshop on Modeling Inter-Organizational Systems in conjunction with OnTheMove Federated Conferences*, Agia Napa, Cyprus, 2004, pp. 810-821.
- [10] S. Narayanan and S. McIlraith, "Simulation, Verification and Automated Composition of Web Services," *Proc. of the 11th International World Wide Web Conference*, Honolulu, USA, 2002, pp. 77-88.
- [11] H. Nezhad, B. Benatallah, A. Martens, F. Curbera and F. Casati, "Semi-Automated Adaptation of Service Interactions", *Proc. of the 16th International World Wide Web Conference*, Banff, Alberta, Canada, May 2007, pp. 993-1002.
- [12] W. Tan, Y.S. Fan and M.C. Zhou, "A Petri Net-based Method for Compatibility Analysis and Composition of Web Services in Business Process Execution Language," accepted by *IEEE Trans. on Automation Science and Engineering*, Sept. 2007.
- [13] A. Martens, "Usability of Web services," *Proc. of the 4th International Conference on Web Information Systems Engineering Workshops*, Rome, Italy, Dec. 2003, pp. 182-190.
- [14] H. Ben-Abdallah and S. Leue, "Syntactic detection of process divergence and non-local choice in message sequence charts", *Proc. of the 2nd International Workshop on Tools and Algorithms for the Construction and Analysis of Systems*, Enschede, Netherlands, Apr. 1997, pp. 259-274.
- [15] P. Wohed, W.M.P. van der Aalst, M. Dumas and A.H.M. ter Hofstede, "Analysis of Web services composition languages: The case of BPEL4WS", *Proc. of 22nd International Conference on Conceptual Modeling*, Chicago, IL, Oct. 2003, pp. 200-215.
- [16] H. Foster, S. Uchitel, J. Magee and J. Kramer, "Tool support for model-based engineering of Web service compositions", *Proc. of the 2005 IEEE International Conference on Web Services*, Orlando, Florida, USA, Jul. 2005, Vol. 1, pp. 95-102.
- [17] X. Fu, T. Bultan and J. Su, "Analysis of Interacting BPEL Web Services", *Proc. of the 13th International World Wide Web Conference*, New York, NY, May 2004, pp. 621-630.
- [18] R. Heckel and L. Mariani, "Automatic Conformance Testing of Web Services", *Proc. of the 8th International Conference on Fundamental Approaches to Software Engineering*, Edinburgh, Scotland, Apr. 2005, pp. 34-48.
- [19] D. Fahland and W. Reisig, "ASM-based Semantics for BPEL: The negative Control Flow", *Proc. of the 12th International Workshop on Abstract State Machines*, Paris, France, Mar. 2005, pp. 131-152.
- [20] H. Foster, S. Uchitel, J. Magee, and J. Kramer, "Model-based verification of Web service composition", *Proc. of the 18th IEEE International Conference on Automated Software Engineering*, Montreal, Canada, Oct. 2003, pp. 152-161.
- [21] A. Ferrara, "Web Services: a Process Algebra Approach", *Proc. of the 2nd International Conference on Service Oriented Computing*, New York City, NY, USA, 2004, pp. 242-251.
- [22] L. Bordeaux, G. Salaun, D. Berardi and M. Mecella, "When Are Two Web Services Compatible?", *Proc. of the 5th VLDB International Workshop on Technologies for e-Services*, Toronto, Canada, 2004, pp. 44-57.
- [23] W.M.P. van der Aalst, "The application of Petri Nets to workflow management", *The Journal of Circuits, Systems and Computers*, Vol. 8, No. 1, 1998, pp. 21-66.
- [24] A. Martens, R. Hamadi and B. Benatallah, "A Petri Net based Model for Web Service Composition", *Proc. of the 14th Australian Database Conference*, Adelaide, Australia, Feb. 2003, pp. 191-200.
- [25] C. Ouyang, E. Verbeek, W.M.P. van der Aalst, S. Breutel, M. Dumas and A.H.M. ter Hofstede, "WofBPEL: A Tool for Automated Analysis of BPEL Processes", *Proc. of the 3rd International Conference on Service Oriented Computing*, Amsterdam, Netherlands, Dec. 2005, pp. 484-489.
- [26] C. Ouyang, H.M.W. Verbeek, W.M.P. van der Aalst, S. Breutel, M. Dumas and A.H.M. ter Hofstede, "Formal Semantics and Analysis of Control Flow in WS-BPEL", *Science of Computer Programming*, Vol. 67, Iss. 2-3, Jul. 2007, pp. 125-332.
- [27] 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, Sept. 2007, pp. 77-91.
- [28] N. Lohmann, P. Massuthe, C. Stahl, and D. Weinberg, "Analyzing Interacting BPEL Processes", *Proc. of the 4th International Conference on Business Process Management*, Vienna, Austria, Sept. 2006, pp. 17-32.
- [29] D. König, N. Lohmann, S. Moser, C. Stahl, K. Wolf, "Extending the Compatibility Notion for Abstract WS-BPEL Processes", *Proc. of the 17th International World Wide Web Conference*, Beijing, China, Apr. 2008, pp. 785-794.
- [30] S. Hinz, K. Schmidt and C. Stahl, "Transforming BPEL to Petri Nets", *Proc. of the 3rd International Conference on Business Process Management*, Nancy, France, Sept. 2005, pp. 220-235.
- [31] A. Martens, "Analyzing Web Service based Business Processes", *Proc. of the 8th International Conference on Fundamental Approaches to Software Engineering*, Edinburgh, Scotland, Apr. 2005, pp. 19-33.

- [32] A. Martens, "Consistency between Executable and Abstract Processes", *Proc. of 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service*, Hong Kong, Mar. 2005, pp.60-67.
- [33] B. Hruz and M.C. Zhou, *Modeling and Control of Discrete Event Dynamic Systems*, Springer, London, UK, 2007.
- [34] M.C. Zhou and K. Venkatesh, *Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach*, World Scientific, Singapore, 1998.
- [35] M.P. Fanti and M.C. Zhou, "Deadlock Control Methods in Automated Manufacturing Systems," *IEEE Trans. on Systems, Man, and Cybernetics, Part A*, Vol. 34, Iss. 1, Jan. 2004, pp. 5-22.
- [36] P.C. Xiong, Y.S. Fan and M.C. Zhou, "QoS-aware Web Service Configuration," *IEEE Trans. on Systems, Man, and Cybernetics: Part A*, Vol. 38, Iss. 4, Jul. 2008, pp. 888-895.
- [37] P.C. Xiong, Y.S. Fan and M.C. Zhou, "Web Service Configuration under Multiple Quality-of-Service Attribute," accepted by *IEEE Trans. on Automation Science and Engineering*, Nov. 2007.
- [38] J. Ezpeleta, J.M. Colom and J. Martinez, "A Petri Net based Deadlock Prevention Policy for Flexible Manufacturing Systems," *IEEE Trans. on Robotics and Automation*, Vol. 11, No. 2, Apr. 1995, pp. 173-184.
- [39] Z.W. Li and M.C. Zhou, "Elementary Siphons of Petri Nets and Their Applications to Deadlock Prevention in Flexible Manufacturing Systems," *IEEE Trans. on Systems, Man, and Cybernetics, Part A*, Vol. 34, No. 1, Jan. 2004, pp. 38-51.
- [40] Z.W. Li and M.C. Zhou, "Comparison of Two Deadlock Prevention Methods for Different-size Flexible Manufacturing Systems", *International Journal of Intelligent Control and Systems*, Vol. 10, No. 3, Sept. 2005, pp. 235-243.
- [41] Z.W. Li and M.C. Zhou, "Two-stage Method to Design Liveness-enforcing Petri Net Supervisor for FMS", *IEEE Trans. on Industrial Informatics*, Vol. 2, No. 4, Nov. 2006, pp. 313-325.
- [42] Z.W. Li and M.C. Zhou, "Clarifications on the Definitions of Elementary Siphons in Petri Nets", *IEEE Trans. on Systems, Man, and Cybernetics: Part A*, Vol. 36, No. 6, Nov. 2006, pp. 1227-1229.
- [43] Z.W. Li, H.S. Hu, and A.R. Wang, "Design of Liveness-enforcing Supervisors for Flexible Manufacturing Systems Using Petri Nets", *IEEE Trans. on Systems, Man, and Cybernetics, Part C*, Vol. 37, No. 4, Jul. 2007, pp. 517-526.
- [44] J. Ezpeleta, J. M. Couvreur and M. Silva, "A new technique for finding a generating family of siphons, traps, and st-Components: Application to colored Petri nets," *Proc. of the 12th International Conference on Applications and Theory of Petri Nets: Advances in Petri Nets*, Denmark, Jun. 1991, pp. 126-147.
- [45] www.activevos.com/
- [46] Z.W. Li and M.C. Zhou, "On Siphon Computation for Deadlock Control in a Class of Petri Nets," *IEEE Trans. on Systems, Man, and Cybernetics: Part A*, Vol. 38, No. 3, May 2008, pp. 667-679.
- [47] Z.W. Li and M.C. Zhou, "Control of Elementary and Dependent Siphons in Petri Nets and their Application," *IEEE Trans. on Systems, Man, and Cybernetics: Part A*, Vol. 38, No. 1, Jan. 2008, pp. 133-148.



PengCheng Xiong is now a research assistant in the College of Computing, Georgia Institute of Technology, Atlanta GA. His research interest includes web service modeling and composition, Petri nets and applications, business process reengineering, and system integration. He has already published a dozen of papers in journals and conference proceedings.

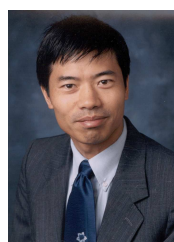


YuShun Fan received the Ph.D. degrees in control theory and application from Tsinghua University, Beijing, China, in 1990. From September 1993 to 1995, he was a Visiting Scientist, supported by Alexander von Humboldt Stiftung, at Fraunhofer Institute for Production System and Design Technology (FhG/IPK), German.

He is currently Professor of Department of Automation, Vice Director National CIMS Engineering Research Center of China, Director of System Integration Institute, and Director of the Networking Manufacturing

Laboratory in Tsinghua University. He is also guest Professor of Shanghai Jiaotong University, Southwest Jiaotong University, and Shenyang Institute of Automation of Chinese Academy of Sciences. He is an editorial member of the International Journal of Computer Integrated Manufacturing, International Journal of Enterprise Information System, International Journal of Advanced Engineering Informatics, Computer Integrated Manufacturing Systems, Aeronautical Manufacturing, Information and Control, China Manufacturing Information Engineering, and New Technology and New Process. He is a member of the IFAC Advanced Manufacturing Technology Committee and vice director of China Standardization Committee for Industrial Automation System and Integration, vice director of Manufacturing Technology Committee of China Automation Association. From 1993 to 2004, he has worked as expert in the Field of Computer Integrated Manufacturing Systems of the China High Technology R & D Program.

He served on the Program Committees of the 1992 International Symposium on CIM, Beijing, China, 1997 IEEE International Conference on Factory Automation & Emerging Technology, 2002 International Workshop on Emergent Technologies in Engineering Cooperative Information Systems, 2003 and 2004 IEEE International Conference on SMC, the 11th ISPE International Conference on Concurrent Engineerin, 2004 IEEE International Conference on E-Commerce Technology for Dynamic E-Business, 2006 International Conference on Computer Supported Collaborative Work in Design, 2006 Multi-Agent Systems in Manufacturing Track, the 7th IFIP International Conference on Information Technology for BALANCED AUTOMATION SYSTEMS in Manufacturing and Services, 2006 IFIP TC8 International Conference on Enterprise Information Systems. He authored ten books and published more than 350 research papers in journals and conferences in his research areas: enterprise integration, enterprise modeling, workflow technology, intelligent agent, object oriented complex system analysis, computer integrated manufacturing.



MengChu Zhou (S'88-M'90-SM'93-F'03) received his B.S. degree from Nanjing University of Science and Technology, Nanjing, China in 1983, M.S. degree from Beijing Institute of Technology, Beijing, China in 1986, and Ph. D. degree in Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY in 1990. He joined New Jersey Institute of Technology (NJIT), Newark, NJ in 1990, and is currently a Professor of Electrical and Computer Engineering and the Director of Discrete-Event

Systems Laboratory. His research interests are in computer-integrated systems, Petri nets, wireless ad hoc and sensor networks, system security, semiconductor manufacturing, and embedded control. He has over 250 publications including 6 books, 110 journal papers, and 16 book-chapters. He co-authored with F. DiCesare *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*, Kluwer Academic, Boston, MA, 1993, edited *Petri Nets in Flexible and Agile Automation*, Kluwer Academic, 1995, co-authored with K. Venkatesh *Modeling, Simulation, and Control of Flexible Manufacturing Systems: A Petri Net Approach*, World Scientific, 1998, co-edited with M. P. Fanti, *Deadlock Resolution in Computer-Integrated Systems*, Marcel Dekker, 2005, co-authored with H. Zhu, *Object-Oriented Programming in C++: A Project-based Approach*, Tsinghua University Press, 2005, and co-authored with B. Hruz, *Modeling and Control of Discrete Event Dynamic Systems*, Springer, 2007.

He was invited to lecture in Australia, Canada, China, France, Germany, Hong Kong, Italy, Japan, Korea, Mexico, Taiwan, and US. He served as Associate Editor of IEEE Transactions on Robotics and Automation from 1997 to 2000, and IEEE Transactions on Automation Science and Engineering from 2004-2007, and currently Managing Editor of IEEE Transactions on Systems, Man and Cybernetics: Part C, Associate Editor of IEEE Transactions on Industrial Informatics, and Editor-in-Chief of International Journal of Intelligent Control and Systems. He served as Guest-Editor for many journals including IEEE Transactions on Industrial Electronics and IEEE Transactions on Semiconductor Manufacturing. He was General Co-Chair of 2003 IEEE International Conference on System, Man and Cybernetics, Washington DC, October 5-8, 2003, Founding General Co-Chair of 2004 IEEE Int. Conf. on Networking, Sensing and Control, Taipei, March 21-23, 2004, and General Chair of 2006 IEEE Int. Conf. on Networking, Sensing and Control, Ft. Lauderdale, Florida, U.S.A. April 23-25, 2006. He was Program Chair of 1998 and 2001 IEEE International Conference on System, Man and Cybernetics

(SMC) and 1997 IEEE International Conference on Emerging Technologies and Factory Automation. He also served other chair positions including Local Arrangement Chair of 2007 American Control Conference, New York, July 2007. He is serving as General Chair of IEEE Conf. on Automation Science and Engineering, Washington D.C., Aug. 2008. He organized and chaired over 80 technical sessions and served on program committees for many conferences. Dr. Zhou has led or participated in over thirty research and education projects with total budget over \$10M, funded by National Science Foundation, Department of Defense, Engineering Foundation, New Jersey Science and Technology Commission, and industry. He was the recipient of NSF's Research Initiation Award, CIM University-LEAD Award by Society of Manufacturing Engineers, Perlis Research Award by NJIT, Humboldt Research Award for US Senior Scientists, Leadership Award and Academic Achievement Award by Chinese Association for Science and Technology-USA, Asian American Achievement Award by Asian American Heritage Council of New Jersey, and Distinguished Lecturer of IEEE SMC Society. He was founding chair of Discrete Event Systems Technical Committee of IEEE SMC Society, and Chair of Semiconductor Manufacturing Automation Technical Committee of IEEE Robotics and Automation Society. He is a life member of Chinese Association for Science and Technology-USA and served as its President in 1999.