



**HAL**  
open science

# A Structural Verification of Web Services Composition Compatibility

Kamel Barkaoui, Maryam Eslamichalandar, Meryem Kaabachi

► **To cite this version:**

Kamel Barkaoui, Maryam Eslamichalandar, Meryem Kaabachi. A Structural Verification of Web Services Composition Compatibility. 2010, pp.30-41. hal-00704572

**HAL Id: hal-00704572**

**<https://hal.science/hal-00704572>**

Submitted on 14 Jun 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Structural Verification of Web Services Composition Compatibility

Kamel Barkaoui , Maryam Eslamichalandar , Meryem Kaabachi

CEDRIC - CNAM Paris, France  
kamel.barkaoui@cnam.fr

**Abstract.** A fundamental feature of service oriented computing is that simple services need to be composed for generating complex services. This work focuses on the analysis and verification of behavior models of web services composition. In particular, we have to check that neither deadlock nor livelock occurs in this composition. Usually, the verification of such integration, with or without mediators, is achieved by using techniques based on state space exploration of a given service formal model. In this paper, we present an approach based on structure theory of Petri nets allowing the recognition of necessary and/or sufficient conditions ensuring compatible composition and a better understanding of the incompatibility sources.

**Key words:** Web service composition, Compatibility, Structure theory of Petri Nets

## 1 Introduction

With the increasing use of the platform independent software architecture such as web-based applications, web services exist in distributed environments. Therefore a web service often depends on other web services which have been implemented by different vendors and their correct usage is governed by constraints specified on their interfaces. Whilst different languages such BPEL4WS [7] have been proposed for describing and executing workflow specifications for a web service composition invocation, we still have a critical need of methods and tools to solve many problems related to service interaction [1,5,8]. In this paper, we deal with the issue of verification of web services composition compatibility by using the structure theory of Petri nets formalism. A WS composition is called compatible if its underlying interaction service is such that each service can terminate properly. Our approach is mainly motivated by the fact that verification techniques particularly structural techniques and tools developed for Petri nets can be fully exploited in the context of web services described by BPEL4WS [3], or others. The main goal of this paper is to show how structure theory of Petri nets can provide some guidelines and solutions for ensuring the correctness of web services composition. This paper is organized as follows. Section 2 gives a brief summary of basics of Petri nets and of its related structure theory. Section 3 introduces the open nets as a formal model for web services and their

composition. In Section 4, using recent results of structure theory of Petri, we deal with the correctness of the WS composition in particular with behavioral compatibility and provide new way of looking at interaction services permitting us the identification of some interface patterns ensuring compatibility between two or more services. Section 5 concludes this paper.

## 2 Basics of Petri nets

In this section, after giving basic definitions and properties of Petri nets, we present some recent structure theory results.

### 2.1 Definitions and notations

A Petri Net  $(P/T) N = (P, T, F, m_0)$  consists of :

- $P$  a finite set of places and  $T$  a finite set of transitions with  $(P \cup T) \neq \emptyset$  and  $(P \cap T = \emptyset)$ ,
- $F \subseteq (P \times T) \cup (T \times P)$  is the flow relation
- $m_0$  is the initial marking where a marking  $m$  is a mapping  $m : P \rightarrow \mathbb{N}$ .

Each node  $x \in P \cup T$  of the net has a pre-set and a post-set defined respectively as follows :  $\cdot x = \{y \in P \cup T / (y, x) \in F\}$   
and  $x \cdot = \{y \in P \cup T / (x, y) \in F\}$ .

The incidence matrix of the net is the matrix  $C$  indexed by  $P \times T$  and defined by  $C(p, t) = W(t, p) - W(p, t)$  with  $W(u) = 1$  if  $u \in F$  and  $W(u) = 0$  otherwise.

An integer vector  $f \neq 0$ , indexed by  $P$  ( $f \in Z^P$ ) is a  $P$ -invariant iff it satisfies  ${}^t f \cdot C = 0$ .

An integer vector  $g, g \neq 0$ , indexed by  $T$  ( $g \in N^T$ ) is a  $T$ -invariant iff it satisfies  $C \cdot g = 0$ .

we denote by  $\|f\| = \{p \in P / f(p) \neq 0\}$  the support of  $f$  ;

$\|f\|^+ = \{p \in P / f(p) > 0\}$  and  $\|f\|^- = \{p \in P / f(p) < 0\}$ , if there exists a  $P$ -invariant  $f / \|f\|^+ = P$  then  $N$  is said to be conservative.

and by  $\|m\| = \{p \in P / m(p) > 0\}$  the support of marking  $m$  .

A transition  $t$  is said to be enabled under  $m$  iff  $\cdot t \subseteq \|m\|$  (i.e. there is a token on every place of  $\cdot t$ ). A transition  $t$  enabled under a marking  $m$  can be fired, leading to a new marking  $m'$  such that :  $\forall p \in P : m'(p) = m(p) + C(p, t)$ . The set of reachable markings from a marking  $m$  in  $N$  is denoted by  $R(N, m)$ .

We recall some behavioral properties of a Petri net  $N$ .

- A marking  $m^*$  is a home state if and only if  $\forall m' \in R(N, m), m^* \in R(N, m')$ .
- $N$  is reversible iff  $m_0$  is a home state.
- $N$  is bounded iff  $\forall p \in P : \exists k \in \mathbb{N}, \forall m \in R(N, m_0), m(p) \leq k$   
i.e.  $R(N, m_0)$  is finite.
- $N$  is structurally bounded iff  $N$  is bounded for any  $m_0$ .
- If  $N$  is conservative then  $N$  is structurally bounded.
- $N$  is quasi-live iff  $\forall t \in T, \exists m \in R(N, m_0)$  for which  $t$  is enabled.

- $N$  is deadlock-free (or weakly live) iff  $\forall m \in R(N, m_0), \exists t \in T$  enabled in  $m$ .
- $N$  is live iff  $\forall t \in T, \forall m \in R(N, m_0) \exists m' \in R(N, m)$  for which  $t$  is enabled.
- $N$  is structurally live iff  $\exists m_0 / (N, m_0)$  is live.
- A bounded and live Petri net is said to be well formed.

## 2.2 Basics of Structure Theory of P/T nets

Structure theory of Petri nets investigates the relationship between the behavior and the structure of the net. The use of structural methods for the analysis of systems present two major advantages with respect to other approaches : the state explosion problem inherent to concurrent systems is avoided , otherwise limited , and this relationship usually leads to a deep understanding of the system.

A remarkable sub structure of Petri nets is that of Siphon.

Let  $N$  be a P/T system. A non empty set  $S \subseteq P$  is called a siphon if and only if  $\cdot S \subseteq S \cdot$ .

$S$  is said to be minimal if and only if it contains no other siphon as a proper subset.

Due to its structure a siphon which is unmarked will never becomes marked.

In this case, transitions of  $S \cdot$  cannot be live so  $S$  need to be controlled.

$S$  is said to be controlled if and only if  $S$  is marked at any reachable marking i.e.  $\forall m \in R(N, m_0), \exists p \in S / m(p) > 0$ .

CS-property:  $N$  is said to be satisfying the controlled-siphon property if and only if all its minimal siphons are controlled.

We recall below two well-known basic relations between liveness and the CS-property [2]. The first states that the CS-property is a sufficient deadlock-freeness condition while the second states that the CS-property is a necessary liveness condition.

**Proposition 1.** *Let  $N$  be a P/T net. If  $N$  satisfies the CS-property then  $N$  is deadlockfree (weakly live).*

**Proposition 2.** *Let  $N$  be a P/T net. If  $N$  is live then  $N$  satisfies the CS-property.*

The following proposition recall the two structural (sufficient but not necessary) conditions permitting us to check if a given siphon is controlled or not.

**Proposition 3.** *Let  $S$  be a siphon of  $N$  satisfying one of the two following conditions, then  $S$  is controlled [2] :*

- i)  $\exists R \subseteq S$  such that  $R \cdot \subseteq \cdot R$  and  $R \cap \|m_0\| \neq \emptyset$
- ii)  $\exists P$ -invariant  $f \in Z^P$  such that  $S \subseteq \|f\|, \|f\|^+ \subseteq S$   
and ,  $\sum_{p \in P} [f(p) \cdot m_0(p)] > 0$

K-Systems : P/T nets for which CS-property is not only necessary but also sufficient liveness condition, in other words , systems for which there is equivalence between liveness and CS-property are called K-systems [2].

Root-place : Let  $t \in T$  be a transition of a P/T net  $N$  and  $r \in \cdot t$  ;  $r$  is called a root place for  $t$  if and only if  $\forall p \in \cdot t, r \subseteq p$  .

Ordered transition : A transition  $t \in T$  is said to be ordered if and only if  $\forall p, q \in \cdot t, p \subseteq q$  or  $q \subseteq p$  , an ordered transition has at least one root place. A transition admitting a root place is not necessarily ordered.

We denote by:

- Root ( $t$ ) the set of root places of  $t$ .
- $T_0(N)$  the set of ordered transitions of  $N$ .
- $T_R(N)$  the set of transitions of  $N$  admitting a root (i.e.  $T_0(N) \subseteq T_R(N)$ ).
- Root ( $N$ ) the set of root places of  $N$ .
- The Root Component of  $N$  is the net  $R_C(N) = (P_C(N), T_C(N), F_C(N))$  defined as follows:
  - $P_C = \text{Root}(N)$ ,  $T_C = T_R(N)$ .
  - $F_C$  is the restriction of  $F$  such that:  
 $(p,t) \in F_C$  iff  $p \in \text{Root}(t)$  and  $(t, p) \in F_C$  iff  $(t, p) \in F$ .

Two main subclasses of K-systems namely ordered nets and root nets can be recognized structurally and effectively [2].

(1)  $N$  is called an Ordered net iff  $T_0(N) = T$  (i.e. all its transitions are ordered).

(2)  $N$  is called a Root net iff  $T_R(N) = T$  , ( $T_0(N) \neq T$  ) and its root component  $R_C(N)$  is bounded and strongly connected.

Note that by definition these two subclasses are disjoint.

**Theorem 1.** *Let  $N$  be an Ordered net or a Root net.  $N$  is live if and only if it satisfies the CS-property [2].*

In particular for well known subclasses of ordered nets for which  $\text{Root}(t) = \cdot t \forall t \in T$ , (therefore  $R_c(N) = N$ ) such Extended Free Choice (EFC) nets, the cs-property ( by condition i) is a necessary and sufficient liveness condition . Moreover, if such nets are bounded then liveness property (i.e. here condition (i)) can be decided in polynomial time [10]. Also as control by trap (i) is preserved after increasing marking (contrary to control by invariant (ii)), liveness property is monotonic for K-systems satisfying CS-property by condition (i). Based on these structural theoretical results , we show in the rest of the paper , how compatibility analysis and verification of web services composition can be under taken efficiently .

### 3 Modeling and Specification of web Service Composition

For specification and modeling services, we focus on the concepts which are independent of a given implementation language [7,9] . First of all, a service has a definition describing its behavior and its interface . An instance of a given service

corresponds to an execution of the activities of this service .These activities are atomic units of work specified in the service definition. The interface of a service consists of a set of ports. A pair of ports can be connected using a channel, thus enabling the exchange of messages sent or received by services .

In this work , we abstract from non-functional properties, data and information semantics. Hence , web service can be viewed as a control structure describing its behavior according to an interface to communicate asynchronously with other services in order to reach a final state (i.e. a state representing a proper termination). As for modeling of business processes and workflows, P/T nets are well appropriate to model such control perspective of web services. A web service is modeled by a P/T net called open net [5] which is an extension of a workflow net [6] or service net [4] by adding, to the internal places, two specific disjoint sets of input and output places (called interface places) modeling the service interface. Services can be composed by connecting the interfaces.

More precisely, each input place (i.e. with empty pre-set) corresponds to an input port of the interface (used for receiving messages from a distinguished channel) whereas an output place (i.e. empty post-set) corresponds to an output port of the interface (used for sending messages via a distinguished channel).

**Definition 1.** *An open net  $N = (P, T, F, I, O, m_0, m_f)$  consists of :  
A Petri net  $N^* = (P, T, F, m_0, m_f)$  such that :*

- $m_0 = s$  ( the initial marking of the service ) ,  $s = \emptyset$
- $m_f = o$  ( the final marking of the service ) ,  $o = \emptyset$

*with an interface places  $(I \cup O \subseteq P)$  such that :*

- $\forall p \in I \cup O, m_f(p) = m_o(p) = 0$
- $\forall p \in I, \dot{p} = \emptyset$  (input interfaces places)
- $\forall p \in O, \dot{p} = \emptyset$  (output interface places)

Our definition of open nets is not restrictive . Indeed any P/T net  $N^*$  with an initial marking defined on more than one initial place , or admitting a set of final markings ( with mutually exclusive supports ) , can be transformed easily to an equivalent open net. Also our open nets are not elementary communicating in the sense that a transition can be connected to more than one interface place.

The basic web services infrastructure provides simple interactions between a client and a web service. However, the implementation of a web services business needs generally the invocation of other web services. Thus it is necessary to combine the functionality of several web services. The process of developing a composite service is called service composition. Composite services are recursively defined as an aggregation of elementary and composite services. The composition of two or more services generates a new service providing both the original behavior of initial services and a new collaborative behavior for carrying out a new composite task [7].

From a modeling point of view, a composite service can be described as a

recursive composition of open nets [8]. Communication between services takes place by exchanging messages via interface places. Thus, composing two open nets is modeled by merging their respective shared constituents which are the equally labeled input and output interface places. Such a fused interface place models a channel and a token on such a place corresponds to a pending message in the respective channel. As it is convenient to require that all communications are bilateral and directed, i.e. every interface place  $p \in (I \cup O)$  has only one open net that sends into  $p$  and only one open net that receives from  $p$ .

Thereby, open nets involved in a composition are pairwise interface compatible i.e. only input interface places of the one open net overlap with output interface places of the other. This interface compatibility is a basic and first requirement for services composition .

**Definition 2.** *Let  $N_1$  and  $N_2$  be two open nets with pairwise disjoint constituents except for the interfaces . If  $I = (I_1 \cap I_2) = \emptyset$  and  $(O_1 \cap O_2) = \emptyset$  then  $N_1$  and  $N_2$  are interface compatible.*

**Definition 3.** *Let  $N_1$  and  $N_2$  two interface compatible open nets. Their composition  $N = N_1 \oplus N_2$  is the open net defined as follows:*

- $P = P_1 \cup P_2$  ;  $T = T_1 \cup T_2$ ;  $F = F_1 \cup F_2$ ;
- $I = (I_1 \cup I_2) \setminus (O_1 \cup O_2)$  ;  $O = (O_1 \cup O_2) \setminus (I_1 \cup I_2)$  ;
- $m_0 = m_{01} \oplus m_{02}$ ;  $m_f = m_{f1} \oplus m_{f2}$

Open net composition is commutative and associative i.e. for interface compatible open nets  $N_1, N_2$  and  $N_3$  :  $N_1 \oplus N_2 = N_2 \oplus N_1$  and  $(N_1 \oplus N_2) \oplus N_3 = N_1 \oplus (N_2 \oplus N_3)$ .

An open net with an empty interface ( $I = \emptyset$  and  $O = \emptyset$ ) is called a closed net. By choreography , we refer to the coordination of messages between services involved in a composite service. Therefore a service choreography can be described as a closed net .

The next section is devoted to check the verification of behavioral properties of a closed obtained by composing open nets.

## 4 Structural Verification of Composition Compatibility

A composite web service modeled as a closed net is a service that consists of coordination of several conceptually autonomous but interface compatible services. Although it is not easy to specify how this coordination should behave, we focus here on these three behavioral requirements :

- *Weak-Compatibility* . A closed net  $N$  is said to be *weak-compatible* iff  $N$  is *deadlock-free*.
- *Compatibility* which excludes not only deadlocks but also livelocks. A closed net  $N$  is said to be compatible iff  $m_f$  is *home state* (final state is always reachable).
- *Strong-Compatibility*. A closed net  $N$  is said to be strong compatible iff  $N$  is compatible and quasi-live (proper termination and no dead activities).

Our contribution in this paper is to show how using recent results of structure theory of Petri nets (that can be interpreted as restrictions or operating guidelines on service interaction patterns), we can check or ensure structurally these behavioral properties.

Let us precise that a deadlock state  $m$  in a closed net  $N$  is a reachable state ( $m \neq m_f$ ) under which no transition is enabled.

Obviously, compatibility implies weak compatibility.

Let  $N = N_1 \oplus N_2 \oplus \dots \oplus N_k$  be a closed net.

Let  $N_i$  be an open net,  $N_i^* = (P_i, T_i, F_i, m_{0i}, m_{fi})$  is called the inner subnet of  $N_i$ . We denote by  $N_i^{**}$  the subnet obtained from  $N_i^*$  by connecting the initial place  $s_i$  to the terminal place  $o_i$  by an additional transition  $t_i^*$ .

Let  $N = N_1 \oplus N_2 \oplus \dots \oplus N_k$  we denote by  $\theta(N)$  the net obtained by substituting in each  $N_i$ ,  $N_i^*$  by  $N_i^{**}$ .

First of all, from the two well known propositions (1) and (2), we can deduce easily the two following propositions :

**Proposition 4.** *Let  $N = N_1 \oplus N_2 \oplus \dots \oplus N_k$  be a closed net. If  $\theta(N)$  satisfies the cs-property then  $N$  is weak compatible.*

**Proposition 5.** *Let  $N = N_1 \oplus N_2 \oplus \dots \oplus N_k$ . If  $N$  is strong compatible then  $\theta(N)$  satisfies CS property. (we prove that  $\theta(N)$  is live )*

Let us consider the closed net obtained by the two open nets of Fig.1 described in [5]. As the cs-property, is not satisfied : the siphon  $S = (\text{food}, \text{money}, P7, P3)$  is empty at  $m_0$ ,  $N$  cannot be live neither deadlockfree. Consequently  $N$  is not weak compatible.

Now, Consider the two interface compatible open nets of Fig.2, the corresponding closed net  $N$  is such that  $\theta(N)$  satisfies the cs-property therefore  $N$  is weak compatible.

However  $N$  is not compatible : indeed the final marking  $m_f = p_4 + p_{14}$  cannot be reached from the accessible marking  $m^* = p_4 + p_{14} + p_7$ .



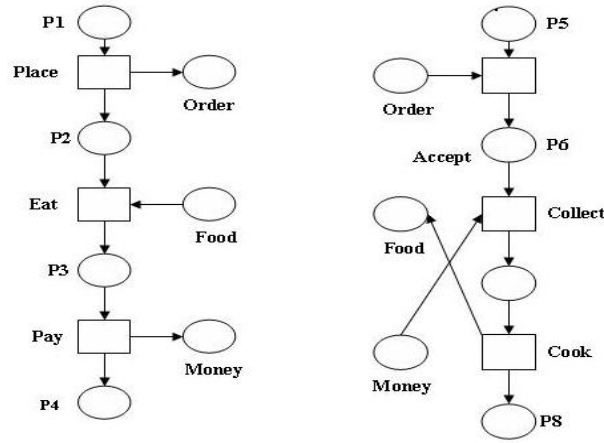


Fig. 1. A not weak compatible closed net

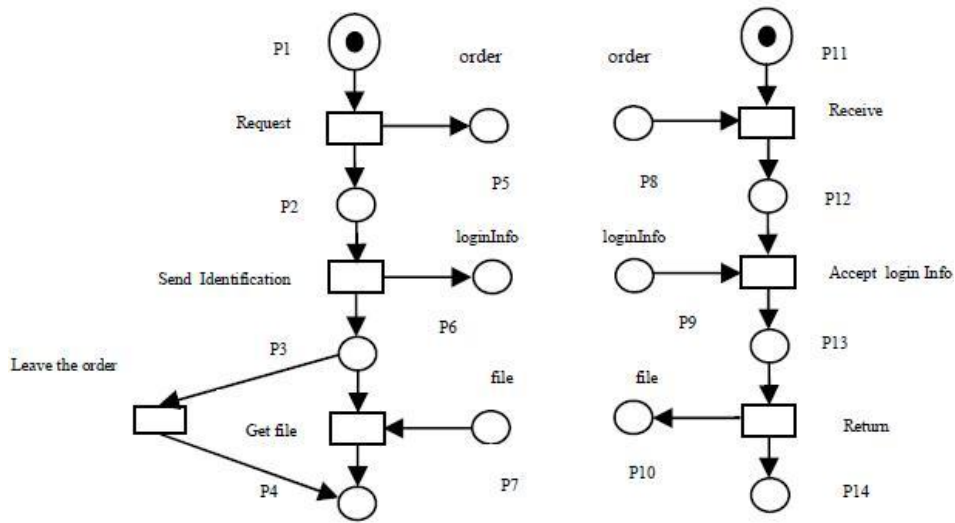


Fig. 2. A weak compatible (but not compatible) closed net

**Theorem 2.** Let  $N = N_1 \oplus N_2 \oplus \dots \oplus N_k$  be a closed net. If  $N$  is strong compatible then all  $N_i^*$  are sound.

*Proof.* Suppose there exists  $N_i^*$  not sound, i.e.  $N_i^{**}$  is not live or not bounded. Case (1):  $N_i^{**}$  is not live i.e. there is transition  $t \in T_i$  not live in  $N_i^{**}$ . As (input) interface places only limit the behavior of the associated open net  $N_i^*$ ,  $t$  remains not live in  $\theta(N)$ , thus  $N$  cannot be strong compatible.

Case (2):  $N_i^{**}$  live but not bounded, thus  $m_f$  cannot be a home state and  $N$  is not compatible.

According to previous results, strong compatibility of open nets requires not only interface compatibility of open nets but also soundness of their inner subnets. We define now two classes of open nets namely Ordered open nets and Root open nets for which soundness is equivalent to cs-property[6].

**Definition 4.** *Let  $N$  be an open net.  $N$  is called an Ordered open net if and only if  $N^{**}$  is an ordered net.*

**Definition 5.** *Let  $N$  be an open net.  $N$  is called a Root open net if and only if  $N^{**}$  is a Root net.*

From this two classes of open nets, we define a large subclass of closed nets called Root closed nets presenting realistic interfaces patterns and for which compatibility can be structurally decided. In this subclass we impose a restriction on the connection nature of interface places such that root internal places are preserved after composition i.e. an input interface place can be a root place but it cannot take the place of another internal one. A larger subclass of composite service can be obtained by applying the basic building process of Root closed nets in a recursive way, i.e. modules can be root closed nets or more complex nets defined in this way.

**Definition 6.** *A  $P/T$  system  $N = (P, T, F, m_0)$  is called a Root Closed net (or simply an RC net) if and only if  $P$  is the disjoint union  $P_1, \dots, P_n$  and  $B$ ,  $T$  is the disjoint union  $T_1, \dots, T_n$  and the following holds:*

- i) For every  $i \in \{1, \dots, n\}$ , let  $N_i = \langle N_i^{**}, I_i, O_i \rangle$  be an open net such that :
  - $(I_i \cup O_i) \subseteq B$
  - $N_i^{**} = (P_i, T_i, F_i, m_{0i}, m_{fi})$  where  $F_i \subseteq (P_i * T_i) \cup (T_i * P_i)$  is an ordered or root open net satisfying CS-property.
- ii) For every  $N_i^{**} i \in \{1, \dots, n\}$ :  $\forall b \in B$ ,  $b$  preserves the sets of root places of  $N_i^{**}$  (i.e.  $\forall t \in T_i, \text{Root}(t)_{N_i^{**}} \subseteq \text{Root}(t)_{N_i}$ )
- iii) There exists a subset  $B' \subseteq B$  such that the sub net induced by the inner subnets  $\subseteq N_i^{**} (i \in \{1, \dots, n\})$  and  $B'$  (denoted by  $\theta(N)_{B'}$ ) is conservative and strongly connected (if  $B' = B$ ,  $\theta(N)_{B'} = \theta(N)$ )

**Theorem 3.** *Let  $N$  be a Root Closed net. The three following assertions are equivalent :*

- $N$  is deadlock free
- $N$  satisfies CS property
- $N$  is live

*Proof.* Root Closed nets are, by construction, a subclass of Synchronized Dead Closed Systems (SDCS) [2] which are a K-systems. Therefore this equivalence holds.

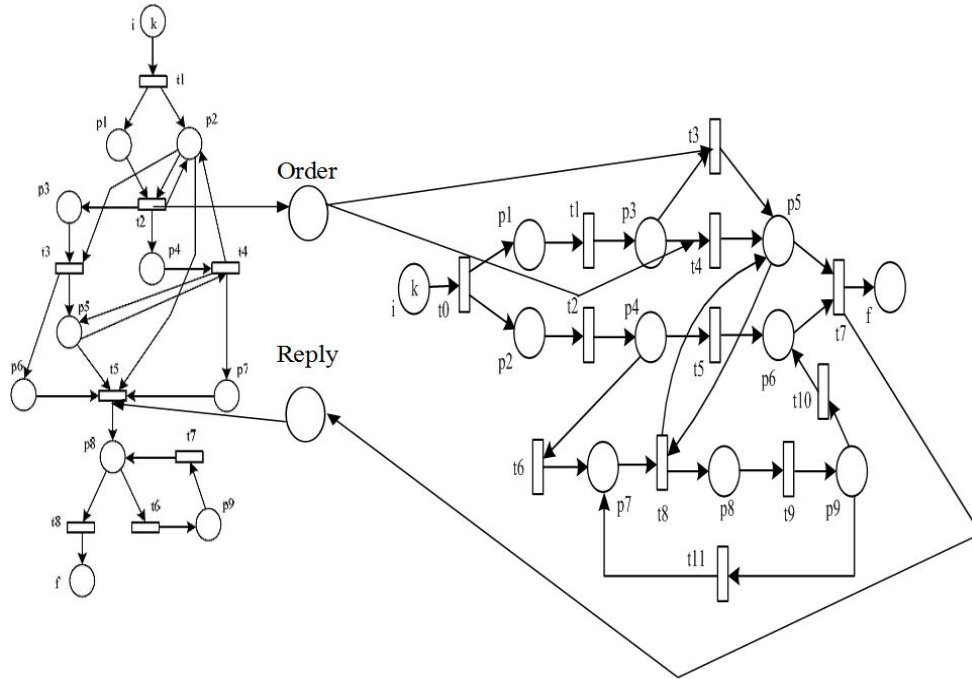
**Corollary 1.** *Let  $N$  be a Root Closed net. If  $\theta(N)_{B'}$  satisfies cs-property then  $N$  is weak compatible. This means that  $N$  is deadlock free but some interface places can be unbounded.*

**Corollary 2.** *Let  $N$  be a Root Closed net such that  $B' = B$ . If  $\theta(N)$  satisfies cs property, then  $N$  is strong compatible.*

*Proof.* Since  $B' = B$ ,  $\theta(N)$  is live and bounded. This means that  $N$  is deadlock free and the final marking is well a home state.

Let us consider now the root closed net  $N = N_1 \oplus N_2$  of Fig.3 where  $N_1$ (on the right) is a sound root open net and  $N_2$  is a sound ordered open net. As  $\theta(N)$  satisfies the cs-property we can claim that  $N$  is strong compatible.

We consider now the closed net  $N$  obtained by composition of  $N_1, N_2$  and  $N_3$  of Fig.4 from [5].  $N_1, N_2$  and  $N_3$  are sound ordered nets, moreover  $N$  satisfies cs-property. However as  $N$  is not a root closed net ( the input interface place  $CMoney$  does not preserve the root place of transition  $t^*$ ) we cannot claim that  $N$  is strong compatible. In fact  $N$  is compatible but not strong compatible (  $t^*$  is not live in  $N$ ).



**Fig. 3.** A strong compatible root closed net

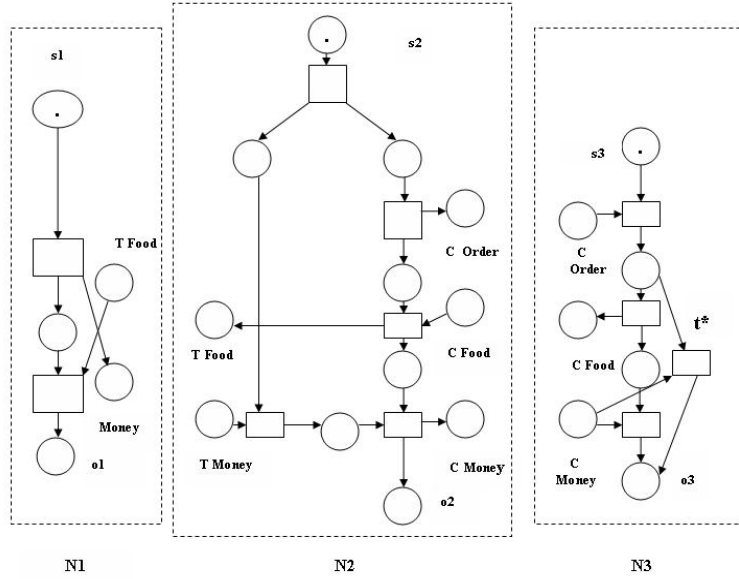


Fig. 4. A compatible (but not strong compatible) closed net

## 5 Conclusion

This paper presented a structural approach to verifying process interactions for coordinated web services composition. Using results of structure theory of Petri net, we have identified necessary and /or sufficient structural conditions on web services interfaces ensuring the composition compatibility.

The main contribution of this paper is to provide a structural technique to check if two or more web services are compatible and a better understanding of the incompatibility sources.

A direction for further work is to exploit these results to develop efficient solutions for the substitutability problem (i.e. the assurance that a given service can be replaced by another one as a better partner in a given composition).

## References

1. Cheng Y., Wang Z.: Research on Reachability Verification of Web Service Composition. World Congress on Software Engineering. pp. 233237 (2009).
2. Barkaoui K., Couvreur J.M., Klai K.: On the equivalence between liveness and deadlock-freeness in Petri nets. Lecture Notes in Computer Science. 3536, 90 107 (2005).

3. Dun H., Xu H., Wang L.: Transformation of BPEL Processes to Petri Nets. Theoretical Aspects of Software Engineering, 2008. TASE'08. 2nd IFIP/IEEE International Symposium on. pp. 166173 (2008).
4. Hamadi R., Benatallah B.: A Petri net-based model for web service composition. Proceedings of the 14th Australasian database conference-Volume 17. p. 200 (2003).
5. Aalst, W.M., Mooij, A.J., Stahl, C., Wolf, K.: Service Interaction: Patterns, Formalization, and Analysis. Formal Methods for Web Services. p. 88 (2009).
6. Barkaoui, K.; Ben Ayed R.; Sbai, Z.: Workflow soundness verification based on structure theory of Petri nets. International Journal of Computing and Information sciences (IJCIS) Vol.5, n1, April 2007.
7. Alves, A., Arkin, A., Askary, S., Barreto, C., Bloch, B., Curbera, F., Ford, M., Golland, Y., Gu?zar, A., Kartha, N., Liu, C., Khalaf, R., Koenig, D., Marin, M., Mehta, V., Thatte, S., Rijn, D., Yendluri, P., Yiu, A.: Web Services Business Process Execution Language Version 2.0 (OASIS Standard). WS-BPEL TC OASIS (2007), <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html>.
8. Lohmann, N., Massuthe, P., Stahl, C., Weinberg, D.: Analyzing interacting WS-BPEL processes using flexible model generation. Data and Knowledge Engineering 64(1), 38-54 (2008).
9. J.Zhang,J-Y.Chung,C. K.Chang,and S.Kim.WS-Net: A Petri-net based specification model for web services. Proc. ICWS,pp. 420-427.IEEE,2004.
10. K.Barkaoui et M.Minoux.A Polynomial Time Graph Algorithm to Decide Liveness of some basic classes of Bounded Petri Nets. In Application and theory of Petri Nets.LNCS 616.pp.62-75,Springer Berlin / Heidelberg.1992