Sufficient and Necessary Condition to Decide Compatibility for a Class of Interorganizational Workflow Nets

Guanjun Liu¹,² and Lijing Chen³

¹Department of Computer Science, Tongji University, Shanghai 201804, China
²Department of Computer Science, Humboldt-Universität zu Berlin, 12489 Berlin, Germany
³Department of Software Engineering, Shandong University of Science and Technology, Qingdao 266590, China

Correspondence should be addressed to Guanjun Liu; liugj1116@163.com

Received 5 June 2014; Revised 23 August 2014; Accepted 26 August 2014

1. Introduction

Petri nets are widely used to model concurrent/distributed systems due to both the intuitive descriptions for these systems and the diversified analysis methods. Researchers usually define different Petri net classes for different systems with different features.

For example, in flexible manufacturing systems [1–7], every product corresponds to one or several manufacturing processes. Every process uses a group of resources (like machines or robots) by a fixed order. These processes are not required to interact or collaborate with each other but have to share common resources. S²PRs [8–10] are famous in modeling these systems. Particularly, researchers utilize structures of Petri nets, such as siphons, resource-transition circuits, and structurally circular waits, to decide liveness or deal with deadlock for these systems [11–14].

Another important application of Petri nets is to model and analyze such concurrent systems as web services, in which multiple parallel processes interact/collaborate via sending/receiving messages. Interorganizational workflow nets (IWF-nets for short) [15–19] are famous in these Petri net classes to model these concurrent systems. They can well characterize the system features, especially on synchronous and/or asynchronous communication. This paper focuses on IWF-nets.

Compatibility [20, 21] is a crucial property of IWF-nets. It implies that the target state can always be reached, no deadlock or livelock takes place, and each event/activity has a (potential) right to execute. If the target state is not reached for a given run, the designed system possibly have problems (e.g., deadlocks). If an event/activity of the designed system is not triggered for any run, then the event/activity should not have been defined. Notice that Section 2 will show that compatibility of IWF-nets is identical with soundness of workflow nets (WF-nets) [22] since an IWF-net can be translated into an equivalent WF-net. IWF-nets mainly focus on interactions/collaborations among different processes. In the next text, terminologies of soundness and compatibility are used without distinction.

van der Aalst et al. have proven that the soundness problem is decidable for general WF-nets [17, 22, 23]. It was also proven that the soundness problem is PSPACE-complete for bounded WF-nets [24]. Fortunately, van der Aalst et al. [22, 25] gave a polynomial-time algorithm to solve the soundness problem for free-choice WF-nets (FCWF-nets for
short), which is based on the rank theory proposed for well-formed free-choice nets by Desel and Esparza [26]. FCWF-nets can well model the structures of business processes such as AND-split, AND-join, OR-split, and OR-join [22, 25, 27]. However, some concurrent models like web services composite have to consider the interaction among different components/processes [15, 20, 21, 28–30], which makes the related models more and more complex so that FCWF-nets cannot model them in many cases. So far, only van der Aalst et al. proposed net-structure-based conditions to decide soundness for a few subclasses of WF-nets such as free-choice and well-structured ones [25]. There is not too much work on the soundness decision for IWF-nets, especially on the basis of the net structures.

Additionally, many design patterns were proposed in order to standardize the system design and reduce the occurrence of some bad things (e.g., deadlocks) [16, 31]. For example, van der Aalst et al. defined some service interaction things. In general, one still needs to verify soundness after a design. Section 2 reviews some notions and notations. Section 3 presents the basis of this condition. SIWF-nets can model many cases of interactions. To the best of our knowledge, it is the first time to propose a net-structure-based condition to solve the compatibility problem for IWF-nets. Therefore, the method proposed in this paper opens up a new way to explore the net-structure-based conditions of deciding the compatibility for more complex subclasses of IWF-nets.

The remainder of this paper is organized as follows. Section 2 reviews some notions and notations. Section 3 defines SIWF-nets. Section 4 proposes a net-structure-based condition to decide compatibility, and Section 5 develops a decision algorithm. Section 6 concludes this paper.

2. Basic Notions

For readability, in this section Petri nets and WF-nets are recalled that are from [22, 32, 33].

2.1. Petri Nets. First, let \( \mathbb{N} = \{0, 1, 2, \ldots\} \) be the set of nonnegative integers. Given \( m \in \mathbb{N} \) and \( m > 0 \), denote \( \mathbb{N}_m = \{1, 2, \ldots, m\} \) by the set of integers from 1 to \( m \). Denote \( \mathbb{N}_0 = \{0\} \).

**Definition 1 (net).** A net is a 3-tuple \( N = (P, T, F) \), where \( P \) is a finite set of places, \( T \) is a finite set of transitions, \( F \subseteq (P \times T) \cup (T \times P) \) is a set of arcs, and \( P \cap T = \emptyset \).

A net may be seen as a directed bipartite graph. Generally, a transition is represented by a rectangle and a place by a circle in a net graph. A path of a net is a nonempty sequence \( x_1x_2\cdots x_n \) of nodes such that for all \( j \in \mathbb{N}_n \), \( (x_j, x_{j+1}) \in F \). A path \( x_1x_2\cdots x_n \) is elementary if and only if for any two nodes \( x_i \) and \( x_k \) of the path, \( i \neq k \Rightarrow x_i \neq x_k \). A path \( x_1x_2\cdots x_n \) is a circuit if and only if no node occurs more than once in it and \( (x_n, x_1) \in F \). A net is acyclic if and only if it has no circuits. A net is strongly connected if and only if for any two nodes \( x \) and \( y \), there is a path from \( x \) to \( y \).

A transition \( t \) is called an input transition of a place \( p \) and \( p \) is called an output place of a transition \( t \) if and only if \( (t, p) \in F \). Input place and output transition can be defined accordingly. Given a net \( N = (P, T, F) \) and a node \( x \in P \cup T \), \( *x = \{y \in P \cup T \mid (x, y) \in F\} \) and \( x' = \{y \in P \cup T \mid (x, y) \in F\} \) are called the preset and postset of \( x \), respectively. Two different transitions \( t_1 \) and \( t_2 \) are in conflict if and only if \( t_1 \cap t_2 \neq \emptyset \).

\( N' = (P', T', F') \) is a subnet of \( N = (P, T, F) \) if and only if \( P' \subseteq P, T' \subseteq T, \) and \( F' = F \cap (T' \times P') \cup ((P' \times T') \cap F) \). \( N' \) contains a subnet of the former.

A marking of \( N = (P, T, F) \) is a mapping \( M : P \rightarrow \mathbb{N} \). A place \( p \) is marked at \( M(p) \) if and only if \( M(p) > 0 \).

Notice that a marking may be viewed as a \(|P|\)-dimensional nonnegative integer vector in which every element represents the number of tokens in corresponding place at this marking. For example, \( M = (1, 0, 6) \) over \( p \in \{p_1, p_2, p_3\} \) represents that \( p_1, p_2, \) and \( p_3 \) have 1, 0, and 6 tokens at \( M \), respectively. When the number of places is very large and the distribution of tokens is sparse, the above two kinds of presentation of a marking are relatively complex. For convenience, \( M \) is denoted as a multiset \( M = \sum_{p \in P} M(p) \cdot p \) in this paper. For the above example, it is written as \( M = p_1 + 6p_3 \) or \( M = \{p_1, 6p_3\} \).

If for all \( p \in *t \), \( M(p) > 0 \), then \( t \) is said to be enabled at \( M \), which is denoted as \( M[t] \). If \( \exists p \in *t \), \( M(p) = 0 \), then \( t \) is said to be disabled at \( M \), which is denoted by \( \neg M[t] \). Firing an enabled transition \( t \) produces a new marking \( M' \), which is denoted by \( M[t]M' \), such that \( M'(p) = M(p) - 1 \) if \( p \in *t \); \( M'(p) = M(p) + 1 \) if \( p \in t \sim *t \); and \( M'(p) = M(p) \) otherwise.

A marking \( M_k \) is reachable from a marking \( M \) if and only if there exists a transition sequence \( \sigma = t_1t_2\cdots t_k \) such that \( M[t_1]M_1[t_2]\cdots M_{k-1}[t_k]M_k \), \( M(\sigma)M_k \) represents that \( M \) reaches \( M_k \) after firing sequence \( \sigma \). The set of all markings reachable from \( M \) in a net \( N \) is denoted by \( R(N, M) \). Given a transition sequence \( \sigma \), \( \Psi(\sigma) \) is denoted by the multiset of transitions occurring in \( \sigma \). For an example of \( \sigma = t_1t_2t_3t_4 \), \( \Psi(\sigma) = \{2t_1, t_2, t_4\} \).

A net \( N \) with an initial marking \( M_0 \) is called a Petri net or net system and denoted by \( (N, M_0) \).

A Petri net \( (N, M_0) = (P, T, F, M_0) \) is called \( k \)-bounded if and only if \( k \in \mathbb{N} \), for all \( p \in P \), for all \( M \in R(N, M_0) : M(p) \leq k \). A Petri net is bounded if and only if it is \( k \)-bounded for some integer \( k \). A 1-bounded Petri net is also called safe. A net \( N \) is structurally bounded if and only if \( (N, M_0) \) is bounded for any initial marking \( M_0 \).
A transition $t$ is said to be \textit{dead} at a marking $M$ if and only if for all $M' \in R(N, M); \neg M'[t]$.

A transition $t$ is said to be \textit{live} at a marking $M$ if and only if for each $M' \in R(N, M)$, $t$ is not dead at $M'$. A Petri net $(N, M_0)$ is \textit{live} if and only if each transition of it is live at $M_0$. A net $N$ is \textit{structurally live} if and only if there is an initial marking $M$, such that $(N, M_0)$ is live.

A net $N = (P, T, F)$ is a \textit{marked graph} if and only if for all $p \in P$: $|p| = |p'| = 1$.

A net $N = (P, T, F)$ is a \textit{free-choice net} if and only if for all $p_1, p_2 \in P$: $(p_1 \cap p_2 \neq 0 \land p_1 \neq p_2) \Rightarrow |p_1'| = |p_2'| = 1$.

2.2. WF-Nets

\textbf{Definition 2 (WF-net).} A net $N = (P, T, F)$ is a WF-net if

(1) $N$ has two special places $i$ and $o$ where $i \in P$ is called source place if $i' = \emptyset$ and $o \in P$ is called sink place if $o' = \emptyset$;

(2) $N^E = (P, T \cup \{b\}, F \cup \{(i, b), (o, b)\})$ is strongly connected where $b \notin T$.

In Definition 2, $N^E$ is called the \textit{trivial extension} of $N$.

\textbf{Definition 3 (soundness of WF-net).} Let $N = (P, T, F)$ be a WF-net, $M_0 = i$, and $M_d = o$. $N$ is sound if

(1) for all $M \in R(N, M_0); M_d \in R(N, M)$;
(2) for all $M \in R(N, M_0); M \geq M_d \Rightarrow M = M_d$;
(3) for all $t \in T$, $\exists M \in R(N, M_0); M[t]$.

This definition was given in the early work of van der Aalst [34], and later he showed that the second requirement is implied by the first one [22]. Generally, $M_0 = i$ and $M_d = o$ are called initial and target markings of a WF-net, respectively. Additionally, a safe (resp., bounded) WF-net means that the WF-net is safe (resp., bounded) at its initial marking. van der Aalst also proved that the soundness of a WF-net is equal to the liveness and boundedness of its trivial extension at the initial marking [25].

A subclass of nets called \textit{inter-organizational workflow nets} (IWF-nets for short) [15] are often used to model the composition of web services, interorganizational business processes, or some other concurrent systems in which multiple processes interact via sending/receiving messages. An IWF-net describes the synchronous and/or asynchronous communication among multiple partners (each partner is modeled by a basic WF-net) [15, 35]. The following definition considers the asynchronous communication only.

\textbf{Definition 4 (IWF-net).} A net $N = (N_1, N_2, \ldots, N_m, P_C, F_C)$ is an IWF-net if

(1) $N_i = (P_i, T_i, F_i), \ldots$, and $N_m = (P_m, T_m, F_m)$ are pairwise disjoint WF-nets where $m \geq 2$ and these WF-nets are called basic;
(2) $P_C$ is a finite set of channel places such that $P_C \cap P_j = \emptyset$ for each $j \in \mathbb{N}_m$;
(3) $F_C \subseteq (P_C \times \bigcup_{j=1}^{m} T_j) \cup (\bigcup_{j=1}^{m} T_j \times P_C)$ is a set of arcs by which channel places connect with the $m$ basic WF-nets;
(4) for all $c \in P_C, \exists j, k \in \mathbb{N}_m; j \neq k \land c \subseteq T_j \land c' \subseteq T_k \land c' \neq 0$.

\textbf{Example 5.} Figures 1(a) and 1(b) are two IWF-nets whose basic WF-nets (see Figures 1(c) and 1(d)) are the same but interactions are different.

From the fourth item of Definition 4 it is known that each channel place is used only by two fixed basic WF-nets. In other words, two different basic WF-nets cannot send messages into the same channel place; similarly, two different basic WF-nets cannot take messages from the same channel place either. Certainly, two different basic WF-nets may use multiple channel places to communicate.

Notice that Definition 1 uses a 3-tuple to represent a net while Definition 4 uses an $(m+2)$-tuple to represent an IWF-net. In fact, if an IWF-net is also represented by a 3-tuple $(P, T, F)$, then $P = P_1 \cup \cdots \cup P_m \cup P_C$, $T = T_1 \cup \cdots \cup T_m$, and $F = F_1 \cup \cdots \cup F_m \cup F_C$. For convenience, an IWF-net is represented by a $(m+2)$-tuple. When no confusion is caused, $T$ is used to represent the set of all transitions of an IWF-net; that is, $T = T_1 \cup \cdots \cup T_m$. For all $j \in \mathbb{N}_m$, denote $i_j$ and $o_j$ by the source and sink places of $N_j$, respectively.

\textbf{Definition 6 (compatibility of IWF-net).} Let $N = (N_1, N_2, \ldots, N_m, P_C, F_C)$ be an IWF-net, $M_{0i} = i_1 + i_2 + \cdots + i_m$, and $M_{0o} = o_1 + o_2 + \cdots + o_m$. $N$ is compatible if

(1) for all $M \in R(N, M_0); M_d \in R(N, M)$;
(2) for all $M \in R(N, M_0); M \geq M_d \Rightarrow M = M_d$;
(3) for all $t \in T$, $\exists M \in R(N, M_0); M[t]$.

For instance, Figure 1(a) is compatible but Figure 1(b) is not. In fact, if two special places $i$ and $o$ and two special transitions $t_i$ and $t_o$ are added to an IWF-net such that $t_i = \{i\} \land t_i' = \{i_1, \ldots, i_m\} \land t_o = \{o_1, \ldots, o_m\} \land t_o' = \{o\}$, then this new net is a WF-net by Definition 2. Particularly, it is easy to prove that the original IWF-net is compatible if and only if the new WF-net is sound. Therefore, an IWF-net may be thought of as a special WF-net and the definitions of compatibility and soundness are equal. In this paper, an IWF-net is sometimes called a WF-net directly.

For convenience, the \textit{trivial extension} of an IWF-net is that a transition is added to the IWF-net such that the outputs of the transition are exactly all source places and the inputs of the transition are exactly all sink places. $M_{0i}$ and $M_{0o}$ in Definition 6 are called the \textit{initial} and \textit{target} markings of $N$, respectively.

If a WF-net is also a free-choice net, then it is called \textit{free-choice WF-net} (FCWF-net for short).

3. SIWF-Nets

This section defines a subclass of IWF-nets named \textit{simple interorganizational workflow nets} (SIWF-nets for short).
Definition 7 (SIWF-net). \(N = (N_1, N_2, \ldots, N_m, P_C, F_C)\) is a simple IWF-net (SIWF-net) if 

1. \(N\) is an acyclic IWF-net;
2. for all \(j \in \mathbb{N}_m: N_j\) is a sound FCWF-net;
3. for all \(c \in P_C: |c^*| = |c| = 1\).

In fact, Figures 1(a) and 1(b) are two SIWF-nets that have the same basic FCWF-nets as shown in Figures 1(c) and 1(d).

First, it is acyclic. As we all know, if an IWF-net has a siphon that does not contain any source place, then the IWF-net is unsound (because all transitions associated with the siphon are dead at the initial marking) and this siphon must contain a circuit. Therefore, an SIWF-net is required to be acyclic, which guarantees that each siphon includes at least one source place.

Second, each basic WF-net is sound and free-choice. For an IWF-net combined by multiple basic WF-nets via a group of channel places, what this paper pays more attention to is the interaction among these basic WF-nets. Therefore, we suppose that these basic WF-nets are sound. In addition, FCWF-nets can not only model many basic structures of workflow, such as AND-split, AND-join, OR-split, and OR-join, but also own a nice property (i.e., their soundness is decidable in polynomial time [25]). Since an SIWF-net is acyclic, each basic FCWF-net of it is also acyclic. Therefore, SIWF-net cannot model an iterative structure.

Finally, Definition 7 considers the most simple case of using a message channel; that is, for a fixed channel place, a message is sent into it by firing a unique transition and the message is taken away from it by firing another unique transition. Certainly, a transition may use multiple channel places.

It is worthy to note that this paper does not require an SIWF-net to observe the patterns proposed in [16] although these patterns can make an SIWF-net much simpler.

Although SIWF-nets seem simple, they can model many interaction cases except for iterative structure. Indeed, iterative structure is sometimes frustrating. Some scholars consider limited iterations so that they can be unfolded into iteration-free structures [8]. Therefore, it is valuable to explore efficient decision methods for SIWF-nets. Next, a novel one is given.

4. Sufficient and Necessary Condition for Compatibility of SIWF-Nets

This section gives a net-structure-based condition to decide compatibility for SIWF-nets. First, some concepts related to the net structures are defined.
Definition 8 (T-component of FCWF-net). Let \( N = (P, T, F) \) be an acyclic FCWF-net. Denote \( i \in P \) and \( o \in P \) by the source and sink places of \( N \), respectively. Then \( N' = (P', T', F') \) is a T-component of \( N \) if \( N' \) is a subnet of \( N \) such that

1. \( P' = T' \cup T'' \); that is, for all \( t \in T' \), the preset and postset of \( t \) in \( N' \) are the same as the preset and postset of it in \( N \), respectively;
2. \( i \in P' \land o \in P' \land \mid T' \cap T' \mid = \mid o \cap T' \mid = 1 \); and
3. for all \( p \in P' \setminus \{i, o\} : \mid p' \cap T' \mid = \mid p \cap T' \mid = 1 \).

Example 9. Figure 2 shows all T-components of the two FCWF-nets in Figures 1(c) and 1(d).

Notice that because of for all \( t \in T' \); \( T' \subseteq P' \) and for all \( p \in P' : \mid p \mid = \mid p' \mid = 1 \), there always exists a path in \( N' \) from \( i \) to \( x \) for any node \( x \) of \( N' \). Therefore, if a transition is added to a T-component such that its input is exactly the sink place and its output is exactly the source place, then a marked graph is produced. This shows that the definition of T-component coincides with the traditional one [26]. Therefore, a sound acyclic FCWF-net is covered by T-components according to the conclusion in [26]. Here, an FCWF-net is covered by T-components if and only if for each transition there is a T-component containing it. In addition, a complete transition sequence of a sound acyclic FCWF-net corresponds to a T-component. Here, a transition sequence is complete if and only if the marking \( M_o = o \) reached from \( M_i = i \) by firing it. van der Aalst proved that a sound FCWF-net is safe [25]. Therefore, each transition of a T-component occurs once and only once and each place of it is marked once and only once when the corresponding complete transition sequence is fired. Notice that a T-component may correspond to multiple complete transition sequences due to the parallel structure (see the examples in Figures 2(c)--2(f)).

Definition 10 (cap of FCWF-net). Let \( N = (P, T, F) \) be an acyclic FCWF-net and let \( i \) be its source place. Then \( N' = (P', T', F') \) is a cap of \( N \) if \( N' \) is a subnet of \( N \) such that

1. \( P' = T' \cup T'' \); that is, for all \( t \in T' \), the preset and postset of \( t \) in \( N' \) are the same as the preset and postset of it in \( N \), respectively;
2. for all \( p \in P' : \mid p \mid = \mid p' \mid = 1 \) in \( N' \); and
3. for all \( t \in T' \), there is a path from \( i \) to \( t \) in \( N' \).

Example 11. Figure 3 shows all caps related to the T-component in Figure 2(f).

Obviously, each T-component is a cap that represents some complete transition sequences. The subnet only containing the source place is also viewed as a cap that reflects the empty transition sequence. Next it is shown that for a sound acyclic FCWF-net each cap is a prefix of some T-component.

Lemma 12. A sound acyclic FCWF-net is covered by T-components and for each cap there exists a T-component such that the cap is a subnet of the T-component.

Proof. Let \( \sigma \) be a complete transition sequence, that is, \( i \sigma o \). Because the FCWF-net is sound, it is safe for the initial marking \( M_0 = i \) (see [25]). Because it is acyclic, each transition in \( \sigma \) occurs once only. Therefore, the transitions in \( \sigma \) and their preset and postset directly form a T-component. Because the FCWF-net is sound, for each transition there exists a complete transition sequence containing it, thereby a T-component containing it. Therefore, the FCWF-net is covered by T-components.

For each cap an enabled transition sequence \( \sigma \) is obtained such that \( \sigma \) contains all transitions of the cap and each transition of the cap occurs in \( \sigma \) once only. Because the acyclic FCWF-net is sound and safe, there exists a complete transition sequence \( \sigma' \) such that \( \sigma \) is a prefix of \( \sigma' \). Therefore, the cap is a subnet of the T-component that corresponds to \( \sigma' \).

In fact, an acyclic FCWF-net is also sound if it is covered by T-components and for each cap there exists a T-component containing it. Because this conclusion is not used in the proofs of other conclusions and only Lemma 12 is used,
the related proof is not described any more, which is very similar to the proof of the sufficiency of Theorem 20. This paper only gives the following conclusion that may be seen as a special case of Theorem 20.

**Theorem 13.** An acyclic FCWF-net is sound if and only if

1. it is covered by T-components;
2. for each cap there exists a T-component such that the cap is a subnet of the T-component.

**Definition 14** (T-component of SIWF-net). Let \( N = (N_1, N_2, \ldots, N_m, P_C, F_C) \) be an SIWF-net. Then \( N' = (N'_1, N'_2, \ldots, N'_m, P'_C, F'_C) \) is a T-component of \( N \) if

1. for all \( j \in [m], N'_j = (P'_j, T'_j, F'_j) \) is a T-component of \( N_j = (P_j, T_j, F_j) \) and is called basic;
2. \( P'_C = \bigcup_{j=1}^m (T'_j \cup T'_j) \cap P_C \); where \( T'_j \) and \( T'_j \) represent the preset and postset of \( T'_j \) in \( N_j \), respectively;
3. \( F'_C = \bigcup_{j=1}^m ((P'_j \times T'_j) \cup (T'_j \times P'_j)) \cap F_C \);
4. for all \( c \in P'_C; \# c = |c'| = 1 \) in \( N'_i \).

**Example 15.** The SIWF-net in Figure 1(a) has two and only two T-components as shown in Figures 4(a) and 4(b). The SIWF-net in Figure 1(b) also has two and only two T-components as shown in Figures 4(c) and 4(d).

A T-component of an SIWF-net includes one and only one T-component of each basic FCWF-net. Additionally, all channel places connected to those basic T-components are also in the T-component of the SIWF-net and must satisfy the closure property (i.e., if the input transition of a channel place is in this T-component, then its output transition must also be in this T-component, and vice versa). From the above examples it can be seen that not every basic T-component is contained in a T-component of the SIWF-net. That is to say, some behaviors of some basic WF-nets are inhibited. For an example of the SIWF-net in Figure 1(a), there are no T-components containing the basic T-component in Figure 2(c).

If a transition is added to a T-component of an SIWF-net such that its inputs are exactly all sink places and its outputs are exactly all source places, then the new net is a marked graph. This marked graph is strongly connected and each circuit contains a source place. Therefore, this marked graph is live and safe at the initial marking (i.e., if the input transition of a channel place is in this T-component, then its output transition must also be in this T-component, and vice versa). From the above examples it can be seen that not every basic T-component is contained in a T-component of the SIWF-net. That is to say, some behaviors of some basic WF-nets are inhibited. For an example of the SIWF-net in Figure 1(a), there are no T-components containing the basic T-component in Figure 2(c).

If a transition is added to a T-component of an SIWF-net such that its inputs are exactly all sink places and its outputs are exactly all source places, then the new net is a marked graph. This marked graph is strongly connected and each circuit contains a source place. Therefore, this marked graph is live and safe at the initial marking (i.e., if the input transition of a channel place is in this T-component, then its output transition must also be in this T-component, and vice versa). From the above examples it can be seen that not every basic T-component is contained in a T-component of the SIWF-net. That is to say, some behaviors of some basic WF-nets are inhibited. For an example of the SIWF-net in Figure 1(a), there are no T-components containing the basic T-component in Figure 2(c).
Definition 16 (cap of SIWF-net). Let $N = (N_1, N_2, \ldots, N_m, P_C, F_C)$ be an SIWF-net. Then $N' = (N'_1, N'_2, \ldots, N'_{m'}, P'_{C'}, F'_{C'})$ is a cap of $N$ if

1. for all $j \in \mathbb{N}_m$, $N'_j = (P'_j, T'_j, F'_j)$ is a cap of $N_j = (P_j, T_j, F_j)$;
2. $F'_C = \bigcup_{j=1}^{m'} (T'_j \cup T''_j) \cap P_C$, where $T'_j$ and $T''_j$ represent the preset and postset of $T'_j$ in $N$, respectively;
3. $F'_C = \bigcup_{j=1}^{m'} (T'_j \times T'_j) \cap (T'_j \times P'_j) \cap F_C$;
4. for all $c \in P'_C$: $|c| = 1 \Rightarrow c' \in N'$.\[\]

Example 17. Figure 5(a) is a cap of the SIWF-net in Figure 1(b), which is a subnet of the $T$-component in Figure 4(c). Although Figure 5(b) is also a cap of the SIWF-net in Figure 1(b), there are no $T$-components containing it. Figure 5(c) is not a cap of the SIWF-net in Figure 1(b) because the input transition of the channel place $c_3$ is not in the subnet; that is, it does not fulfill the fourth condition in Definition 16. Figure 5(d) is not a cap of the SIWF-net in Figure 1(b) because $c_3$, as the output of $t_{5,5}$ and the input of $t_{5,5}$, is not in the subnet; that is, it does not satisfy the second condition of Definition 16.

Notice that each $T$-component of an SIWF-net is also a cap of the SIWF-net and each cap of an SIWF-net contains one and only one cap of each basic FCWF-net.

Lemma 18. Let $N = (N_1, N_2, \ldots, N_m, P_C, F_C)$ be an SIWF-net.

1. (For each enabled transition sequence $\sigma$ in $(N, i_1 + i_2 + \cdots + i_m)$, i.e., $(i_1 + i_2 + \cdots + i_m)[\sigma]$), there is a cap $N' = (N'_1, N'_2, \ldots, N'_m, P'_{C'}, F'_{C'})$ of $N$ such that the cap is produced by the transitions in $\sigma$ as well as their pre- and postsets.
2. (For each cap $N' = (N'_1, N'_2, \ldots, N'_m, P'_{C'}, F'_{C'})$ of $N$ that exists an enabled transition sequence $\sigma$ in $(N, i_1 + i_2 + \cdots + i_m)$ such that each transition in $\sigma$ occurs once and only once in $\sigma$ and the transitions in $\sigma$ are just those in $N'_j$.\[\]

Proof. (1) Because $\sigma$ is an enabled transition sequence in $(N, i_1 + i_2 + \cdots + i_m)$, $\sigma \uparrow N_j$ is also an enabled transition sequence in $(N_j, i_j)$ for each $j \in \mathbb{N}_m$ where $\sigma \uparrow N_j$ is the projection of $\sigma$ over the transitions of $N_j$. By the proof of Lemma 12 it is known that there is a $N'_j$ of $N_j$ corresponding to $\sigma \uparrow N_j$. Because each transition $t$ in $\sigma$ is fired, each channel place, which is an input of $t$, must be an output of some transition $t'$ where $t'$ is fired earlier than $t$ in $\sigma$. Therefore, the transitions of $\sigma$ and their preset and postset can form a subnet of $N$ and this subnet satisfies the conditions in Definition 16.

(2) Because the cap $N' = (N'_1, N'_2, \ldots, N'_m, P'_{C'}, F'_{C'})$ is acyclic and each place that has no inputs (i.e., the source places) is marked, there are enabled transitions in $(N'_1, i_1 + i_2 + \cdots + i_m, F'_{C'})$. One of these enabled transitions is selected to fire and then delete this fired transition as well as its input places.

Then, the new Petri net is also acyclic and each place that has no inputs is also marked. By doing the above operations until the net has no transitions, a transition sequence is obtained that satisfies the conditions of the lemma.

Definition 19 (covered). An SIWF-net is covered by $T$-components if for each transition of it there is a $T$-component of it containing the transition.

Theorem 20. An SIWF-net $N = (N_1, N_2, \ldots, N_m, P_C, F_C)$ is compatible if and only if

1. (it is covered by $T$-components);
2. (for each cap $T$-component such that the cap is a subnet of the $T$-component).

Proof. (⇒) Let $\sigma$ be a complete transition sequence, that is, $(i_1 + i_2 + \cdots + i_m)[\sigma]$. Then, for all $j \in \mathbb{N}_m$, $\sigma \uparrow N_j$ is a complete transition sequence of $(N_j, i_j)$ where $\sigma \uparrow N_j$ is the projection of $\sigma$ over the transitions of $N_j$. By Lemma 12 it is known that $\sigma \uparrow N_j$ corresponds to a $T$-component of $N_j$. Additionally, all channel places have no tokens after firing $\sigma$ because $N$ is compatible. Therefore, a channel place as an output of $\sigma$ must be an input of $\sigma$, and vice versa. Therefore, all transitions of $\sigma$ and their preset and postset form a $T$-component of $N$ by Definition 14. Because $N$ is sound, for each transition there is a complete transition sequence containing it, thereby a $T$-component containing it. Therefore, $N$ is covered by $T$-components.
By the second conclusion in Lemma 18 it is known that, for each cap $N^1 = (N'_1, N'_2, \ldots, N'_{m'}, P'_0, F'_0)$ of $N$, there exists an enabled transition sequence $\sigma$ in $(N, i_1 + i_2 + \cdots + i_m)$ such that the transitions in $\sigma$ are exactly those in $N'_1$. Because $N$ is compatible, there is a complete transition sequence $\sigma'$ such that $\sigma$ is a prefix of $\sigma'$. By the first conclusion in Lemma 18 we can construct a cap $N''$ for $\sigma'$. Obviously, $N'$ is a subnet of $N''$ and $N''$ is a $T$-component.

($\Leftarrow$) (by contradiction) It is assumed that $N$ is incompatible. Then, by Definition 6 we know that one of the following three cases must take place: Case (1) $\exists t \in T$, for all $M \in R(N, i_1 + i_2 + \cdots + i_m): M[t]$; or Case (2) there is a marking $M \in R(N, i_1 + i_2 + \cdots + i_m)$ that marks not only all sink places but also some channel places; or Case (3) there is marking $M \in R(N, i_1 + i_2 + \cdots + i_m)$ that does not mark all sink places but all transitions are dead.

Case 1 does not hold: because $N$ is covered by $T$-components, for each transition there is a $T$-component containing it. By Lemma 18 it is known that for each $T$-component there is an enabled transition sequence that contains all transitions of the $T$-component. Therefore, for all $t \in T, \exists M \in R(N, i_1 + i_2 + \cdots + i_m): M[t]$.

Case 2 does not hold either: let enabled transition sequence $\sigma$ lead to a marking that marks all sink places as well as some channel places. By the first conclusion in Lemma 18 it is known that there is a cap $N'$ corresponding to $\sigma$. Because each sink place is marked after firing $i_t$, $N'$ contains one $T$-component of each basic FCWF-net and these marked channel places have no outputs in $N'$. By the given condition it is known that for $N'$ there must exist a $T$-component of $N$ such that $N''$ is a subnet of the $T$-component. To produce such a $T$-component $N''$, we must add some $T$-components of some basic FCWF-nets into $N'$ in order to make those marked channel places have output, which makes the required "$T$-component" $N''$ containing multiple basic $T$-components that are from the same basic FCWF-net. This does not satisfy the requirements of the definition of $T$-component; that is, a $T$-component of an SIWF-net contains one and only one basic $T$-component of each basic FCWF-net (see Definition 14).

Case 3 holds neither: let $\sigma$ be an enabled transition sequence leading to a marking at which some sink places are not marked but all transitions have been dead. Similar to the analysis of Case 2, it is known that, for the cap related to $\sigma$, there are no $T$-components containing it. If there is a $T$-component containing the cap, then there must exist an enabled transition sequence $\sigma'$ such that $\sigma'$ corresponds to the $T$-component and $\sigma$ is a prefix of $\sigma'$. This contradicts that all transitions are dead after firing $\sigma$. 

Example 21. Figure 1(a) is compatible. It is easy to verify that it is covered by $T$-components (see Figures 4(a) and 4(b)) and each cap is contained in a $T$-component.

Example 22. Figure 1(b) is incompatible. Although it is covered by $T$-components (see Figures 4(c) and 4(d)), there is a cap (e.g., the cap in Figure 5(b)) such that any $T$-component does not contain it. This example corresponds to Case 3 in the proof of "$\Leftarrow$" of Theorem 20. The example in Figure 6(a) shows Case 2 in the proof of "$\Rightarrow$" of Theorem 20. It has two $T$-components as shown in Figures 6(b) and 6(c), but it is incompatible because it has a cap, just as shown in Figure 6(d), which is not a subnet of any $T$-component. Figure 6(e) demonstrates Case 1 in the proof of "$\Leftarrow$" of Theorem 20 in which the transition $t_3$ is dead at the initial marking.

Definition 23 (maximal cap of SIWF-net). A cap of an SIWF-net is maximal if there are no other caps properly containing it.

Example 24. Figure 5(b) is a maximal cap of Figure 1(b) and each of Figures 6(b)–6(d) is a maximal cap of Figure 6(a).

Notice that each $T$-component is a maximal cap. Therefore, the following conclusion is drawn.

Corollary 25. An SIWF-net is compatible if and only if

1. it is covered by $T$-components;
2. each maximal cap is a $T$-component.

Proof. ($\Rightarrow$) By Theorem 20 it is known that a compatible SIWF-net is covered by $T$-components. If a maximal cap is not a $T$-component, then there are no $T$-components containing the maximal cap by the definition of maximal (see Definition 23), thereby making the SIWF-net incompatible by Theorem 20. Therefore, each maximal cap is a $T$-component.

($\Leftarrow$) For each cap, there is a maximal cap containing it. Therefore, for each cap there exists a $T$-component containing it. Therefore, the SIWF-net is compatible by Theorem 20.

In fact, the decision conditions in Theorem 20 and Corollary 25 can be sharpened and a simpler one is obtained;
Lemma 26. If each maximal cap of an SIWF-net is a T-component, then the SIWF-net is covered by T-components.

Proof. (by contradiction) Assume that the SIWF-net is not covered by T-components; that is, there exists a transition \( t \) that does not belong to any T-component. Because each maximal cap is a T-component (i.e., for each cap there exists a T-component containing it), \( t \) does not belong to any cap. Thus, one of the following two cases must exist.

Case I. If \( P_C \cap t \neq \emptyset \) (i.e., \( t \) has input channel places), then for all \( t' \in (P_C \cap t) \setminus t \), \( t' \) does not belong to any cap either. This is because if \( \exists t'' \in (P_C \cap t) \setminus t \) belongs to some cap, then there is a maximal cap containing \( t'' \) (by Definition 23); thereby a T-component containing \( t'' \) exists (by the given condition); thus the T-component has to contain \( t \) (by the fourth item of Definition 14).

Case 2. If \( P_C \cap t = \emptyset \) (i.e., \( t \) has no input channel places), then for all \( t' \in \{ t : \} \setminus t \), \( t' \) does not belong to any cap either (notice that \( t \) and \( t' \) are in the same FCWF-net). The following two cases are used to explain the reason.

Case 2.1 \((t = \{ p \})\). If there is \( t' \in \{ p : \} \) such that there is a cap containing \( t' \), then we know by the second conclusion in Lemma 18 that there is an enabled transition sequence containing \( t' \). Thus, place \( p \) is marked after firing \( t' \), which leads to \( t \) being enabled. Therefore, it is known by the first conclusion in Lemma 18 that there is a cap containing \( t \). This contradicts the assumption that no cap contains \( t \). Therefore, for all \( t' \in \{ p : \} \setminus t \), \( t' \) does not belong to any cap.

Case 2.2 \((|t| > 1)\). It is known by the free-choice property that after all \( p \in t : p' = \{ t \} \). If there is \( t' \in \{ t : \} \) such that there is a cap containing \( t' \), then there must exist a T-component \( N' \) containing \( t' \) (by the given condition). Let \( p \in t'' \) and \( p'' = \{ t \} \). Then, it is known by Definition 14 that \( t' \) must belong to \( N' \). This also contradicts the assumption that \( t \) does not belong to any T-component. Therefore, for all \( t' \in \{ t : \} \setminus t \), \( t' \) does not belong to any cap.

The above cases indicate that if \( t \) does not belong to any cap, then there must exist a transition, as the input transition of some input place of \( t \), which does not belong to any cap either. Because the set of transitions is finite and the SIWF-net is acyclic, we finally find a transition such that (1) it has no input channel places, (2) only some source place is its input, and (3) it does not belong to any cap. However, this transition as well as its input and output forms a cap. A contradiction takes place. Therefore, the SIWF-net is covered by T-components if each maximal cap of an SIWF-net is a T-component. \( \square \)

Theorem 27. An SIWF-net is compatible if and only if each maximal cap is a T-component.

Proof. The necessity is derived directly by Corollary 25 and the sufficiency directly by Corollary 25 and Lemma 26. \( \square \)

Procedure 1

Corollary 28. An SIWF-net is compatible if and only if for each cap there is a T-component containing it.

Example 29. The SIWF-nets in Figures 1(b) and 6(a) are incompatible because there exist maximal caps (see Figures 5(b) and 6(d)) that are not T-components. The SIWF-net in Figure 6(e) is incompatible since each maximal cap of it is not a T-component. Notice that the first two examples are covered by T-components but the last one is not.

5. Algorithm to Solve the Compatibility Problem for SIWF-Nets

Here a recursive algorithm is developed to solve the compatibility problem for an SIWF-net based on the decision conditions in Theorem 27 and Corollary 28. Let \( N = (N_1, N_2, \ldots, N_m, P_C, F_C) \) be an SIWF-net, \( M_0 = \{ i_1, \ldots, i_m \} \) the set of all source places, \( M_d = \{ o_1, \ldots, o_m \} \) the set of all sink places, and \( N' \) a cap of \( N \). Notice that \( M_0 \) may be seen as the most basic cap. See Procedure 1.

Please notice that \( X \) means a reachable marking by firing all transitions in \( N' \) and \( Y \) records all transitions of \( N \) that are enabled at \( X \).

\( X \neq M_d \wedge Y = \emptyset \) represents that after firing all transitions in \( N' \) the system reaches a marking such that it is not the target marking and no transition is enabled at it. Therefore, \( X \neq M_d \wedge Y = \emptyset \) means that the current cap is maximal but not a T-component; that is, the SIWF-net is incompatible. In this case, therefore, the program outputs this counterexample and terminates early.

\( X \neq M_d \wedge Y \neq \emptyset \) means that \( N' \) is still not maximal. Therefore, for each bigger cap (i.e., \( N'' := N' \cup \{ t \} \cup \{ t \times t \} \) this procedure is recursively called to do the same decision. Please note that \( N' \cup \{ t \} \cup \{ t \times t \} \) means that the enabled transition \( t \) as well as all places and arcs related to it in \( N \) is added to \( N' \).

\( X = M_d \) means that the current procedure will end correctly and return the previous layer correctly.

Also please notice that if two transitions \( t_1 \) and \( t_2 \) in \( Y \) can be concurrently fired at \( X \), then the procedure \( IsT-component(N, N'' \) will be called twice where \( N'' \) is the net
generated by adding $t_1$ and $t_2$ as well as the related places and arcs to $N'$. To avoid those repeated calls, $Y$ should be reduced such that for those transitions that can be concurrently fired at $X$ only one is left in $Y$. This operation can be completed in polynomial time and is omitted here for the simplification of this algorithm.

Therefore, when this procedure is called in a main procedure, $N'$ should be $M_0$ (i.e., it is the most basic cap). When IsT-component($N, M_0$) is executed, the worst case is that all maximal caps are $T$-components; that is, each maximal cap is tested. The number of $T$-components grows exponentially for the following very special example: an SIWF-net is composed of $m$ basic FCWF-nets, each basic FCWF-net has $k$ $T$-components, and there are no channel places. The SIWF-net has $k^m$ $T$-components. In practice, however, the number of maximal caps of an SIWF-net does not increase like the above example. If the number $n$ of maximal caps grows polynomially but not exponentially, then the number of recursively calling the procedure is polynomial, that is, $O(|T| \times n)$.

6. Conclusion

This paper gives a necessary and sufficient condition to decide compatibility for a subclass of IWF-nets, which advances the state-of-the-art in the area of deciding the compatibility problem based on net structures. Future work may focus on some bigger subclasses of IWF-nets in which each basic WF-net may permit circuits.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This paper is supported in part by the Alexander von Humboldt Foundation and in part by the National Nature Science Foundation of China (Grant no. 61202016).

References


