Relaxed Soundness Verification for Interorganizational Workflow Processes

Lígia Maria Soares Passos[1,2] and Stéphane Julia[2]

[1]*Computer Science Department, Federal Rural University of Rio de Janeiro, Nova Iguaçu, Rio de Janeiro, Brazil*

[2]*Computing Faculty, Federal University of Uberlândia, Uberlândia, Minas Gerais, Brazil*

Keywords: Interorganizational Workflow Processes, Relaxed Soundness, Verification, Linear Logic, Petri Nets.

Abstract: This paper presents a method for the *Relaxed Soundness* verification of interorganizational workflow processes. The method considers Interorganizational WorkFlow net models and is based on the analysis of Linear Logic proof trees. To verify the *Relaxed Soundness* criterion, a Linear Logic proof tree is built for each different scenario of an *unfolded* Interorganizational WorkFlow net. These proof trees are then analysed considering two conditions: the first verifies if the analysed scenario can finish properly, without spare tokens and the second verifies if every activity concerning the global process was covered by at least one possible scenario. The Interorganizational WorkFlow net is then considered as relaxed sound if the scenarios satisfy these conditions.

1   INTRODUCTION

Workflow processes that involve several business processes belonging to different organizations and which need to coordinate their actions in order to reach a common goal are known as interorganizational workflow processes (Captarencu, 2012).

According to (van der Aalst, 1998b), an interorganizational workflow is essentially a set of loosely coupled workflow processes where, typically, there exist local workflow processes that are involved in one global workflow process. These local workflow processes need to communicate for the correct execution of those cases treated by the global workflow management system.

According to (Lim et al., 2012), interorganizational workflow systems play a fundamental role in business partnerships and forming an alliance with appropriate business partners is a common strategy for enterprises to remain competitive by offering a wider range of products and services to its clients.

Many studies have already considered the qualitative analysis of interorganizational workflow processes. In (van der Aalst, 1998b; Yamaguchi et al., 2007; Sun and Du, 2008; Soares Passos and Julia, 2013) for example, the proposed approaches are related to classical Soundness verification, a qualitative property of Interorganizational WorkFlow nets. The Soundness correctness criterion considers the interorganizational workflow process as a whole, i.e if it

does not satisfy Soundness, it needs to be redesigned to satisfy Soundness and guarantee that the model is deadlock-free, for example. In (van der Aalst, 1998b; Yamaguchi et al., 2007) and (Sun and Du, 2008), the proposed approaches for classical Soundness verification of interorganizational workflow processes are based on the construction and analysis of reachability graphs. In (Soares Passos and Julia, 2013), a Linear Logic based approach is presented for classical Soundness verification in the context of interorganizational workflow processes. The approach presented in (Soares Passos and Julia, 2013) is based on the construction and analysis of Linear Logic proof trees. It is important to highlight that these studies are concerned with the classical Soundness verification for interorganizational workflow processes and the Relaxed Soundness verification for interorganizational workflow processes is not taken into account.

The ideal scenario is the one in which the interorganizational processes are sound, once that Soundness ensures important criteria, such as absence of deadlock and proper termination. However, according to (Fahland et al., 2011), the checking of 735 industrial business process models from financial services, telecommunications, and other domains has shown that only 46% of the process models were sound. So, as an interorganizational workflow is essentially a set of local workflow processes involved in one global workflow process (van der Aalst, 1998b), and a large percentage of these local workflow processes are un-

sound (Fahland et al., 2011), the global workflow process that is based on unsound local workflow processes will also be unsound according to the Soundness definition proposed by (van der Aalst, 1998b). In this context, a wide variety of interorganizational workflow processes may be unsound and may lead the services of the business organization to deadlock situations. Furthermore, workflow processes are important assets for the organizations that are not always willing to redesign or adjust their processes to fully fit the specific need of a specific partner, specially as different business partners may have different needs.

In the context of single workflow processes, i.e. workflow processes that are not interorganizational, (Dehnert and Rittgen, 2001) proposed to relax the Soundness criterion. The new defined criterion is the Relaxed Soundness criterion. The idea behind Relaxed Soundness is that the system's behavior is correct if there exist sufficient executions which terminate properly (Dehnert and Rittgen, 2001). So, the notion of Relaxed Soundness ensures that there is at least one run that enables each task of the workflow process model which can be carried from the initial state forward to the final state. In (Siegeris and Zimmermann, 2006), various workflow model composition proposals are summarized and the authors investigate the ability of these composition mechanisms to preserve the Relaxed Soundness criterion. However, to preserve Relaxed Soundness, the workflow processes that are used in the composition have to satisfy the Relaxed Soundness criterion too. So, if this is not the case, these workflow processes have to be redesigned first to satisfy the Relaxed Soundness criterion before they can be used in a composition.

Considering that a wide variety of interorganizational workflow processes may be unsound and that the idea behind the Relaxed Soundness is that the system's behavior is correct if there exist sufficient executions which terminate properly, it is of great interest to verify Relaxed Soundness in the context of interorganizational workflow processes. In these circumstances, i.e. in the cases in which the interorganizational workflow processes are unsound, the Relaxed Soundness criterion ensures that the main business relationship between the involved organizations can be provided safely, with no obligation to the redesigning of the involved processes in order that they satisfy the Soundness criterion. So, the approach presented in this paper considers the Relaxed Soundness verification for an interorganizational workflow process, where the set of local workflow processes are not necessarily Relaxed Sound and the global workflow process is or may be unsound.

Therefore, this paper presents a method for Relaxed Soundness verification for interorganizational workflow processes modelled by Interorganizational WorkFlow nets (IOWF-nets) (van der Aalst, 1998b). Thus, the organizations involved in the interorganizational workflow process will be able to verify if their main business services can finish properly, considering the global process, avoiding deadlock situations whenever they occur, without redesigning their local or global workflow processes. This method is based on the analysis of Linear Logic proof trees built considering each scenario of the *unfolded* IOWF-net (van der Aalst, 1998b).

This paper is structured as follows. In section 2 the definition of the Interorganizational WorkFlow nets is provided. In section 3, an overview of Linear Logic is given. The method for Relaxed Soundness verification for Interorganizational WorkFlow nets is presented in section 4. Finally, the last section concludes this work with a short summary, an assessment of the presented approach and an outlook on future work proposals.

2 INTERORGANIZATIONAL WORKFLOW NETS

In this section, the concepts related to Interorganizational WorkFlow nets (IOWF-nets) are presented. These concepts are necessary to better comprehend the approach presented in section 4.

To define IOWF-nets, it is necessary first to introduce the definition of WorkFlow nets. According to (van der Aalst, 1998a), a Petri net (Murata, 1989) that models a workflow process is called a WorkFlow net (WF-net). A WF-net satisfies the following properties (van der Aalst, 1998a): it has only one source place named i and only one sink place named o, that are special places such that the place i has only outgoing arcs and the place o has only incoming arcs; a token in i represents a case that needs to be handled and a token in o represents a case that has been handled; every task t (transition) and condition p (place) should be on a path from place i to place o.

Following, the formal definition of WF-nets is presented.

Definition 1 (WorkFlow-net). *A Petri net $PN = \{P,T,F\}$, where P is a finite set of places, T is a finite set of transitions ($P \cap T = \emptyset$) and $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs (flow relation), is a WF-net if, and only if (van der Aalst, 1998a):*

1. PN has two special places: i and o. Place i is a source place: $\bullet i = \emptyset$. Place o is a sink place: $o \bullet = \emptyset$.

2. Every node is on a path from i to o.

An Interorganizational WorkFlow net (IOWF-net) is a Petri net that models an interorganizational workflow process and can be seen as a global workflow process that has *n* business partners involved in it, according to (van der Aalst, 1998b). Each partner has its own local workflow process. So, an interorganizational workflow is composed of at least two local workflow processes. Thus, an IOWF-net is composed of at least two Local WorkFlow nets (LWF-nets). In (van der Aalst, 1998b), a global workflow process consists of a set of local workflow processes plus an interactive structure composed of asynchronous and synchronous communication mechanisms. According to (van der Aalst, 1998b), synchronous communication corresponds to the melting (fusion) of some transitions and the asynchronous communication corresponds to the exchange of messages between local workflow processes. In this paper, the synchronous case will not be considered, since we consider that each organization controls its own process so that there is no melting of transitions. Therefore, only asynchronous communication protocols will be represented. Following this, the IOWF-net definition proposed by Aalst in (van der Aalst, 1998b) is specially adapted to the asynchronous case.

Definition 2 (IOWF-net). *An Interorganizational WorkFlow net (IOWF-net) is a tuple $IOWF - net = \{PN_1, PN_2, ..., PN_n, P_{AC}, AC\}$, where:*

1. $n \in \mathbb{N}$ is the number of LWF-nets;

2. for each $k \in \{1, ..., n\}$, PN_k is a WF-net with a source place i_k and a sink place o_k;

3. for all $k, l \in \{1, ..., n\}$, if $k \neq l$, then $(P_k \cup T_k) \cap (P_l \cup T_l) = \emptyset$;

4. $T^* = \bigcup_{k \in \{1, ..., n\}} T_k$, $P^* = \bigcup_{k \in \{1, ..., n\}} P_k$ and $F^* = \bigcup_{k \in \{1, ..., n\}} F_k$ (relations between the elements of the LWF-nets);

5. P_{AC} is the set of asynchronous communication elements (communication places);

6. $AC \subseteq P_{AC} \times \mathbb{P}(T^*) \times \mathbb{P}(T^*)$ represents the asynchronous communication relations[1].

Each asynchronous communication element corresponds to a place name in P_{AC} and the asynchronous communication relation *AC* specifies a set of input transitions and a set of output transitions for each asynchronous communication element (van der Aalst, 1998b).

To clarify the concepts defined above, the interorganizational workflow process presented in (van der

---
[1] $\mathbb{P}(T^*)$ is the set of all non-empty subsets of T^*

Aalst, 1998b) is contemplated. Such a process models a process that precedes the presentation of a paper at a conference and its description can be found in (van der Aalst, 1998b). The highlighted IOWF-net area in Figure 1 shows the IOWF-net that models that process. This IOWF-net has two LWF-nets: Author and PC (Program Committee). Each of these has only one source and one sink place. In the LWF-net Author, the source place is *start_flow_author* and the sink place is *end_flow_author*. In the LWF-net PC, the source and sink place are *start_flow_PC* and *end_flow_PC*, respectively. The places *draft*, *ack_draft*, *reject*, *accept*, *too_late*, *final_version* and *ack_final* are examples of asynchronous communication places.

In (van der Aalst, 1998b), the Unfolded Interorganizational WorkFlow net is defined. The *unfolding* of an IOWF-net is a WF-net. In the *unfolded* net, i.e. the *U*(IOWF-net), all the LWF-nets are included into a single workflow process considering a start transition t_i and a termination transition t_o. A global source place *i* and a global sink place *o* have to be added in order to respect the basic structure of a simple WF-net, and the asynchronous communication elements are mapped into ordinary places according to (van der Aalst, 1998b). Figure 1 shows an *U*(IOWF-net).

In (Dehnert and Rittgen, 2001), the authors proposed to relax the Soundness criterion, a well-known criterion defined by (van der Aalst, 1998a), to a new criterion named Relaxed Soundness. They argue that this criterion is closer to the intuition of the modeller. According to (Dehnert and Rittgen, 2001), Relaxed Soundness is intended to represent a more pragmatic view on correctness which is weaker (in a formal sense) than the Soundness criterion. To (Dehnert and Rittgen, 2001), Relaxed Soundness means that there exist sufficient executions that terminate properly (i.e. without spare tokens). In this context sufficient means, according to (Dehnert and Rittgen, 2001), each transition of the process is covered at least once when considering the set of sound firing sequences.

The definition of Relaxed Soundness, proposed by (Dehnert and Rittgen, 2001), is the following.

Definition 3 (Relaxed Soundness). *A process specified by a WF-net PN = (P, T, F) is relaxed sound if and only if every transition is in a firing sequence that starts in state i and ends in state o.*

Formally:

$\forall t \in T : \exists M, M' : (i \xrightarrow{*} M \xrightarrow{t} M' \xrightarrow{*} o)$, *where M and M' are markings.*

The Relaxed Soundness criterion was then defined in the context of WF-nets only and the IOWF-nets were not formally taken into account. However, this
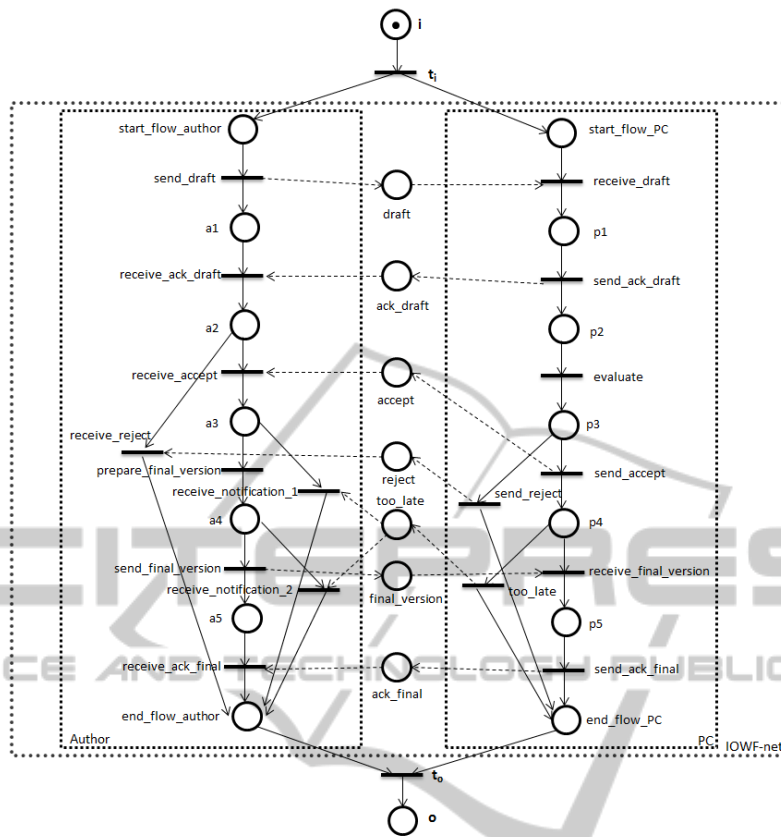
Figure 1: An unfolded IOWF-net.

criterion is also important in the context of IOWF-nets, specially in the cases where the Soundness criterion is not satisfied. Therefore, as the *unfolding* of an IOWF-net, the U(IOWF-net), has the same structure of a WF-net, as shown in (van der Aalst, 1998b), we can verify the Relaxed Soundness criterion for the IOWF-nets, considering the analysis of its *unfolded* net.

3 LINEAR LOGIC

In this section, an overview of Linear Logic is presented. The concepts presented here are necessary for a better comprehension of the method presented in the next section.

The first proposals for Linear Logic were made in (Girard, 1987). In Linear Logic, propositions are considered as resources, i.e. atoms, which are consumed and produced at each state change (Riviere et al., 2001). Linear Logic introduces new connectives. In this paper just two Linear Logic connectives will be used:

- The *times* connective, denoted by $\otimes$, that repre-

sents simultaneous availability of resources. For instance, $A \otimes B$ represents the simultaneous availability of resources A and B.

- The *linear implies* connective, denoted by $\multimap$, that represents a state change. For instance, $A \multimap B$ denotes that by consuming A, B is produced (it is important to note that after the production of B, A will not be available).

The translation of a Petri net into formulas of Linear Logic, presented in (Riviere et al., 2001), is the following. A marking M is a monomial in $\otimes$ and is represented by $M = A_1 \otimes A_2 \otimes \ldots \otimes A_k$ where A_i are place names. For instance, the initial marking on the U(IOWF-net) in Figure 1 is simply i because of the token in place i. A transition is an expression of the form $M_1 \multimap M_2$ where M_1 and M_2 are markings. For example, transition *evaluate* of the LWF-net PC in Figure 1 is noted $evaluate = p2 \multimap p3$.

A sequent $M, t_k \vdash M'$ represents a scenario where M and M' are respectively the initial and final markings, and t_k is a list of non-ordered transitions. For instance, considering the U(IOWF-net) shown in Figure 1, the sequent i, t_i, *send_draft*, *receive_draft*, *send_ack_draft*, *receive_ack_draft*,

evaluate, *send_reject*, *receive_reject*, $t_o \vdash o$ represents one possible scenario of this U(IOWF-net), where i is the initial marking, t_i, *send_draft*, *receive_draft*, *send_ack_draft*, *receive_ack_draft*, *evaluate*, *send_reject*, *receive_reject*, t_o is a list of non-ordered transitions and o is the final marking. A sequent can be proven by applying the rules of the sequent calculus. It was proven in (Girault et al., 1997) that a proof of the sequent calculus is equivalent to the corresponding reachability problem in a Petri net.

In this paper, only some rules of Linear Logic will be considered. These rules will be used to build the proof trees. In this context, F, G, and H are formulas and Γ and Δ are considered blocks of formulas. The following rules will be those used in the present paper (Riviere et al., 2001):

- The $\multimap_L$ rule, $\dfrac{\Gamma \vdash F \quad \Delta, G \vdash H}{\Gamma, \Delta, F \multimap G \vdash H}$ $\multimap_L$, expresses a transition firing and generates two sequents such that the right-hand sequent represents the subsequent which remains to be proven and the left-hand sequent represents the tokens consumed by this particular firing. For example, considering the transition $t_i = i \multimap start\_flow\_author \otimes start\_flow\_pc$ of the U(IOWF-net) shown in Figure 1, when this transition is fired, two sequents are generated: $i \vdash i$ represents the token consumed by this firing and $start\_flow\_author \otimes start\_flow\_pc$ the remaining subsequent to be proven.

- The $\otimes_L$ rule, $\dfrac{\Gamma, F, G \vdash H}{\Gamma, F \otimes G \vdash H}$ $\otimes_L$, transforms a marking in an atoms list. For example, the subsequent $start\_flow\_author \otimes start\_flow\_pc$ generated by the firing of transition $t_i = i \multimap start\_flow\_author \otimes start\_flow\_pc$ of the U(IOWF-net) shown in Figure 1 will use the rule $\otimes_L$ to be transformed into a list of atoms $start\_flow\_author, start\_flow\_pc$.

- The $\otimes_R$ rule, $\dfrac{\Gamma \vdash F \quad \Delta \vdash G}{\Delta, \Gamma \vdash F \otimes G}$ $\otimes_R$, transforms a sequent such as $A, B \vdash A \otimes B$, into two identity sequents $A \vdash A$ and $B \vdash B$. For example, considering the firing of the transition $receive\_accept = a2 \otimes accept \multimap a3$ of the U(IOWF-net) shown in Figure 1, the sequent that represents the tokens consumed by this firing, $a2, accept \vdash a2 \otimes accept$, also needs to be proven, using the $\otimes_R$ rule, i.e. $\dfrac{a2 \vdash a2 \quad accept \vdash accept}{a2, accept \vdash a2 \otimes accept}$ $\otimes_R$.

In the approach presented in this paper, a Linear Logic proof tree is read from the bottom up. The proof stops when the atom that represents the place o is produced, i.e. the identity sequent $o \vdash o$ appears in the proof tree, when there is not any rule that can be applied, or when all the leaves of the proof tree are identity sequents.

4 RELAXED SOUNDNESS VERIFICATION FOR IOWF-nets

To verify Relaxed Soundness for the IOWF-nets, it is necessary to build and prove linear sequents of Linear Logic. This approach considers the analysis of the *unfolded* IOWF-net, U(IOWF-net), which has the same structure of a WF-net. So, it is then necessary to built and prove linear sequents of Linear Logic that represent the U(IOWF-net).

Initially, the U(IOWF-net) has to be represented through the use of Linear Logic formulas. The U(IOWF-net) can be represented by more than one linear sequent, each linear sequent representing a possible scenario of the U(IOWF-net).

A scenario in the context of U(IOWF-nets), corresponds to a well defined route mapped into the corresponding U(IOWF-net). If the U(IOWF-net) has more than one route (places with two or more output arcs), it is necessary then to build a different linear sequent for each existing scenario (Soares Passos and Julia, 2013). For example, for the U(IOWF-net) shown in Figure 1, there exist five different scenarios: the first scenario, Sc_1, where task *send_reject* will be executed (firing of transition *send_reject*); the second scenario, Sc_2, where tasks *too_late* and *receive_notification_1* will be executed (firing of transitions *too_late* and *receive_notification_1*); the third scenario, Sc_3, where tasks *too_late* and *receive_notification_2* will be carried out (firing of transitions *too_late* and *receive_notification_2*); the fourth scenario, Sc_4, where tasks *send_final_version* and *receive_final_version* will be carried out (firing of transition *send_final_version* and *receive_final_version*) and the fifth scenario, Sc_5, where tasks *too_late* and *send_final_version* will be executed (firing of transitions *too_late* and *send_final_version*).

In this approach, each one of these scenarios is then represented by a specific linear sequent that considers the initial and final markings of the U(IOWF-net) and a non-ordered list of transitions involved in it. Each linear sequent has only one atom which represents the initial marking of the U(IOWF-net). For example, the scenario Sc_1 is represented by: i, t_i, *send_draft*, *receive_draft*, *send_ack_draft*, *receive_ack_draft*, *evaluate*, *send_reject*, *receive_reject*, $t\_o \vdash o$.

After the definition of the linear sequents that represent the different scenarios of the U(IOWF-net), the

linear sequents need to be proven through the building of Linear Logic proof trees. After the construction of these proof trees, each scenario of the analysed U(IOWF-net) must be analysed respecting the following steps:

1. For each proof tree that represents a scenario:

   (a) If just one atom o, that corresponds to an atom in the sink place of the U(IOWF-net), was produced in the proof tree (this is represented in the proof tree by the identity sequent $o \vdash o$), then the analysed scenario was finished properly.

   (b) If there is not any available atom to be consumed on the proof tree, i.e. all places are empty, then the execution terminates without spare tokens.

2. Considering the proof trees for scenarios $Sc_1, Sc_2, ..., Sc_i$ of the analysed U(IOWF-net) that satisfy step 1, each transition $t \in T$ needs to appear in, at least, one of these proof trees. This proves that all transitions were fired at least once and that every activity of the global process was covered by at least one possible scenario.

If the conditions 1 and 2 are satisfied, the analysed U(IOWF-net) is *Relaxed Sound*.

To illustrate the proposed approach, the IOWF-net shown in Figure 1 is considered. To prove Relaxed Soundness for this IOWF-net, the corresponding U(IOWF-net) is considered. It is necessary to prove five linear sequents, each one representing one of the following scenarios: Sc_1, Sc_2, Sc_3, Sc_4 and Sc_5.

The transitions of the U(IOWF-net) shown in Figure 1 are represented by the following formulas of Linear Logic:

$t_i = i \multimap start\_flow\_author \otimes start\_flow\_PC$,
$t_1 = send\_draft = start\_flow\_author \multimap a1 \otimes draft$,
$t_2 = receive\_ack\_draft = a1 \otimes ack\_draft \multimap a2$,
$t_3 = receive\_accept = a2 \otimes accept \multimap c3$,
$t_4 = receive\_reject = a2 \otimes reject \multimap end\_flow\_author$,
$t_5 = prepare\_final\_version = a3 \multimap a4$,
$t_6 = receive\_notification\_1 = a3 \otimes too\_late \multimap end\_flow\_author$,
$t_7 = send\_final\_version = a4 \multimap a5 \otimes final\_version$,
$t_8 = receive\_notification\_2 = a4 \otimes too\_late \multimap end\_flow\_author$,
$t_9 = receive\_ack\_final = a5 \otimes ack\_final \multimap end\_flow\_author$,
$t_{10} = receive\_draft = start\_flow\_PC \otimes draft \multimap p1$,
$t_{11} = send\_ack\_draft = p1 \multimap ack\_draft \otimes p2$,
$t_{12} = evaluate = p2 \multimap p3$,
$t_{13} = send\_reject = p3 \multimap reject \otimes end\_flow\_PC$,
$t_{14} = send\_accept = p3 \multimap accept \otimes p4$,
$t_{15} = too\_late = p4 \multimap too\_late \otimes end\_flow\_PC$,

$t_{16} = receive\_final\_version = final\_version \otimes p4 \multimap p5$,
$t_{17} = send\_ack\_final = p5 \multimap ack\_final \otimes end\_flow\_PC$,
$t_o = end\_flow\_author \otimes end\_flow\_PC \multimap o$.

For readability, the atoms of the linear sequents and proof trees will be represented just by the initial letters of their complete name. For example, *draft*, *too_late* and *final_version* will be represented by d, tl and fv, respectively. A set of transitions t_i, t_j, t_k and t_l may be also represented as $t_{i,j,k,l}$.

By considering the U(IOWF-net) shown in Figure 1, five different scenarios, and consequently linear sequents, are defined:

$Sc_1 = i, t_i, t_1, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o$,
$Sc_2 = i, t_i, t_1, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o$,
$Sc_3 = i, t_i, t_1, t_2, t_3, t_5, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o$,
$Sc_4 = i, t_i, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16,17,o} \vdash o$,
$Sc_5 = i, t_i, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o$.

Following are the proof trees for each one of these scenarios. The proof tree for scenario Sc_1 is as follows:

$$
\cfrac{
\cfrac{
\cfrac{
\cfrac{
\cfrac{
\cfrac{
\cfrac{
\cfrac{
\cfrac{
\cfrac{
\cfrac{
\cfrac{
\cfrac{efa \vdash efa \quad efp \vdash efp}{efa, efp \vdash efa \otimes efp} \otimes_R \quad o \vdash o}{efp, efa, efa \otimes efp \multimap o \vdash o} \multimap_L
}{\cfrac{a2 \vdash a2 \quad r \vdash r}{a2, r \vdash a2 \otimes r} \otimes_R \quad efp, efa, efa \otimes efp \multimap o \vdash o} \multimap_L}{...}
}{...}}{...}}{...}}{...}}{...}}{...}}{...}}{...}}{...}}{i, t_i, t_1, t_2, t_4, t_{10}, t_{11}, t_{12}, t_{13}, t_o \vdash o}
$$

For space reasons, just the first and the last linear sequent are shown in the next proof trees. So, the proof tree for scenario Sc_2 is as follows:

$$
\cfrac{o \vdash o}{\vdots} \multimap_L
$$
$$
\overline{i, t_i, t_1, t_2, t_3, t_6, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o}
$$

The proof tree for scenario Sc_3 is as follows:

$$\frac{o \vdash o}{\vdots} \multimap_L$$

$$\overline{i, t_i, t_1, t_2, t_3, t_5, t_8, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o}$$

The proof tree for scenario Sc_4 is the following one.

$$\frac{o \vdash o}{\vdots} \multimap_L$$

$$\overline{i, t_i, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, o \vdash o}$$

And finally the proof tree for scenario Sc_5 is as follows:

$$\frac{tl, efp, a5, fv, t_9, t_o \vdash o}{\vdots} \otimes_L$$

$$\overline{i, t_i, t_1, t_2, t_3, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{15}, t_o \vdash o}$$

The next step is to analyse the proof trees produced. By considering the proof trees for scenarios Sc_1, Sc_2, Sc_3, Sc_4 and Sc_5, it is necessary to verify the condition 1 for each scenario Sc_i and the condition 2 for the scenarios that satisfy condition 1. It is easy to note that the last sequent in the proof trees for scenarios Sc_1, Sc_2, Sc_3 and Sc_4 is $o \vdash o$. So, conditions 1a and 1b are satisfied, i.e. just one atom o was produced in these proof trees and as the last sequent is an identity sequent, there is not any available atom for consumption, i.e. the execution for these scenarios finishes without spare tokens. The last sequent for scenario Sc_5 is $tl, efp, a5, fv, t_9, t_o \vdash o$, as no atom o was produced by this scenario, the condition 1a is not verified for this scenario. This sequent also contains available atoms for consumption, as the atoms tl, efp, $a5$ and fv. Consequently, it does not satisfy the condition 1b. Therefore, for the second part of the verification (step 2), scenarios Sc_1, Sc_2, Sc_3 and Sc_4 will be considered. Each transition $t \in T$ appears in at least one of these scenarios. So, the condition 2 is also satisfied and the IOWF-net shown in Figure 1 is relaxed sound. The scenarios Sc_1, Sc_2, Sc_3 and Sc_4 are the ones that terminate properly. The scenario Sc_5 is the one where the process deadlocks.

According to (van der Aalst, 1998b), the classical Soundness verification for IOWF-nets is based on the proof of liveness and boundedness for $(n+1)$ WF-nets using standard techniques. According to (Dehnert and Rittgen, 2001), there exist no structural properties such as liveness and boundedness from which the Relaxed Soundness property can be derived. As presented in (Dehnert and Rittgen, 2001),

Relaxed Soundness can be proven only by enumeration of sufficient sound firing sequences. For this purpose, classical approaches based on reachability graphs have to find sound firing sequences for every transition (Dehnert and Rittgen, 2001). In particular, for classical Soundness verification of interorganizational workflow processes as the one presented in (van der Aalst, 1998b), if the analysed model is not sound, it is then necessary to re-analyse and re-explore the whole model to verify if it satisfies the Relaxed Soundness criterion.

The Linear Logic based approach presented in (Soares Passos and Julia, 2013) verifies classical Soundness for IOWF-nets considering the construction and analysis of the Linear Logic proof trees that represent each scenario of the Local WorkFlow nets and each scenario of the corresponding U(IOWF-net). The approach presented here to verify Relaxed Soundness considers the construction and analysis of the proof trees that represent each scenario of the U(IOWF-net), i.e. the building and analysis of a subset of the scenarios considered in the classical Soundness verification. So, when an approach based on Linear Logic to verify the Soundness criterion for IOWF-nets, as the one presented in (Soares Passos and Julia, 2013) is considered, if the analysed IOWF-net is unsound, a subset of the proof trees built to prove Soundness for the IOWF-net can be reused in the context of this approach, performing only the analysis steps of the proof trees that represent each scenario of the U(IOWF-net) to decide whether the analysed IOWF-net is relaxed sound. It is important to highlight that the approach presented in this paper relaxes the conditions of verification presented in (Soares Passos and Julia, 2013), as well as Relaxed Soundness relaxes the Soundness correctness criterion. And, although the reuse of Linear Logic proof trees is achieved, the verification methods are distinct.

5 CONCLUSIONS

This paper presented an approach for the *Relaxed Soundness* verification of interorganizational workflow processes modelled by Interorganizational WorkFlow nets (IOWF-nets). The approach was based on the construction and analysis of proof trees of Linear Logic that represent scenarios of the analysed *unfolded* IOWF-net. To verify Relaxed Soundness for an IOWF-net, it is necessary in particular to encounter all sound scenarios that allow the process to reach the final state of the global business process and to verify that every activity associated with the transition of the U(IOWF-net) appears at least once

in one of the encountered scenarios.

The advantages of such an approach are diverse. Initially, we extended the Relaxed Soundness criterion to the context of the interorganizational workflow processes to guarantee that the main business relationship between the involved organizations can be provided safely, with no obligation of redesigning the involved individual processes to satisfy the Relaxed Soundness criterion before the composition, for example.

The fact of working with Linear Logic permits one to prove the Relaxed Soundness criterion considering the proper structure of the IOWF-net, without considering the corresponding automata (reachability graph). Furthermore, when an approach based on Linear Logic to verify the Soundness criterion for IOWF-nets, as the one presented in (Soares Passos and Julia, 2013) is considered, if the analysed IOWF-net is unsound, a subset of the proof trees built to prove Soundness for the IOWF-net can be reused in the context of this approach. Thus, performing only the analysis steps in the proof trees that represent each scenario of the U(IOWF-net) to decide whether the analysed IOWF-net is relaxed sound.

As a future work proposal, it will be interesting to implement a kind of real time supervisory control able to follow the valid scenarios encountered during the execution of the workflow management system, avoiding in particular deadlock situations that may exist in the relaxed sound model, as is the case of scenario Sc_5 of the U(IOWF-net) shown in Figure 1. It is important since the Relaxed Soundness criterion does not ensure that the process is deadlock-free.

REFERENCES

Captarencu, O. (2012). Modelling and verification of interorganizational workflows with security constraints: A petri nets-based approach. In Bajec, M. and Eder, J., editors, *Advanced Information Systems Engineering Workshops*, volume 112 of *Lecture Notes in Business Information Processing*, pages 486–493. Springer Berlin Heidelberg.

Dehnert, J. and Rittgen, P. (2001). Relaxed soundness of business processes. In Dittrich, K., Geppert, A., and Norrie, M., editors, *Advanced Information Systems Engineering*, volume 2068 of *Lecture Notes in Computer Science*, pages 157–170. Springer Berlin Heidelberg.

Fahland, D., Favre, C., Koehler, J., Lohmann, N., Völzer, H., and Wolf, K. (2011). Analysis on demand: Instantaneous soundness checking of industrial business process models. *Data Knowl. Eng.*, 70(5):448–466.

Girard, J.-Y. (1987). Linear logic. *Theoretical Computer Science*, 50(1):1–102.

Girault, F., Pradin-Chézalviel, B., and Valette, R. (1997). A logic for Petri nets. *Journal Européen des Systèmes Automatisés*, 31(3):525–542.

Lim, H. W., Kerschbaum, F., and Wang, H. (Sept.-Oct. 2012). Workflow signatures for business process compliance. *Dependable and Secure Computing, IEEE Transactions on*, 9(5):756–769.

Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.

Riviere, N., Pradin-Chezalviel, B., and Valette, R. (2001). Reachability and temporal conflicts in t-time petri nets. In *Petri Nets and Performance Models, 2001. Proceedings. 9th International Workshop on*, pages 229–238.

Siegeris, J. and Zimmermann, A. (2006). Workflow model compositions preserving relaxed soundness. In *Proceedings of the 4th International Conference on Business Process Management*, BPM'06, pages 177–192, Berlin, Heidelberg. Springer-Verlag.

Soares Passos, L. M. and Julia, S. (2013). Qualitative analysis of interorganizational workflow nets using linear logic: Soundness verification. In *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, pages 667–673.

Sun, H. C. and Du, Y. Y. (2008). Soundness Analysis of Inter-Organizational Workflows. *Information Technology Journal*, 7:1194–1199.

van der Aalst, W. M. P. (1998a). The application of Petri nets to Workflow Management. In *The Journal of Circuits, Systems and Computers*, pages 21–66.

van der Aalst, W. M. P. (1998b). Modeling and analyzing interorganizational workflows. In *Application of Concurrency to System Design, 1998. Proceedings., 1998 International Conference on*, pages 262–272.

Yamaguchi, S., Matsuo, H., Ge, Q.-W., and Tanaka, M. (2007). Wf-net based modeling and soundness verification of interworkflows. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E90-A(4):829–835.