

A Verification Method for Collaborative Business Processes

Jorge Roa¹, Omar Chiotti², and Pablo Villarreal¹

¹ CIDISI, Universidad Tecnológica Nacional - Facultad Regional Santa Fe, Lavaisse 610, S3004EWB, Santa Fe, Argentina

jroa, pvillarr@frsf.utn.edu.ar

² INGAR-CONICET, Avellaneda 3657, S3002GJC, Santa Fe, Argentina

chiotti@santafe-conicet.gov.ar

Abstract. The verification of collaborative processes is a key issue to consider in cross-organizational modeling methodologies. Some of the existing verification approaches provide only partial support, whereas others impose some restrictions to verify models with advanced control flow, compromise (completely or partially) the enterprise autonomy, or are focused on technology-dependent specifications. In order to deal with these issues we introduce Global Interaction Nets, which are based on Hierarchical and Colored Petri Nets, and the Global Interaction Soundness property, which was adapted from the classical definition of soundness, as the main correctness criterion. The method can be used to formalize and verify models defined with different modeling languages. In addition, we apply the method through a case study modeled with UP-ColBPIP, which is a modeling language for collaborative processes, and formalize its constructs by means of Global Interaction Nets.

Keywords: Verification, Collaborative Business Processes, Petri Nets, B2B Collaborations.

1 Introduction

Nowadays, organizations are being encouraged to integrate their business processes and adapt themselves to dynamic market opportunities by setting up Business-to-Business (B2B) and cross-organizational collaborations. Collaborative Business Processes (CBPs) define the global view (choreography) and the behavior of the interactions of cross-organizational collaborations, as well as the way the organizations B2B systems will interact [1,2]. Errors in the behavior of a CBP may affect several internal processes and cause an error propagation across the boundaries of an organization. In addition, any error in the business solution would be propagated to the technological solution of a B2B collaboration. Therefore, it is essential to provide a verification method which guarantees the control flow of CBPs is well-defined. Thus, business analysts and system designers can get quick feedback regarding the correctness of the behavior of the collaboration, previous to the generation of the organizations private processes and the corresponding technological solution [1].

Currently, several modeling languages can be used to model the control flow of CBPs, such as BPMN [3], UMM [4], UP-ColBPIP [2,1], IOWF [5]. Providing formal semantics of such languages is an important task to eliminate language's ambiguities and enable the definition of formal verification methods. At present, there is no agreement on the proper construct to represent interactions in CBPs: interaction activities [3], business transactions [4], or messages [5,2]. Thus, to enable the verification of different representations of CBPs and make the reuse of verification methods easier, it is important to abstract methods from such differences and make them independent of modeling languages.

In addition, important aspects to consider in verification methods include the fulfillment of requirements for CBPs [1,2], the support to models with advanced control flow such as cancellation regions and advanced synchronization [6,7], and a good performance in process analysis. However, existing verification approaches for CBPs [5,8,9,10] do not tackle all these issues.

Therefore, in this work we propose a verification method for the global view of interactions of CBP models that copes with the above issues. The purpose is to detect errors in technology-independent CBP models. The method is independent of the modeling language whose models are going to be verified. To this aim, we introduce a Global Interaction Net, which is a particular type of Hierarchical and Colored Petri Net, and the Global Interaction soundness property as the main correctness criterion. We apply the method to UP-ColBPIP [1,2] and use CPN Tools [11] to formalize and verify UP-ColBPIP models, which may have advanced control flow, by means of Global Interaction Nets and the Global Interaction soundness property. From now on, we use GI to denote Global Interaction.

This work is structured as follows. Section 2 describes the verification method. Section 3 apply the method to UP-ColBPIP models. Section 4 presents some related work. Finally, section 5 presents the conclusions and future work.

2 Verification Method for CBP models

We define the verification method for CBPs as follows.

Definition 1. *The verification method for the global view of interactions of CBP models is an eight-tuple $V_{CBP} = (\lambda, GI_M^a, \phi, GI_M^c, GI_N, R, P, A)$, where*

1. λ is a finite set of modeling constructs that defines a conceptual modeling language for CBPs,
2. GI_M^a is a finite set of abstract GI Modules where each $am \in GI_M^a$ formalizes a set of constructs $C \subset \lambda$,
3. ϕ is a CBP model defined by an ordered finite set of model elements such that each element $e \in \phi$ is associated with a construct $c \in \lambda$,
4. GI_M^c is a finite set of concrete GI Modules where each concrete GI module $cm \in GI_M^c$ formalizes a set of model elements $E \subset \phi$ and each cm is associated with an abstract GI module $am \in GI_M^a$,

5. GI_N is a GI Net defined by an ordered finite set of concrete GI Modules GI_M^c that formalize the model ϕ ,
6. $R : \phi \rightarrow GI_M^c$ is a mapping function that defines a transformation from a set of model elements $E \subset \phi$ to a concrete GI Module $cm \in GI_M^c$,
7. P is a finite set of correctness properties,
8. $A : GI_M^c \rightarrow \phi$ is a relation function that associates each concrete GI module $cm \in GI_M^c$ with a set of CBP model elements $E \subset \phi$.

The aim of the proposed method is to verify the global view of interactions of a CBP model ϕ , defined with a language λ . A set of constructs of λ can be formalized with an abstract GI Module GI_M^a , whereas a set of elements of the CBP model ϕ can be formalized with a concrete GI Module GI_M^c . The difference between abstract and concrete GI module is analogous to that of a class and its instances in the object oriented paradigm. To apply this method and verify models of a specific CBP modeling language, formal semantics of each construct must be defined with abstract GI modules. The GI Net, which formalizes a CBP model, is composed of a set of concrete GI Modules, and is generated by applying the function R to the CBP model. Retrieving the element $e \in \phi$, which is the source of an error in a CBP model ϕ , is possible by means of function A , which establishes a direct association between each concrete GI module and its corresponding element of the CBP model. The output of this method is the set of model elements of ϕ that do not hold the correctness properties P . If no error is found, an empty set is returned. Following, the elements and main features of this method are described.

2.1 Preliminaries

A *Colored Petri Net (CP-Net)* [11] is a nine-tuple $CPN = (P, T, A, \Sigma, V, C, G, E, I)$, where P is a finite set of *places*, T is a finite set of *transitions*, and A is a finite set of directed *arcs* that connects places to transitions (and vice versa). A CP-Net has a finite set of *color sets* Σ , a finite set of variables V of one type (color) of Σ , and a *color set function* C that assigns a color set to each place. Transitions may have a *guard function* G , and arcs may have an *arc expression function* E . I is an *initialization function* of the CP-Net. $\bullet p$ is the set of input transitions $t \in T$ of the place p , and $p\bullet$ is the set of output transitions of p . $\bullet t$ is the set of input places of the transition t , and $t\bullet$ is the set of output places of t . $|\bullet p|$ and $|p\bullet|$ are the number of input and output transitions of place p respectively. $|\bullet t|$ and $|t\bullet|$ are the number of input and output places of transition t respectively. For any two markings $M1$ and $M2$, $M1 \geq M2$ iff for each $p \in P : M1(p) \geq M2(p)$. $M1 > M2$ iff $M1 \geq M2$ and $M1 \neq M2$. For a place p and marking M , $|M(p)|$ is the number of tokens on p in marking M .

A *CP-Net module* [11] is a tuple $CPN_M = (CPN, T_{sub}, P_{port}, PT)$, where CPN is a CP-Net that has a set of *port places* P_{port} , which have a *port type* PT (IN,OUT,I/O). It may also have a set of *substitution transitions* T_{sub} .

A *Hierarchical and Colored Petri Net (HCP-Net)* [11] is a four-tuple $CPN_H = (S, SM, PS, FS)$, where S is a finite set of *CP-Net modules*. Each substitution

transition have its corresponding CPN-Net module by means of the *submodule function* SM . The input and output places of a CP-Net module are associated with their corresponding places of the substitution transition by means of a *port-socket relation function* PS . FS is a set of *fusion sets* of places. P^s is the set of places of the module $s \in S$. T^s is the set of transitions of the module $s \in S$. $PT(p)$ is the port type of a place p . $E(p, t)$ is the arc expression on the input arc from p to t . $E(t, p)$ is the arc expression on the output arc from t to p . $Var[e]$ is the set of *free variables* in an expression e , and $Type[v]$ is the type (color) of a variable v . $Var(t)$ denotes the variables of a transition $t \in T$. Let K be a set of tokens. A *multiset* over K is a function $t : K \Rightarrow \mathbb{N}$ that maps each element $k \in K$ into the number of appearances $t(k) \in \mathbb{N}$ of k in t .

2.2 A Formal Language for Global Interactions of CBPs

A CBP model is composed of a choreography of interactions between organizations. In this work, we introduce a Global Interaction Net (GI-Net), which is a type of HCP-Net, to formalize CBP models. In addition, we introduce a Global Interaction module (GI module) to represent an interaction or a set of control flow elements of a CBP model. Since a GI-Net defines the control flow and interactions from a global (synchronous) point of view, interacting roles of a CBP are not represented. A GI-Net is a structured tree whose elements are GI modules. GI-Nets and GI modules have two types of places: *interaction place*, which represents the expected state of interactions and its color set is $GINT$, and *control place*, which is used to restrict and control the advanced control flow of interactions and its color set is $CTRL$. Control places could be used, for instance, in exception constructs (see section 3.1) to block the control flow of parallel interactions while the exception is being executed. To this aim, different modules are connected through control places. A GI-Net allows the connection of different modules only through their corresponding input and output interaction places, or by a set of control places that are part of the same fusion set. We define GI-Net and GI module as follows.

Definition 2. A CP-Net module $GI_M = (CPN, T_{sub}, P_{port}, PT)$ is a GI module iff:

1. $P_I \subset P$ is a non-empty set of interaction places such that for each $p \in P_I$ the color set of p is $GINT$,
2. $P_C \subset P$ is a set of control places such that for each $p \in P_C$ the color set of p is $CTRL$,
3. There is only one input place $ip \in P_I$ such that $PT(ip) = IN \wedge \bullet ip = \emptyset$,
4. There is only one output place $op \in P_I$ such that $PT(op) = OUT \wedge op \bullet = \emptyset$,
5. For each element $n \in (P \cup T) \wedge n \neq op$, there is a direct path from ip to n .

Definition 3. Given a CP-Net module $r \in S$, a HCP-Net $GI_N = (S, SM, PS, FS)$ is a GI-Net iff:

1. GI_N is an ordered finite set of GI modules which is structured as a tree, such that the root of the tree is the module $r \in S$,

2. For each transition $t \in T^r$, t is associated with a GI module by means of the submodule function SM ,
3. For each GI module $s \in S$ such that $s \neq r$, there is a direct path from r to s ,
4. $P_I \subset P^r$ is a non-empty set of interaction places such that for each $p \in P_I$ the color set of p is $GINT$,
5. $P_C \subset P^r$ is a set of control places such that for each $p \in P_C$ the color set of p is $CTRL$,
6. There is only one input place $ip \in P_I$ such that $\bullet ip = \emptyset$,
7. There is only one output place $op \in P_I$ such that $op \bullet = \emptyset$, and op is a member of the fusion set $End \subset FS$,
8. For each element $n \in (P \cup T)$, n is on a directed path from ip to op ,
9. For each place $p \in P^r$ and $t \in T^r$ such that $p \neq ip \wedge p \neq op$, $|p \bullet| = |\bullet p| = |t \bullet| = |\bullet t| = 1$,
10. For each $s1, s2 \in S$ and each $p1 \in P^{s1}, p2 \in P^{s2}$ such that $s1 \neq s2$, if $p1, p2$ are members of the fusion set $f \subset FS$, then $Type[p1] = Type[p2] = CTRL$.

2.3 Correctness Properties for Global Interactions

Formalizing advanced control flow constructs with Petri nets may imply tokens in different places in the final marking of a net. The classical soundness definition [7] is too restrictive and does not support this. Other variants of soundness relax this restriction [7], but they do not guarantee that places (different from the final place) have a proper state in the final marking. In a GI-Net, we want to be sure that, in the final marking, all the *interaction places*, except the final place, are in the empty state, and *control places* are in a predefined state (may be different from empty). To deal with these issues, we propose GI soundness as follows.

Definition 4. (*GI soundness*) Let $GI_N = (S, SM, PS, FS)$ be a GI-Net. Let M_E, M_0 be the empty and initial markings respectively of GI_N , such that $\forall_{p \in P_c} |M_E(p)| \geq 0 \wedge \forall_{p \in P_I} |M_E(p)| = 0 \wedge M_0 = M_E + M(ip)$. Furthermore, let M_F be the final marking of GI_N , such that $M_F = M_E + M(op)$. Then, GI_N is GI sound iff:

1. Option to complete: $\forall_M (M_0 \xrightarrow{*} M) \Rightarrow (M \rightarrow M_F)$.
2. Proper completion: $\forall_M (M_0 \xrightarrow{*} M \wedge M \geq M_F) \Rightarrow (M = M_F)$.
3. No dead transitions: $\forall_{t \in T} \exists_{M, M'} M_0 \xrightarrow{*} M \xrightarrow{t} M'$.

In order to determine if a given GI-Net GI_N is GI sound, we define the extended net \overline{GI}_N by adding a transition t^* which connects op and ip to GI_N .

Theorem 1. If (\overline{GI}_N, ip) is live and bounded, then GI_N is GI sound.

Proof. (\overline{GI}_N, ip) is live. Since op is the input place of t^* , for any marking M reachable from marking M_0 it is possible to reach a marking with at least one token in place op . Consider an arbitrary reachable state $M' + M(op)$, i.e. a marking with one token in place op . In this marking t^* is enabled. If t^* fires, the marking $M' + M(ip)$ is reached. Since (\overline{GI}_N, ip) is also bounded, M' should be equal to M_0 . Hence requirements 1 and 2 hold and proper termination is guaranteed. Requirement 3 holds since (\overline{GI}_N, ip) is live. Hence, GI_N is GI sound.

In case a GI-Net is not GI sound, we want to determine the reasons. For a sound GI-Net GI_N the following rules hold: 1) There must be exactly one dead marking [11] M_D and one home marking [11] M_H , such that both markings are the same, and they are also the final marking M_F of GI_N . i.e., $M_D = M_H = M_F$, 2) For each transition $t \in T$, t is not dead. If the first rule does not hold, then there is at least one deadlock different from the final marking M_F in the GI-Net. The location of the deadlock is determined by checking each marking M' which is part of the set of deadlock markings M_{dl} , such that $M' \neq M_F$. For each place $p \in P^s$ and module $s \in S$, if $M'(p) \neq M_F(p)$ there is a deadlock originated in module s , and at least one of the transitions $p \bullet$ is the source of the deadlock. If the second rule does not hold, it means that there is at least one interaction that will not be carried out. The GI module causing the unexpected behavior is determined by checking the set of dead transitions T_d of the GI-Net.

GI soundness is determined with CPN Tools by inspecting the properties of the report that it generates as follows. Given a non-negative integer $n \in \mathbb{N}$, if each place p in the report holds that $|M(p)| \leq n$ then the GI-Net is bounded. Liveness is determined by checking if t^* is a member of the set of “live transition instances”. CPN Tools also reports the sets of “dead markings” and “dead transitions”, so if a GI-Net is not GI sound, the reasons can be determined by inspecting these sets.

3 Applying the Verification Method to UP-ColBPIP

The UP-ColBPIP language [1,2] extends the UML2 semantics and encourages the modeling of technology-independent CBPs in a top-down approach. UP-ColBPIP supports the modeling of interaction protocols to represent the global view of interactions through a choreography of business messages between organizations who play different roles. In interaction protocols, internal activities of organizations cannot be defined; hence, the organizations autonomy is preserved.

In UP-ColBPIP, *Organizations and the Role* they fulfill are represented through lifelines. The basic building blocks of an interaction protocol are the business messages. A *Business Message* defines a one-way asynchronous interaction between two roles: a sender and a receiver. Protocols have an implicit termination. A *Termination* represents an explicit end event of a protocol, which can be labeled *Success* or *Failure*. A *Control Flow Segment (CFS)* represents complex message sequences. It contains a *Control Flow Operator* and one or more interaction paths. An *Interaction Path* contains an ordered sequence of protocol elements: messages, termination events, protocol references, and nested CFSs. The semantics of a CFS depends on the operator used. The control flow operators of UP-ColBPIP are: And, Xor, Or, If, Loop, Exception, Cancel, and Multiple Instances. Further details of this language can be found in [1,2].

3.1 Formal Semantics for the UP-ColBPIP Language

To represent a UP-ColBPIP model with a GI-Net GI_N , for each construct of UP-ColBPIP an abstract GI module $am \in GI_M^a$ must be defined. The input

and output places of each GI module are ip and op respectively, and their port types are $PT(ip) = IN$ and $PT(op) = OUT$. If the color set of a place $p \in P$ is not defined, then $Type[p] = GINT$ must be assumed. Following the abstract GI modules for UP-ColBPIP are described. We omit modules for simple control flow constructs (Xor, And, etc.) since they can be easily inferred from [12].

Interaction Protocol, Business Message, Interaction Path, and Termination. Let $r = \text{InteractionProtocol}$ be a module of S such that r is the root of the tree that defines the GI-Net GI_N . Let $s = \text{BusinessMessage}$ be a GI module of S where $bmt \in T^s$ is a transition such that $bmt \in ip \bullet \wedge bmt \in \bullet op \wedge |ip \bullet| = |bmt \bullet| = |bmt \bullet| = |bmt \bullet| = |bmt \bullet| = 1$. Let $ip = \text{InteractionPath}$ be a GI module of S where for each place $p \in P^{ip}$ and $t \in T^{ip}$, p, t are on directed path from ip to op , and if $p \neq ip \wedge p \neq op$ then $|p \bullet| = |\bullet p| = |t \bullet| = |\bullet t| = 1$. Let $t = \text{Termination}$ be a GI module of S where $tt \in T^s$ is a transition such that $tt \in ip \bullet \wedge |ip \bullet| = |\bullet tt| = 1 \wedge |tt \bullet| = |\bullet op| = 0$.

Or. The complete GI module for the Or construct is composed of a split and a synchronization part. Since the Or construct can be synchronized in different ways, the common part of the Or split is first formally described.

Let $s \in S$ be a GI module in which the Or split is defined. Given a non-negative integer $m \in \mathbb{N}$ that represents the max number of parallel paths, the transition $it \in T^s$ is connected to the place $ip \in \bullet it$ such that $|\bullet it| = 1$ and $|it \bullet| = m$. Let V' be a subset of the set of variables V such that $V' \subseteq Var(it)$. Let $EXPR_{V'}$ be the set of expressions $se \in EXPR$ such that $Var[se] \subseteq V'$. Let A' be a subset of the set of arcs A such that $A' \subseteq it \times it \bullet$. For each place $ipp \in it \bullet$ there is an arc $a \in A'$, a substitution transition $t \in T_{sub}^s$ that represents an interaction path, a place $opp \in P^s$, an arc expression $e = E(a) = E(it, ipp) \in EXPR_{V'}$, and a variable $v \in Var[e]$ such that $Type[v] = \text{BOOL} \wedge ipp \in \bullet t \wedge opp \in t \bullet \wedge |t \bullet| = |t \bullet| = 1$, and the multiset of tokens K added to the place ipp is given by the evaluation of the arc expression e , where

$$e = E(a) = E(it, ipp) = \begin{cases} k \in K & \text{if } v = \text{true} \\ \emptyset_{MS} & \text{if } v = \text{false} \end{cases}$$

Following, the Or 's synchronization types are described. Both the split and synchronization descriptions must be read together as a whole

Or with Synchronizing Merge Let $s = \text{OrSyncMerge}$ be a GI module of S . For each place $opp \in \bullet ot$ there is an arc $a \in A'$, an arc expression $e1 = E(a) = E(it, opp) \in EXPR_{V'}$ such that the multiset of tokens $k1$ added to the place opp is given by the evaluation of the arc expression $e1$, where

$$e1 = E(a) = E(it, opp) = \begin{cases} k \in k1 & \text{if } v = \text{false} \\ \emptyset_{MS} & \text{if } v = \text{true} \end{cases}$$

Or with N-out-of-M Let $s = \text{Or-N-out-of-M}$ be a GI module of S . Let the non-negative integer $n \in \mathbb{N}$ be the number of interactions to be synchronized. Let

$p1 \in P^s$ be a place such that $|\bullet p1| = M$. For each place $opp \in P^s$ there is a transition $ot \in T^s$, such that $ot \in \bullet p1$ and $ot \in opp\bullet$. Transition $ct \in T^s$ is connected to places $p3, rp \in ct\bullet$ and places $p1, p2 \in \bullet ct$. Transition $rt \in T^s$ is connected to places $p2, p4 \in rt\bullet$ and places $p1, rp \in \bullet rt$. Transition $bt \in T^s$ is connected to $p3, p4 \in \bullet bt$ and $op \in bt\bullet$. Let A' be a subset of the set of arcs A such that $A' \subseteq (p1 \times \bullet rt)$. The arc $a \in A'$ has an arc expression $e = E(a) = E(p1, rt) \in \text{EXPR}_{V'}$, such that the multiset of tokens K removed from the place $p1$ is given by the evaluation of the arc expression $e = E(a) = E(p1, rt) = t(k)'k$ such that $k \in K$ and $t(k) = m - n$.

Multiple Instances With a Priori Design-Time Knowledge. Let $s = \text{MI-DT}$ be a GI module of S . Let the non-negative integer $n \in \mathbb{N}$ be the number of multiple instances known at design time. Let transitions $it, ot \in T^s$ and the place $p1 \in P^s$ be given such that $it \in ip\bullet \wedge it \in \bullet p1$ and $ot \in p2\bullet \wedge ot \in \bullet ot$. Let $t \in T_{sub}^s$ be a substitution transition representing an interaction path such that $t \in p1\bullet \wedge t \in \bullet p2$. Let A' be a subset of the set of arcs A such that $A' \subseteq (it \times \bullet p1)$. The arc $a \in A'$ has an arc expression $e = E(a) = E(it, p1) \in \text{EXPR}_{V'}$, such that the multiset of tokens K added to the place $p1$ is given by the evaluation of the arc expression e , where $e = E(a) = E(p1, rt) = t(k)'k$ such that $k \in K$ and $t(k) = n$. The arc $a1 \in A'$ has an arc expression $e1 = E(a1) = E(p2, ot) \in \text{EXPR}_{V'}$, such that the multiset of tokens $K1$ removed from place $p2$ is given by the evaluation of the arc expression $e1$, where $e1 = E(a1) = E(p2, ot) = t(k)'k$ such that $k \in K1$ and $t(k) = n$.

Exception. Let $s = \text{Exception}$ be a GI module of S . Let transitions $it, ot1, ot2 \in T^s$ be given such that $it \in ip\bullet \wedge ot1 \in \bullet op \wedge ot2 \in \bullet op$. Let the substitution transition $exct \in T_{sub}^s$ be the expected control flow of the module and let the set of substitution transitions $T_{trigt}, T_{handt} \subseteq T_{sub}^s$ be the exception triggers and handlers respectively. $exct$ is connected to places $eip, eep \in P^s$ such that $it \in \bullet eip \wedge exct \in it\bullet$ and $exct \in \bullet eep \wedge ot1 \in eep\bullet$. For each transition $handt \in T_{handt}$ there is a transition $trigt \in T_{trigt}$ such that there is a place $hip \in \bullet handt \wedge hip \in trigt\bullet$, and a place $hop \in handt\bullet \wedge hop \in \bullet ot2$. Let $Reset, Ctrl \subset FS$ be sets members of the set of fusion places. Let $rp \in \bullet ot2$ be a place member of the fusion set $Reset$ such that $Type[rp] = CTRL$. Let $ctrp \in ot2\bullet$ be a place member of the fusion set $Ctrl$ such that $Type[ctrp] = CTRL$. Let the non-negative integer $n \in \mathbb{N}$ be the max number of parallel paths within the exception scope. The output transition $ot2$ has a guard expression $G(ot2)$, such that the multiset of tokens K removed from the place rp is given by the evaluation of the arc expression e , where $e = E(a) = E(p1, rt) = n'k$ such that $k \in K1$ and $t(k) = n$.

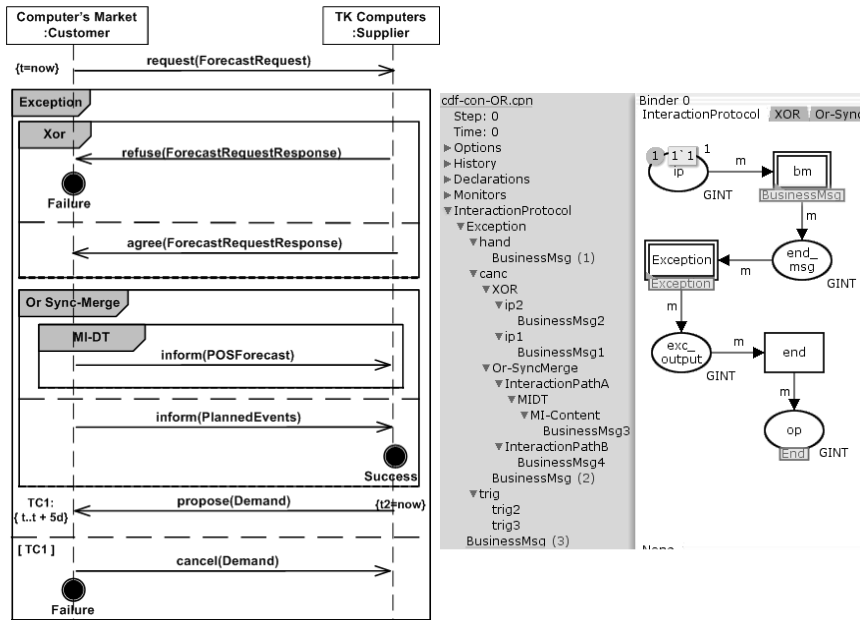
Let $se = Excc$ be a GI module of S associated with transition $exct \in T^s$. Let $q \subset S$ be the set of submodules of s . For each transition $t \in T^q$ there is a place $c \in P^q$ member of the fusion set $Ctrl$ such that $Type[c] = CTRL$ and $c \in \bullet t \wedge t \in c\bullet$. Let the set of fusion sets $epfs \subset FS$ be given. For each place $p \in P^q$ that is a trigger place, i.e. a point where an exception can occur, p is member of the fusion set $epfs$ such that $Type[c] = GINT \wedge p \neq ip \wedge p \neq op$.

Let $st = Trig$ be a CP-Net module of S associated with transition $trigt \in T^s$. Let the sets of fusion sets $rpfs, ctrlfs \subset FS$ be given. Let $rp, ctrlp \in P^{st}$ be places members of the fusion sets *Reset* and *Ctrl* respectively, such that $Type[rp] = Type[ctrlp] = CTRL$. For each trigger place in se , there are a place $ep \in P^{st}$ and transitions $ct, et \in T^{st}$ such that ep is member of the fusion set $epfs$ and $Type[ep] = GINT \wedge ep \neq op \wedge ep \in \bullet et \wedge ep \in \bullet ct \wedge et \in ctrlp \bullet \wedge et \in \bullet op \wedge et, ct \in \bullet rp \wedge ct \in rp \bullet$. Transition et represents the raising of an exception. The place $ctrlp$ guarantees that only one exception can be raised at any given time within the exception scope. The place rp enables the cleaning of tokens after the exception was raised. Finally, let $sh = Hand$ be a GI module of S associated with transition $handt \in T^s$ where for each place $p \in P^{sh}$ and $t \in T^{sh}$, p, t are on directed path from ip to op , and if $p \neq ip \wedge p \neq op$ then $|p \bullet| = | \bullet p| = |t \bullet| = | \bullet t| = 1$.

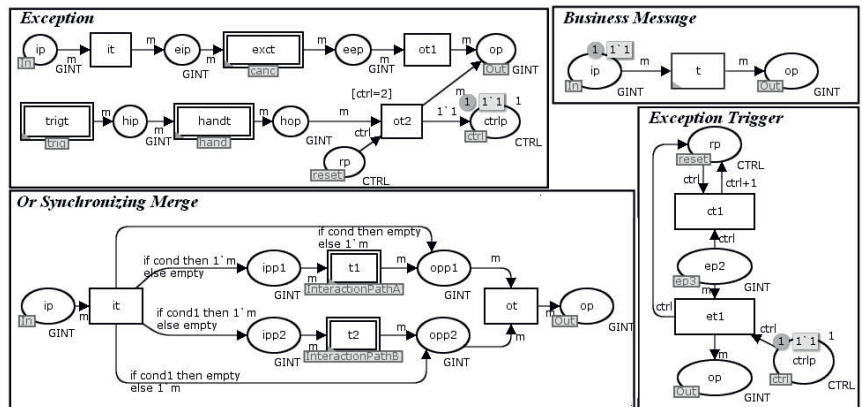
Cancel. The difference between this module and the Exception module is that the arc going from transition $ot2$ to place op is not part of the Cancel GI module. Then, when an exception occurs, the Cancel finishes the process execution.

3.2 Applying the Verification Approach to an Example

We applied the verification method to a collaborative model for the supply chain management of desktop computers and notebooks, taken from a real world case study. In this scenario, there are two organizations collaborating to agree on a demand forecast. Fig. 1a shows the interaction protocol for this process. The *Success* element in the Or construct was intentionally added to show the results of the verification approach. Fig. 1b shows the GI-Net that represents the interaction protocol of the scenario in CPN Tools. The left part of Fig. 1b shows the tree structure of the GI-Net. Fig. 1c shows three concrete GI modules of the GI-Net: Or with Synchronizing Merge, Business Message, and Exception. The last one is composed of three main modules: the expected (normal) interaction flow ($exct$), the exception trigger ($trigt$) (which is also shown in Fig. 1c), and the exception handler ($handt$). The report generated by CPN Tools returned three dead markings. One of them represents the proper termination of the CBP, whereas the other two represent a deadlock caused by the definition of the termination “Success” in one of the interaction paths of the Or construct. The semantics defined for the Or construct establishes that once a path has been enabled, it must be synchronized. In the example, the synchronization will never occur since the termination construct “Success” finishes the CBP before the synchronization can be realized. Furthermore, since both paths in the Or construct are concurrent, interactions may occur even when the CBP has already finished through the termination “Success”. We get similar results if we define a Cancel construct within a path of the Or construct. In larger CBPs, this could be a recurrent problem without the aid of a verification method.



(a) Interaction Protocol of the Scenario (b) GI-Net of the Interaction Protocol



(c) Concrete GI Modules Defined in the GI-Net

Fig. 1. Collaborative Demand Forecast Process

4 Related Work

In [5,8], the verification of IOWFs was proposed, but these approaches compromise (completely or partially) the enterprises' autonomy. The method we are proposing verifies the choreography of interactions before public and private views are generated, which is appropriate in a top-down approach for modeling CBPs [13] and do not compromise the enterprises' autonomy. A verification approach based on HCP-Nets was presented in [9]. However, it does not guarantee that CBP models are deadlock-free neither it provides any support for advanced control flow. Other approaches are based on technology-specific languages [10], but they focus on verifying process specifications instead of process models when the business solution is defined. The use of structured processes was proposed to improve the performance of verification methods [14]. However, a structured formalization of advanced control flow constructs is not easy to accomplish. We showed that the structure of a GI-Net enables the reuse of formal definitions and delimits the scope of a problem in a module.

5 Conclusions and Future Work

In this work, we proposed a method for the verification of technology-independent CBP models. It enables the detection of errors in the global view of interactions of CBPs keeping the organizations' autonomy safe, since no private aspects of the participating organizations are considered. The method is independent of the semantics of any specific CBP modeling language, which makes it flexible and adaptable to be used with different CBP modeling languages. To this aim, it provides mechanisms to formalize the constructs of a language and verify its models. In particular, it allows the representation of the formal semantics of the cross-organizational message exchange that any CBP language should provide.

To support the method, we defined Global Interaction Nets (GI-Nets), which are based on Hierarchical and Colored Petri Nets, and are used to formalize CBP models. A GI-Net is composed of concrete GI modules that formally represent the elements of a CBP model, i.e. interaction or control flow elements. Concrete GI modules are derived and defined according to abstract GI modules that formalize the constructs of a CBP modeling language. Abstract GI-modules of a language can be used to develop a transformation process that generates a GI-Net from a model defined with such a language.

Since a GI-Net is a structured tree of GI modules, its corresponding CBP model must be structured as well. In block-based languages, which only generate structured models such as UP-ColBPIP, this is direct. However, to apply this method to a non-structured CBP model, e.g. BPMN-based choreographies, it has to be first transformed into a structured one before deriving its corresponding GI-Net. The structure of a GI-Net makes the reuse of formal definitions easier

and delimits the scope of a problem to a module. GI modules for simple control flow constructs (and, xor, etc.) can be structurally defined according to [14] to reduce the analysis complexity. GI-Nets for models with advanced control flow constructs can be defined as structured as possible by decomposing the solution into modules.

In addition, we proposed GI soundness as the main correctness criterion. With this property, the final marking is not restricted to having a token only in the final place, as it happens in the classical soundness definition. Furthermore, a random final marking, which could stem from miss-behaved situations, is avoided since it has a predefined set of control tokens in the control places and no tokens in the interaction places, except for the final (interaction) place.

The proposed verification method was applied to UP-ColBPIP. Constructs for advanced synchronizations, cancellation and exception management were formalized with abstract GI modules, and a verification example of a UP-ColBPIP model was given. We showed that GI-Nets can be used to formalize these constructs in a modular way. In addition, we described how to use CPN Tools to define GI-Nets that formalize CBP models, verify them according to the GI soundness property, and detect the modules that cause errors in a GI-Net.

Ongoing work is concerned with the empirical validation of this method and its implementation into an Eclipse-based tool for modeling and verifying CBPs. We are also going to apply this method to BPMN-based choreography models. Technical and performance aspects are also part of future work.

References

1. Villarreal, P., Salomone, H., Chiotti, O.: Modeling and specifications of collaborative business processes using a MDA approach and a UML profile. In: Enterprise Modeling and Computing with UML, pp. 13–45. Idea Group Inc. (2007)
2. Villarreal, P.D., Lazarte, I., Roa, J., Chiotti, O.: A Modeling Approach for Collaborative Business Processes Based on the UP-ColBPIP Language. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBP, vol. 43, pp. 318–329. Springer, Heidelberg (2010)
3. OMG, BPMN 2.0: <http://www.omg.org/spec/BPMN/2.0/>
4. Huemer, C., Liegl, P., Motal, T., Schuster, R., Zapletal, M.: The development process of the UN/CEFACT modeling methodology. In: ICEC, pp. 1–10. ACM (2008)
5. van de Aalst, W.M.P.: Modeling and analyzing interorganizational workflows. In: Proc. of the ACSD, pp. 262–272. IEEE Comp. Soc., Washington, DC (1998)
6. Aalst, W.M.: Discovery, verification and conformance of workflows with cancellation. In: Ehrig, H., Heckel, R., Rozenberg, G., Taentzer, G. (eds.) ICGT 2008. LNCS, vol. 5214, pp. 18–37. Springer, Heidelberg (2008)
7. van der Aalst, W., van Hee, K., ter Hofstede, A., Sidorova, N., Verbeek, H., Voorhoeve, M., Wynn, M.: Soundness of workflow nets: classification, decidability, and analysis. *Formal Aspects of Computing*, 1–31 (2010)
8. Norta, A., Eshuis, R.: Specification and verification of harmonized business-process collaborations. *Information Systems Frontiers* 12, 457–479 (2010)

9. Stuit, M., Szirbik, N.: Towards Agent-Based Modeling and Verification of Collaborative Business Processes: an Approach Centered on Interactions and Behaviors. *Int. Journal of Cooperative Information Systems (IJCIS)* 18, 423–479 (2009)
10. Breugel, F., Koshkina, M.: Models and Verification of BPEL, <http://www.cse.yorku.ca/~franck/research/drafts/tutorial.pdf>
11. Jensen, K., Kristensen, L.M.: *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*, 1st edn. Springer Publishing Company, Incorporated Heidelberg (2009)
12. Workflow Patterns: <http://www.workflowpatterns.com>
13. Lazarte, I., Tello-Leal, E., Roa, J., Chiotti, O., Villarreal, P.: Model-Driven Development Methodology for B2B Collaborations. In: *EDOCW*, pp. 69–78. IEEE (2010)
14. Vanhatalo, J., Völzer, H., Leymann, F.: Faster and More Focused Control-Flow Analysis for Business Process Models Through SESE Decomposition. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) *ICSOC 2007*. LNCS, vol. 4749, pp. 43–55. Springer, Heidelberg (2007)