



**HAL**  
open science

## Towards Timed Business Process View Generation

Saoussen Cheikhrouhou, Slim Kallel, Nawal Guermouche, Mohamed Jmaiel

► **To cite this version:**

Saoussen Cheikhrouhou, Slim Kallel, Nawal Guermouche, Mohamed Jmaiel. Towards Timed Business Process View Generation. 10th IEEE International Conference on e-Business Engineering (ICEBE'13), Sep 2013, United Kingdom. hal-00805532v1

**HAL Id: hal-00805532**

**<https://hal.science/hal-00805532v1>**

Submitted on 28 Mar 2013 (v1), last revised 20 Dec 2013 (v2)

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

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

# Modeling, Safely Advertising and Verifying Time-aware Business Processes : Towards a holistic Approach

Saoussen Cheikhrouhou<sup>1</sup>, Slim Kallel<sup>1</sup>, Nawal Guermouche<sup>2,3</sup>, and Mohamed Jmaiel<sup>1</sup>

<sup>1</sup> ReDCAD Laboratory, National Engineering School of Sfax, University of Sfax B.P. 1173, 3038 Sfax, Tunisia

`saoussen.cheikhrouhou@redcad.org`, `Slim.kallel@fsegs.rnu.tn`,  
`mohamed.jmaiel@enis.rnu.tn`

<sup>2</sup> CNRS-LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France

<sup>3</sup> Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France

`Nawal.Guermouche@laas.fr`

**Abstract.** Nowadays, the business process model and notation BPMN standard is gaining widespread use in the business world. In this context, several underlying issues must be considered. In this paper, we are particularly interested in the problem of getting control over the business process outsourcing through *views* generation. Indeed, the concept of a *view* is essential since it allows to organizations to choose the parts that can be exposed and to keep secret the critical parts of their business processes. In this context, we are specially interested in considering time properties when building public views from private processes. First, we propose a BPMN extension for capturing temporal requirements during the business process modelling (BPM) phase. Second, based on these extensions, our work attempts to cope with industrial privacy preservation in inter-organizational business processes (IOBPs) by proposing a Time-aware Automatic Process View Generation APVG approach. The temporal constraints advertisement will be ensured by a set of algorithms focusing particularly on the propagation of the temporal constraints from private to public processes. Finally, we present a model checking based verification process to foresee the possible temporal violations enabling thus to react to them predictively.

**Keywords:** Temporal constraints and dependencies : Business Process Modeling (BPM), Inter-Organizational Business Process (IOBP), Process Views, BPMN, Model checking, Verification

## 1 Introduction

With the advent of open communication infrastructures like Internet, the business-to-business (B2B) e-commerce market is expected to expand rapidly. Within B2B transactions, multiple entities, such as manufacturers, parts suppliers, shippers, and specialized subcontractors collaborate together and form Inter-Organizational

Business Processes (IOBP) involving different processes which can depend on different parameters such as *time*. In fact, business process field is influenced by a wide range of temporal requirements which rise from legal, regulatory, and managerial rules. Failing to consider temporal information in process models turns out in higher process execution costs, either by loss of productivity, lack of coordination, or missed deadlines committed with the involved customers [1]. Consequently, organizations aiming at providing cost-competitive products, are striving to include and to consider the temporal dimension in their processes. This, in turn, has led to an increasing demand for innovative mechanisms and technologies that support the time modelling and management in the process lifecycle.

One of the important and challenging issues in the IOBPs domain is the preservation of the internal process logic and the business secrecy of the involved partners. In this context, the concept of *process views* is widely used to enable organizations to expose only some of their activities while keeping secret the critical parts of their private processes. By revealing all private details and business secrets, providers runs the risk of losing their competitive edge. By disclosing their know-how, other partners might turn from collaborators into competitors. Indeed, it is essential to cope with industrial privacy preservation in inter-organizational business processes because there are serious consequences for organizations entirely exposing their business processes.

In this paper, we are interested in the problem of modelling and managing time in business processes and more particularly we focus on the problem of managing *public timed processes views* construct from private timed processes. To do so, we first propose a BPMN extension to capture temporal requirements during the business process modelling (BPM) phase. Based on the proposed model, our aim is to generate timed views and to manage time propagation when generating public views from private processes. Finally, we propose a model checking based approach to foresee the possible temporal violations enabling thus to react to them predictively.

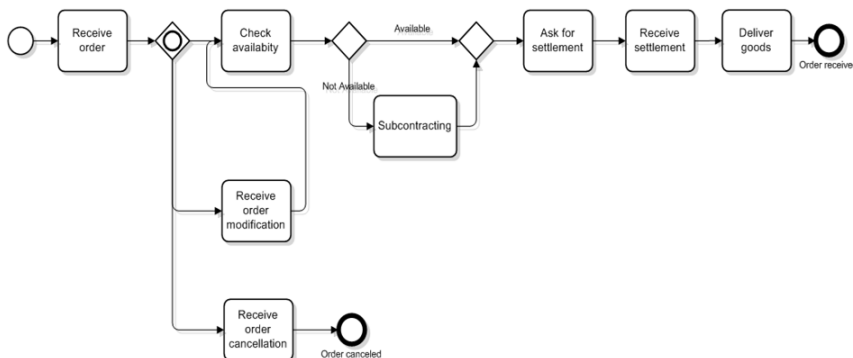
The contributions of this paper can be summarized as follows:

- We provide a BPMN temporal extension to enable the depiction of a large set of temporal constraints on the process model.
- We assist business stakeholders and system implementers with a *Time-aware Automatic Process View Generation* (TAPVG) approach. A set of algorithms enabling particularly the propagation of the temporal constraints from private to public processes views are subsequently presented.
- In the premise of ensuring the correctness of already defined process models, namely the private and public processes, we propose a verification approach based on the model checking technique.

This paper is organized as follows. A motivating example is introduced in Section 2. Section 3 presents a brief description of the proposed BPMN temporal extension. The TAPVG approach is outlined in Section 5. A review of related literature is given in Section 6. Finally, Section 7 concludes.

## 2 Motivating example

To show the underlying issues of the problem we handle, let us consider the BPMN diagram of the purchase order process in a manufacturing organization depicted in Fig. 1.



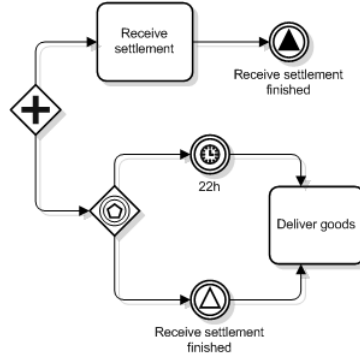
**Fig. 1.** A BPMN process diagram of the purchase order handling

The process is triggered when a customer submits a purchase order (Receive order). Then, it can check whether the ordered articles are available or not (Check availability), modify (Receive order modification) and even cancel the requested order (Receive order cancellation). In case of order modification, the availability of the order is checked again. If the ordered articles are available in stock, the customer is asked for financial settlement and the goods are subsequently delivered, otherwise the organization needs to launch a subcontracting activity. In case of order modification, the availability of the order is checked again. By the reception of an order cancellation, the process meets its end.

Within business processes, the temporal perspective is crucial since temporal constraints dependencies must be respected. For instance, we can mention the following constraints:

- **constraint1:** During processing time it may be possible for the customer to cancel an order, but once a subcontracting activity is triggered, it cannot be cancelled unless organization will make a considerable loss,
- **constraint2:** The activity *Deliver goods* of the purchase order handling process have to start no later than 22h once the process starts,
- **constraint3:** The manufacturing organization can not receive orders in the evening (from 22h- 8h),
- **constraint 4:** If an order cancellation exists, it should be finished before the subcontracting activity begins, and
- **constraint 5:** Order modification is only allowed maximum 1 day after the received order.

BPMN allows to specify some kind of temporal constraints. For example, constraints 2 which states that *the activity Deliver goods of the purchase order handling process have to start no later than 22h once the process starts* can be specified as shown in Fig. 2.



**Fig. 2.** A BPMN diagram to depict the **constraint2**

In spite BPMN offers some capabilities to specify temporal constraints, it remains very poor and cannot allow to capture a wide range of temporal constraints. For example, it is not possible to specify the temporal perspective presented in constraint1 using the BPMN notation, since the latter standard does not allow for depicting dual temporal dependencies between two activities [1].

It is appropriate to emphasize the complexity of the BPMN notation to depict the constraint2 as well. It is clear that tackling such temporal constraints with BPMN can be overwhelming without a simplified extension.

Failing to include complex temporal data in process models turns out in higher process execution costs, either by loss of productivity, lack of coordination, or missed deadlines committed with the involved customers. In the following, we describe the proposed BPMN extension for capturing temporal requirements during business process modelling.

### 3 Temporal Constraints Specification

This section is devoted to give an overview of the basic process modelling concepts of the proposed BPMN extension to well support the temporal perspective in process models. We distinguish three major categories of temporal constraints:

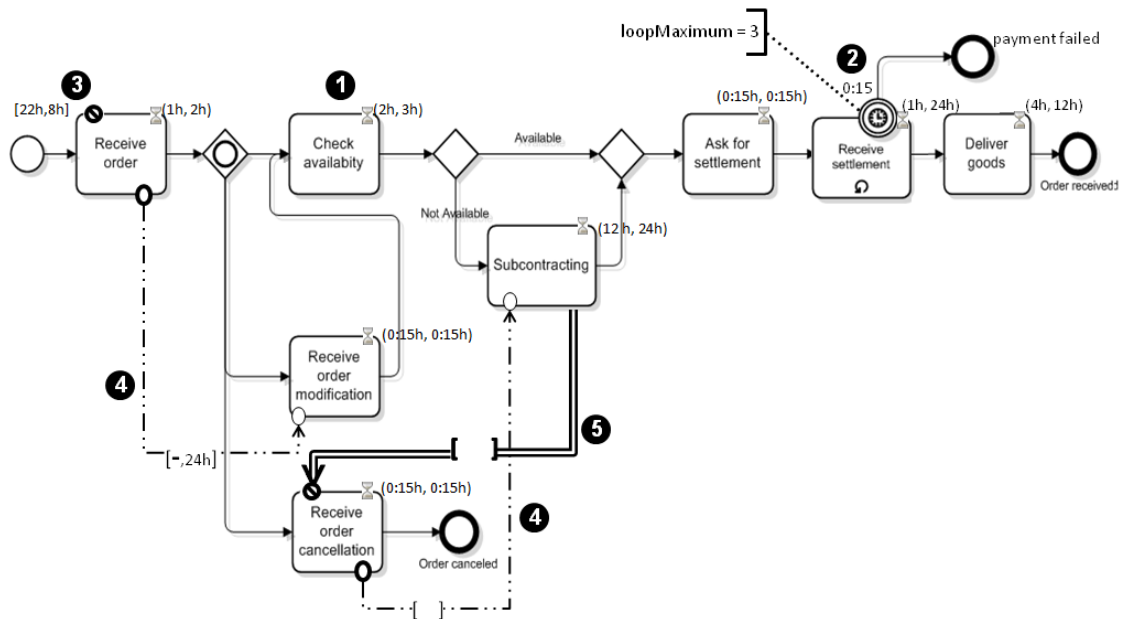
1. Intra-activity temporal constraints
2. Inter-activity temporal constraints
3. Inter-process temporal constraints

Hereafter, we describe each category.

### 3.1 Intra-activity temporal constraints

This category gathers temporal constraints (TC) associated to one activity within a business process such as : (1) *duration*, (2) *TC over cardinality*, (3) *start/end TC*, and (4) *the intra activity absence constraint*.

- **Duration:** BPMN lacks for means to specify the turnaround time of business activities such as the minimum and maximum execution times. Given this limitation, it is mandatory to extend BPMN by `Min_Time` and `Max_Time` attributes for the process model tasks [3]. An activity decorator with the minimum and maximum duration values is proposed and respectively depicted by the label (1) in Fig.3.



**Fig. 3.** The purchase order handling process enriched with proposed temporal constraints

- **TC over cardinality:** This temporal constraint denotes that an activity can be executed successively  $n$  times within a time period. The BPMN standard already provides constructs to specify such constraints (i.e. A loop activity with an Intermediate Event attached to its boundary can be used for that purpose). We use the BPMN notation depicted by the label (2) in Fig.3 to specify that during 15 minutes, a customer can only do 3 failed payment trials.

- **Start/End TC** : Based on the existing temporal BPMN extensions presented in [4, 1], we provide a BPMN extension to specify a wide range of complex temporal requirements. The succeeding listing summarizes the specified temporal constraints to control the start and finish times of the process activities.
  - Must Start On (MSO), Must Finish On (MFO)
  - As Soon As Possible (ASAP), As Late As Possible (ALAP)
  - Start No Earlier Than (SNET), Finish No Earlier Than (FNET), and
  - Start No Later Than (SNLT), Finish No Later Than (FNLT).
  
- **The intra-activity absence constraint** : In a business context, specifying the limited availability of some business activities is of paramount importance. The BPMN specification lacks for means to specify delays within which activities must not occur. To address this requirement, we propose the use of an absence decorator attached to the boundary of an activity as an extension to BPMN. The temporal constraint, denoted by the label (3) in Fig.3 depicts a constraint which states that the manufacturing organization can not receive orders in the evening (from 22h- 8h) due to the availability of some workers for example. We propose hence a solution to the depiction of constraint 3 of Section 2.

### 3.2 Inter-activity temporal constraints

This second category is related to temporal constraints crossing the boundary of an activity in the Process Model such as : (1) *temporal dependency*, and (2) *inter-activity absence constraint*.

- **Temporal Dependency**: Since the BPMN standard does not offer an explicit way to depict dependencies between two activities, temporal dependencies have been proposed to tackle this issue. A temporal dependency is a relationship between two processes or activities, say  $A_1$  and  $A_2$ , in order to coordinate their starting and finishing times. The temporal dependencies suggested in this paper enhance the expressiveness of previous proposals in representing such dependencies, such as Time-BPMN [4] which assigns lead and lag times to a temporal dependency relation. In our work, we propose a fine grained temporal dependencies constraints and we propose the following four temporal dependencies:
  - Start-to-Finish (SF):  $A_2$  can not finish until  $A_1$  has started within a given time interval
  - Start-to-Start (SS) :  $A_2$  can not begin before  $A_1$  starts within a time interval
  - Finish-to-Start (FS) :  $A_2$  can not begin before  $A_1$  ends within a time interval
  - Finish-to-Finish (FF) :  $A_2$  can not finish until  $A_1$  has finished within a time interval

Let us consider the two constraints (4) and (5) presented in section 2 which state respectively:

- If an order cancellation exists, it should be finished before the subcontracting activity begins
- Order modification is only allowed maximum 1 day after the received order

Using the temporal dependencies presented above, these two constraints can be specified as shown in label (4) in Fig. 3.

- **The inter-activity absence constraint:** This kind of constraints enables specifying delays within which activities must not occur. For instance, to specify constraint 1 of Section 2 which denotes that *the order can not be cancelled after the beginning of a subcontracting activity* is not possible in BPMN. To tackle such feature, we propose the BPMN extension denoted by the label (5) in Fig. 3.

### 3.3 Inter-process temporal constraints

This category of constraints is related to temporal constraints crossing the boundary of one process such as : (1) *exchanged temporal data*, and (2) *deadline of message exchange*.

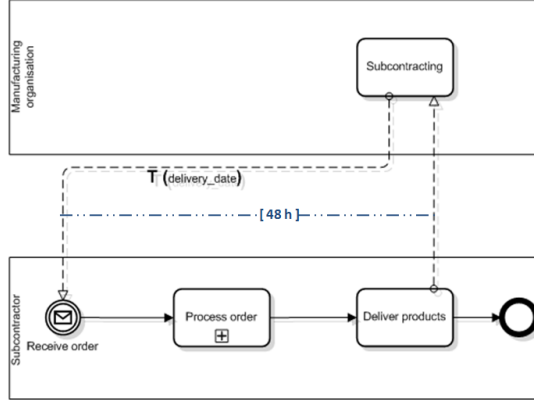
- **Exchanged Temporal Data :** In this subsection, we attempt to lay great stress upon the need to distinguish messages with temporal data from other kind of data in the specification step. The goal is the consequent use of these temporal data to specify complex temporal constraints and relationships among the activities of the process model. Hence, the message flow label **T** (*delivery\_date*) presented in Fig. 4 is the proposed notation to insist on the fact that *delivery\_date* is a temporal information.
- **Deadline of message exchange :** In an inter-organizational context, allowing to impose constraints on messages exchanged between the different partners seems to be necessary. For instance, 48h of delay is specified between the messages presented in Fig. 4.

Our work comes up with other interesting temporal constraints such as the deadline of a group of activities, the deadline of the business process or the deadline of inter-organizational business processes IOBPs. Furthermore, we accordingly tackle modelling issues related to temporal constraints correlated with both resource and data constraints. These modelling concepts have been omitted due to space limitation.

## 4 Temporal Constraints advertisement : The Time-aware Automatic Process View Generation TAPVG

Previously, we have shown the different temporal constraints we consider. In this section, we will present the approach of views generation from private processes.





**Fig. 4.** Exchanged temporal data and the deadline of message exchange temporal constraints

Indeed, process views are increasingly gaining importance in modern business process management. This setting aims to preserve the industrial privacy while engaging in inter-organizational collaborations. In an inter-organizational collaboration, each partner communicates and interacts through its process view while keeping private the underlying internal business process. It is neither necessary nor desirable to exhibit all details of the provider's internal process, and on the other hand, partners do not prefer to be overloaded by unnecessary data needless for their collaboration. By using views, organizations are allowed to expose as little information as possible but enough to well communicate with process partners. A private process can have many views each customized to one partner. Considering the different temporal constraints presented above while constructing process views is a tedious task and error prone. Our framework aims to automatically recalculate and propagate the corresponding temporal constraints from the private business process (called *private temporal constraints*) to a corresponding view (resp. called *public temporal constraints*). To do so, we consider two different operations: *Abstraction* and *Aggregation*.

Before explaining the steps to the temporal constraints advertisement, we consider some assumptions and introduce some definitions.

**Assumption:** In this paper, we assume a structured representation of process models [5]. Additionally, processes studied in this work are restricted to acyclic processes (without loops or cycles). The fact that it is possible to represent unstructured models in the BPMN notation does not limit the scope of our work. Indeed, the authors in [5] has shown that most unstructured process models can be automatically translated into structured ones.

In essence, a process model is represented as a tree whose leaves represent activities and whose internal nodes represent either events (eg. Start Event SE) or gateways (eg. sequence (SEQ), parallel (PAR)). Space limitations prevent presenting this tree data-structure.

We intend to formally capture the structured process models as follows.

**Definition 1: (Process Graph)**

Let  $\Gamma$  be a set of types of nodes. A Process Graph  $P$  is a tuple  $(N, E, \tau, \gamma)$ , in which:

- $N$  is the set of nodes;
- $E \subseteq N * N$  is the set of edges; and
- $\tau : N \rightarrow \Gamma$  is a function that maps nodes to their types
- $\gamma$  is the set of the temporal constraints of the process.

Actually,  $\Gamma$  supports the following types of nodes : activities (A), events (i.e. Start Event(SE) and End Event(EE)) and gateways (i.e. sequence(SEQ), parallel(PAR), inclusive(INCL) and exclusive(EXCL)).

Let  $P = (N, E, \tau, \gamma)$  be a process graph and  $N_i \in N$  be a node. We introduce some preliminary definitions related to the process:

**Definition 2: (pre\_set, post\_set, pre\_activity\_set, post\_activity\_set)**

- $\text{pre\_set}(N_i, P) = \{M_i \in N \mid \exists (M_i, N_i) \in E\}$ , denotes the pre\_set nodes of  $N_i$ ,
- $\text{post\_set}(N_i, P) = \{M_i \in N \mid \exists (N_i, M_i) \in E\}$ , denotes the post\_set nodes of  $N_i$ ,
- $\text{pre\_activity\_set}(N_i, P) = \{M_i \in N \mid \exists (M_i, N_i) \in E \wedge \tau(N_i) = A\}$ , denotes the activity nodes of the pre\_set nodes of  $N_i$ , and
- $\text{post\_activity\_set}(N_i, P) = \{M_i \in N \mid \exists (N_i, M_i) \in E \wedge \tau(N_i) = A\}$ , denotes the activity nodes of the post\_set nodes of  $N_i$ .

Meanwhile, for sake of simplicity, the proposed internal tree data-structure helps to define the following elementary functions :

**Definition 3: (parent-node, child-node, next-node)**

- $\text{parent\_node}(N_i, P)$  is a function that maps a node  $N_i$  to its parent node,
- $\text{child\_node}(N_i, P)$  is a function that maps a given node  $N_i$  to a node  $N_j$  such that : if  $N_i$  is in a sequential flow, the returned node  $N_j$  denotes the first node of the sequence. If the  $N_i$  node belongs to a gateway (i.e. PAR, INCL or EXCL) the different nested nodes (gateways or activities) are respectively returned.
- $\text{next\_node}(N_i, P)$  is a function used only to add an order to childs of a node sequence(SEQ). In other words, this function points to the next node of the sequential flow.

At first, the temporal constraints sets  $\gamma$  of the considered processes are restricted to the Duration temporal constraint. Considering other temporal constraints is beyond the scope of this paper. Accordingly, the function **Add\_constraint** ( $\gamma, TC_i$ ) serves to add the temporal constraints  $TC_i$  to the temporal constraint set  $\gamma$ .

Let  $A_i$  be an activity of the process (i.e.  $A_i \in N \wedge \tau(A_i) = A$ ); and the minimum (resp. maximum) duration  $MinD \in \mathbb{N}$  (resp. MaxD).

We denote that the duration of a given activity, say  $A_i$ , is in the interval  $[\text{minD}, \text{maxD}]$  by  $TC_i = \text{Duration}(A_i, \text{MinD}, \text{MaxD})$ .

Furthermore, let  $N_i$  (resp.  $N_j$ ) be nodes of the process (i.e.  $N_i \in N$ );  $TC_i = TD(t, A_i, A_j, \text{MinD}, \text{MaxD})$  is the proposed notation to mention a temporal de-

pendency constraint, with a type  $t \in \{SS : \text{Start to Start}, FS : \text{Finish to Start}, SF : \text{Start to Finish}, FF : \text{Finish to Finish}\}$ .

The first algorithm (Algorithm 1) ensures the calculation of the temporal constraints set for the process view resulting from an abstraction process. For that, the ABSTRACTION procedure takes, as input, the private process graph  $P_{pr}(N_{pr}, E_{pr}, \tau_{pr}, \gamma_{pr})$ , the source node ( $src$ ) and the destination node ( $dst$ ) (i.e. the node from which the abstraction process begins/ends). It proceeds by adding temporal dependency constraints Finish to Start (FS) to the temporal constraint set of the calculated view  $\gamma_v$ . The goal is to add a certain delay to the process view as depicted in Fig. 6. We differentiate between three major parts of algorithm 1. The first one aims at finding the source set  $src\_TC\_set$  of the added temporal constraint Finish to Start (FS). The first part (lines 5-21) is classified into four subparts. First subpart (lines 6-10) is devoted to source nodes which are in a sequential process flow and there are activities preceding them in the corresponding sequence flow. Second subpart (lines 11-15) deals with activities in gateways (i.e. PAR, INCL, EXCL). The Third subpart (lines 16-18) deals with source nodes which are the first activities in the sequential process in which they appear. Consequently, the fourth subpart (lines 19-21) adds the Start Event (SE) if the source node is the first activity of the process. The second part (line 22-38) is respectively classified into four subparts and aims at finding the destination set  $dst\_TC\_set$  of the added temporal constraint Finish to Start (FS).

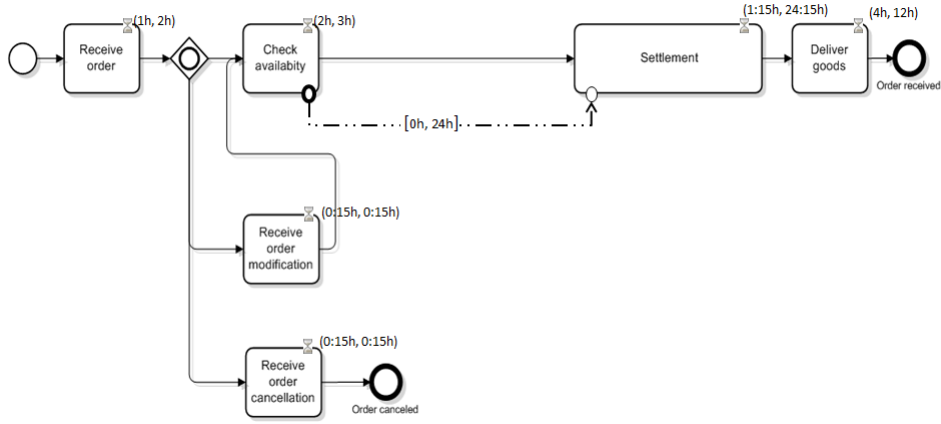
The third part (lines 39-46) is dedicated to calculate and add MaxD (resp. MinD) attributes of the FS temporal dependency using the *Max\_Duration* (resp. *Min\_Duration*) function.

Space limitations prevent a detailed exhibition of the *Min\_Duration* and *Max\_Duration* functions. A detailed version can be found in [?].

*Note :* In the case of activities, the  $src\_state$  Start (resp.  $dst\_state$  End) state denotes its beginning (resp. its firing). In the other case (i.e. in the case of gateways), the  $src\_state$  Start (resp.  $dst\_state$  End) state denotes the split (resp. the join) of the gateway.

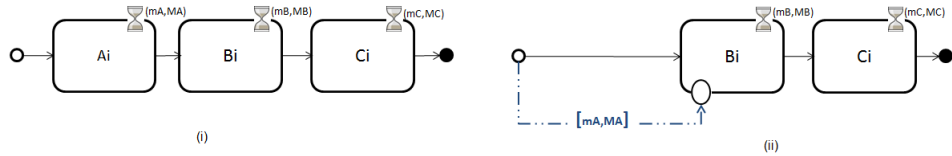
Algorithm 4 exhibits the AGGREGATION which calculate the temporal constraints set for the process view resulting from an aggregation process. The AGGREGATION procedure uses the *Min\_Duration* and *Max\_Duration* functions to calculate the minimum and maximum duration of the aggregation activity *Agg\_activity*. The calculated values MinD and MaxD are used to add a Duration temporal constraint to the temporal constraint set of the calculated view  $\gamma_v$ .

Consider the motivating example process presented in Fig. 1. We lay stress on the fact that the algorithms included in this paper are restricted to the Duration temporal constraint. Hence, only Duration constraints of Fig. 3 are considered as input for the AGGREGATION and ABSTRACTION algorithms. In order not to overload partners with unnecessary data, the designer can aggregate the *Ask for settlement* and *Receive settlement* activities using a unique activity; *Settlement* as an aggregation activity. Moreover, to preserve industrial privacy, the designer need to omit the *Subcontracting* activity from some partners' process views.



**Fig. 5.** The application of the TAPVG approach on the motivating example : the generated process view

Fig. 5 depicts the result of the application of the AGGREGATION algorithm followed by the ABSTRACTION algorithm on the given example. Therefore, our approach helps organizations to automatically recalculate and propagate the corresponding temporal constraints from private to public processes. Thus, our approach ensures the non-disclosure of professional secrecy and reduces the risk of error while generating business process views.



**Fig. 6.** A private business process model (i) and its generated process view (ii)

### 5 The Verification framework

The definition of temporal constraints allows to specify constrained process models that may encounter a deadlock situation due to inconsistencies between nested constraints. Our approach allows the formal verification of deadlock freedom and aims to verify other structural properties of of generated views (i.e. dead tasks and bottlenecks). Moreover, our work goes far beyond the simple verification of the structural properties of the generated views (eg. deadlock and

---

**Algorithm 1** The calculation of the temporal constraints set for the process view resulting from an Abstraction

---

```

1: procedure ABSTRACTION( $P_{pr}, src, dst, \gamma_v$ )
2:   Input  $P_{pr}(N_{pr}, E_{pr}, \tau_{pr}, \gamma_{pr}), src, dst$  /*the private process, the source and destination nodes*/
3:   Input/Output  $\gamma_v$  /*temporal constraints set of the process view*/
4:   local  $src\_TC\_set, dst\_TC\_set, MinD, MaxD$ 
5:   /*The calculation of the source set  $src\_TC\_set$  of added constraints*/
6:    $src\_TC\_set \leftarrow \phi$ 
7:   /*the source node  $src$  is in a sequential process flow*/
8:   if  $\tau_{pr}(parent\_node(src, P_{pr})) = SEQ$  then
9:     for all  $Activity A_i \in pre\_activity\_set(src, P_{pr})$  such that
10:     $parent\_node(src, P_{pr}) = parent\_node(A_i, P_{pr})$  do /*for all activities that precede  $src$  in the sequence*/
11:       $src\_TC\_set \leftarrow A_i$ 
12:    end for
13:  end if
14:  /* $src$  is not in a sequential flow or  $src$  is the first activity of the sequence*/
15:  if  $src\_TC\_set = \phi$  then
16:    for all  $Activity A_i \in N_{pr}$  such that  $post\_activity\_set(A_i, P_{pr}) = post\_activity\_set(src, P_{pr})$  do /* $src$  is in (PAR, INCL, EXCL)*/
17:       $src\_TC\_set \leftarrow A_i$ 
18:    end for
19:  end if
20:  if  $src\_TC\_set = \phi$  then /* $src$  is the first activity of the private process*/
21:     $src\_TC\_set \leftarrow SE$  /*the addition of the start Event*/
22:  end if
23:  /*The calculation of the destination set  $dst\_TC\_set$  of added constraints*/
24:   $dst\_TC\_set \leftarrow \phi$ 
25:  /*the destination node  $dst$  is in a sequential process flow*/
26:  if  $\tau_{pr}(parent\_node(dst, P_{pr})) = SEQ$  then
27:    for all  $Activity A_i \in post\_activity\_set(dst, P_{pr})$  such that
28:     $parent\_node(dst, P_{pr}) = parent\_node(A_i, P_{pr})$  do /*for all activities following  $dst$  in the sequence*/
29:       $dst\_TC\_set \leftarrow A_i$ 
30:    end for
31:  end if
32:  /* $dst$  is not in a sequential flow or  $dst$  is the last activity of the sequence*/
33:  if  $dst\_TC\_set = \phi$  then
34:    for all  $Activity A_i \in N_{pr}$  such that  $pre\_activity\_set(A_i, P_{pr}) = pre\_activity\_set(dst, P_{pr})$  do /* $dst$  is in (PAR, INCL, EXCL)*/
35:       $dst\_TC\_set \leftarrow A_i$ 
36:    end for
37:  end if
38:  if  $dst\_TC\_set = \phi$  then /* $dst$  is the last activity of the process*/
39:     $dst\_TC\_set \leftarrow EE$  /*the addition of the end Event*/
40:  end if
41:  /*The calculation and the addition of the FS constraint*/
42:  for all  $Noeud A_i \in src\_TC\_set$  do
43:    for all  $Noeud A_j \in dst\_TC\_set$  do
44:       $MinD \leftarrow Min\_Duration(A_i, End, A_j, Start, P_{pr})$ 
45:       $MaxD \leftarrow Max\_Duration(A_i, End, A_j, Start, P_{pr})$ 
46:      if  $MinD \geq 0$  and  $MaxD \geq 0$  then
47:        Add_constraint( $\gamma_v, TD(FS, A_i, A_j, MinD, MaxD)$ )
48:      end if
49:    end for
50:  end for
51: end procedure

```

---

---

**Algorithm 2** The calculation of the Duration Temporal Constraint of a given Aggregation Activity

---

```

1: procedure AGGREGATION(src,dst,Ppr,Agg_activity,γv)
2:   Input src, dst, Ppr, Agg_activity
3:   Input/Output γv /*temporal constraints set of the private process*/
4:   local MinD, MaxD
5:   MinD ← Min_Duration(src, Start, dst, End, Ppr)
6:   MaxD ← Max_Duration(src, Start, dst, End, Ppr)
7:   if MinD ≥ 0 and MaxD ≥ 0 then Add_constraint
      (γv, Duration(Agg_activity, MinD, MaxD))
8:   end if
9: end procedure

```

---

dead task). Precisely, we aim the verification of user-defined temporal constraints such as deadline constraints of the public views. For instance, the designer can verify delays between two activities  $A_i$  and  $A_j$  of a view or between the start of the process view and its end. We argue that it is not enough to verify the views separately. Indeed, temporal constraints violations can eventually happen during the collaboration of a set of generated process views.

In this context, we use the real-time model checker UPPAAL for the formal verification of process views. The UPPAAL model checker proves systems that can be modelled as timed automata(TA) against a desired set of properties defined using a rich subset of CTL (computation tree logic). Thus, as a first step, process views are mapped onto the networks of timed automata. The mapping outlined in this paper builds upon our previous work [6].

Based on the generated UPPAAL models, we performed the verification of the following CTL properties:

**A[] not deadlock:** to ensure deadlock freeness of the process,

**A[] (ProcessView.EndProcess imply  $t1 \leq 60$ ):** to verify the process view deadline is met.

Finally, both the Timed Automata models and queries for temporal constraints are input into the Uppaal model checking engine. For our example, we notice that properties already verified on the private process are also verified on the process view as well. By using our verification process, every organization can verify the correctness of the temporal constraints of both its private and public processes. Indeed, it provides means to ensure that once a set of requirements are already verified on the private process, these requirements are further verified on the process view generated by the TAPVG approach.

## 6 Related works

Representing and reasoning with temporal requirements has long been a subject of research in business process management area. Hence, there are been several attempts to model a variety of temporal constraints using the defacto industrial standard for business process modeling, BPMN [4, 1, 3, 8]. Nevertheless, little

consideration is given to the depiction of complex temporal constraints for concurrent business processes. When considering the business process model itself, there are several research efforts specifying different constraints for one business process. Others take into account the cooperation between more than one process both in the web service composition field [9, 8, 10] and in the inter-organizational business process field [11–15]. When addressing the issue of IOBP, it is worthwhile to consider the migration of the different constraints between the private and the public processes.

Consequently, the authors in [16, 17] focus on deriving a process view from a given private process. In most of these works, the authors focus on the setting of rules and algorithms to the construction of process views. In the context of Inter-Organizational cooperation, Eder and Tahamtan [14, 15] use the concept of temporal plans and use timed activity graphs to model private as well as process views. The proposed approach checks if the interorganizational workflow is temporally consistent by checking if its participating views are temporally consistent. Compared to our work, they neither provide algorithms detailing how process views are constructed nor consider temporal constraints crossing the boundary of an activity such as inter-activities temporal constraints. Furthermore, in [14, 15], the duration of activities is presented by deterministic values which limits the scope of this work. The authors in [12, 13] propose an extension based on Time Petri nets for modeling and advertising temporal requirements for cooperative activities on the process views. Nevertheless, this work permits only to add deadline constraints between cooperative activities (which are visible by the cooperation candidates). Obviously, it is not enough compared to the rich set of temporal constraints proposed in our approach. The key feature of the approach detailed in this paper is to use the power of constraints while still preserving the industrial privacy.

We turn our attention to the temporal requirements verification issue. Temporal verification mechanisms are of paramount importance since they enable to detect, early on, possible temporal conflicts and to react to them effectively. Indeed, many efforts confound the time conflict verification of the model (i.e. the violation of some temporal constraints) with the structure verification (i.e. the analysis of dead tasks, deadlocks, bottlenecks and loops) [7, 3]. In contrast, there are some works which have tried to differentiate the two verification processes [18–20]. To deal with the time conflict verification, there are some works which have neglected the intra-activity temporal requirements (eg. the duration of the process activities) [18]. On the other hand, there are some works which have neglected the inter-activity temporal dependencies [3, 19]. Otherwise the approach detailed in [20] has tried to include the two different temporal requirements.

Our proposed approach presents a helpful tool in hand for process designers and managers to specify and safely advertise a large set of temporal constraints. It enables moreover to foresee the possible temporal violations and react to them predictively.

## 7 Conclusion and future work

In this paper, we exhibited the weakness of the defacto industrial standard for process modelling, BPMN, to represent the temporal dimension of processes. We particularly presented how the temporal dimension can be specified at design time, through temporal constraints. We do not claim to provide an exhaustive list of extensions required for the modelling of any possible temporal constraint.

In this context, we have shown the added value of the proposed BPMN temporal extension by means of its application to a purchase order handling process. Besides, based on the proposed timed model, we have presented algorithms which enable generating process views from private business process through automatic recalculation and propagation of temporal constraints. Finally, we tackled with the verification process at design time using the formal formalism of Timed Automata and the UPPAAL model checking tool. Our verification framework aims to assist in mitigating risk and facilitate the early discovery of temporal violations during the business process management life-cycle.

Currently, we are working on tool support for the presented BPMN extensions. We are implementing a prototype using the Activiti BPMN Platform (<http://www.activiti.org/>). Indeed, we plan to further investigate on this problem to allow the later automation of the proposed extensions through process execution engines. In addition, we plan to extend the TAPVG framework to handle a larger set of temporal constraints beyond those presented in this paper. Another avenue of research we are working on is to define violation identification mechanisms and to propose relevant primitives to resolve violations.

## References

1. Flores, C., Sepúlveda, M.: Temporal Specification of Business Processes through Project Planning Tools. In: Proceedings of the International Workshops and Education Track on Business Process Management Workshops (BPM). Volume 66 of LNBIP., Springer (2010) 85–96
2. : Object Management Group (OMG), Business Process Model and Notation (BPMN), Version 2.0 (January 2011)
3. Watahiki, K., Ishikawa, F., Hiraishi, K.: Formal Verification of Business Processes with Temporal and Resource constraints. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, IEEE (2011) 1173–1180
4. Gagné, D., Trudel, A.: Time-BPMN. In: Proceedings of the IEEE Conference on Commerce and Enterprise Computing (CEC), IEEE Computer Society (2009) 361–367
5. Polyvyanyy, A., García-Bañuelos, L., Dumas, M.: Structuring Acyclic Process Models. In: Business Process Management. Volume 6336 of LNCS., Springer (2010) 276–293
6. Guermouche, N., Godart, C.: Timed Conversational Protocol Based Approach for Web Services Analysis. In: Proceedings of the 8th International Conference on Service-Oriented Computing, ICSOC. Volume 6470 of LNCS., Springer (2010) 603–611



7. Kallel, S., Charfi, A., Dinkelaker, T., Mezini, M., Jmaiel, M.: Specifying and Monitoring Temporal Properties in Web Services Compositions. In Eshuis, R., Grefen, P.W.P.J., Papadopoulos, G.A., eds.: Proceedings of the 7th IEEE European Conference on Web Services (ECOWS), IEEE Computer Society (2009) 148–157
8. Guermouche, N., Zilio, S.D.: Towards Timed Requirement Verification for Service Choreographies. In: 8th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, Pittsburgh, États-Unis (October 2012) 10
9. Kazhamiakin, R., Pandya, P.K., Pistore, M.: Representation, Verification, and Computation of Timed Properties in Web. In: Proceedings of the IEEE International Conference on Web Services (ICWS ), IEEE Computer Society (2006) 497–504
10. Benattallah, B., Casati, F., Ponge, J., Toumani, F.: On Temporal Abstractions of Web Service Protocols. In: Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE). Volume 161 of CEUR Workshop Proceedings., CEUR-WS.org (2005) 13–17
11. Chebbi, I., Dustdar, S., Tata, S.: The view-based approach to dynamic inter-organizational workflow cooperation. *Data & Knowledge Engineering* **56**(2) (2006) 139–173
12. Makni, M., Hadj-Alouane, N.B., Tata, S., Yeddes, M.M.: Negotiating Deadline Constraints in Inter-organizational Logistic Systems: A Healthcare Case Study. In: Proceedings of the International Workshops and Education Track on Business Process Management Workshops (BPM). Volume 100 of LNBIP., Springer (2011) 108–118
13. Makni, M., Tata, S., Yeddes, M.M., Hadj-Alouane, N.B.: Satisfaction and Coherence of Deadline Constraints in Inter-Organizational Workflows. In: Proceedings of the Confederated International Conferences: CoopIS on On the Move to Meaningful Internet Systems: OTM 2010. Volume 6426 of LNCS., Springer (2010) 523–539
14. Eder, J., Tahamtan, A.: Temporal Conformance of Federated Choreographies. In Bhowmick, S.S., Küng, J., Wagner, R., eds.: Proceedings of the 19th International Conference on Database and Expert Systems Applications (DEXA). Volume 5181 of LNCS., Springer (2008) 668–675
15. Eder, J., Tahamtan, A.: Temporal Consistency of View Based Interorganizational Workflows. In Kaschek, R., Kop, C., Steinberger, C., Fliedl, G., eds.: Proceedings of the 2nd International United Information Systems Conference on Information Systems and e-Business Technologies (UNISCON). Volume 5 of LNBIP., Springer (2008) 96–107
16. Zhao, X., Liu, C., Sadiq, W., Kowalkiewicz, M., Yongchareon, S.: Implementing process views in the web service environment. *World Wide Web* **14**(1) (2011) 27–52
17. Eshuis, R., Grefen, P.W.P.J.: Constructing customized process views. *Data & Knowledge Engineering* **64**(2) (2008) 419–438
18. Guermouche, N.: Etude des Interactions Temporisées dans la Composition de Services Web. PhD thesis, Université Henri Poincaré (2010)
19. Huai, W., Liu, X., Sun, H.: Towards Trustworthy Composite Service Through Business Process Model Verification. In IEEE, ed.: Proceedings of the 7th International Conference on Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC). (2010) 422 – 427
20. Du, Y., Xiong, P., Fan, Y., Li, X.: Dynamic Checking and Solution to Temporal Violations in Concurrent Workflow Processes. *IEEE Transactions on Systems, Man, and Cybernetics, Part A* **41**(6) (2011) 1166–1181