



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, Coventry, United Kingdom. 6p. hal-00805532v2

HAL Id: hal-00805532

<https://hal.science/hal-00805532v2>

Submitted on 20 Dec 2013

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.

Time-aware Automatic Process View Generation

Saoussen Cheikhrouhou¹, Slim Kallel¹, Nawal Guermouche^{2,3} and Mohamed Jmaiel¹

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

²CNRS-LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France

³Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France

Abstract—Nowadays, the OMG standard business process model and notation BPMN 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 *views* is essential since it allows 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 temporal 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 this extension, our work preserves privacy in inter-organizational business processes (IOBPs) by a Time-aware Automatic Process View Generation TAPVG approach. Finally, a verification approach based on the model checking technique is used to diagnose potential temporal violations of the process model.

I. 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*. Failing to consider temporal information in process models turns out in higher process execution costs. 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 run the risk of losing their competitive edge. By disclosing their know-how, some 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 temporal properties during processes views generation. In the premise of ensuring the correctness of already defined process models, namely the

private and public processes, a verification approach based on the model checking technique is used. To summarize, the work presented in this paper aims to assist stakeholders and system implementers with a Time-aware Automatic Process View Generation (TAPVG) approach.

This paper is organized as follows. A motivating example is introduced in Section II. Section III presents a brief description of the proposed BPMN temporal extension and details the TAPVG approach. Section IV outlines the use of a formal verification approach to detect the temporal violations of process models. A review of related literature is given in Section V. Finally, Section VI concludes.

II. MOTIVATING EXAMPLE

Let us consider the BPMN diagram of a manufacturing organization, say organization A, depicted in Figure 1. The process is triggered when a customer submits a purchase order (*Receive order*). Then, he can check whether the ordered articles are available or not (*Check availability*). If the ordered articles are not available in stock, the organization needs to launch a subcontracting activity (*Subcontracting*). Following that, the customer is asked for financial settlement (*Receive settlement*) and the goods are subsequently prepared for further shipment (*Prepare shipment*). The goods have to undergo an export handling procedure (*Export handling*) and a security check (*Security check*). If all checks are fulfilled successfully, the goods are finally shipped and the process meets its end.

As far as we consider B2B applications, the manufacturing organization (organization A) need to collaborate with external partners through its process (depicted in Figure. 1) which in turn may take part in the interorganizational business process through their process views. We assume that each partner has a private process which is only visible to its own business entities. Nevertheless, external parties have no knowledge about the process structure and internal logic of the private 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. A private process can have many views each customized to one partner. Exposing the private process (see Figure 1) while collaborating with a potential customer, the organization A have to divulgate some private information such as the *Subcontracting* activity. Unfortunately, divulgating this information seems to be fraught for potential customers and may decrease customer loyalty. Furthermore, organization A prefers to hide the delivery details of its private process since they are useless for the communication with the considered customer. So organization A needs to generate a *customer*

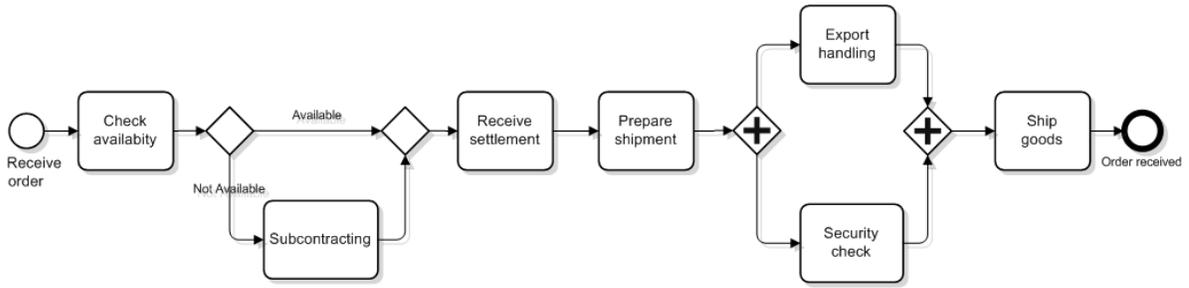


Fig. 1. The purchase order handling process of a manufacturing organization A

process view (as depicted in Figure 2) from its private process (see Figure 1). The omitted details of the private process of organization A, are considered to be needless for communicating with the process of the customer. In contrast, the latter details can be of a major importance while communicating with a process of logistics services provider and vice versa. A provider of logistics services is not interested in details of how the purchase order is handled. All activities regarding the shipment operation are however of paramount importance.

Within business processes, the temporal perspective is crucial since temporal constraints must be respected. All business experts agree upon the fact that time is a key resource for processes within organizations. Unfortunately, the de facto standard BPMN lacks for means to specify the turnaround time of business activities such as the minimum and maximum execution times. Given this limitation, the authors in [2] extended BPMN by minimum and maximum attributes for the process model activities. We have extended their work by proposing an activity decorator with the minimum (*MinD*) and maximum (*MaxD*) duration values as depicted in Figure 3.

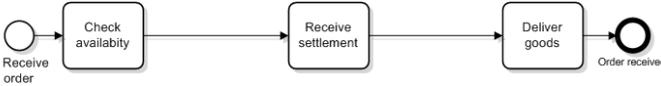


Fig. 2. The customer process view

We assume now that each activity of the private process of organization A has a duration constraint. Figure 4 shows the process of the *manufacturing organization* enriched with *Duration* temporal constraints. Consequently, the integration of time constraints in inter-organizational business processes is an important issue. Since each organization exposes a customized version of his private process (i.e a process view), some information is kept hidden and not visible to all partners. The inter-enterprise business process is obtained by joining process views that have relevant roles within the context of the global operation. Assume a situation when a process view is defined and this view takes part in a given interorganizational business process. It is necessary to calculate the temporal constraints of the views in order to avoid temporal conflicts and exceptions. Indeed, it must be clear for the other partners in which time they can send and expect messages. Therefore, adding the needed temporal constraints for the generated *customer process view* (see Figure 2) seems to be of paramount importance.

Assisting business designers to automatically generate process views from private processes while considering temporal



Fig. 3. The Duration temporal constraint for the activity A_i

constraints is still a challenging task.

III. THE TIME-AWARE AUTOMATIC PROCESS VIEW GENERATION TAPVG

In this section, we present an approach to automatically generate process views from private processes. 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 private business process. 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 temporal constraints of the private process 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.

Definition 1: Duration constraint

Let $s(A_i)$ (resp. $e(A_i)$) be the starting (resp. the ending time) of the activity A_i . Let $MinD$ and $MaxD$ be two relative time values representing respectively the minimum and maximum durations of an activity A_i . The Duration constraint $Duration(A_i, MinD, MaxD)$ is defined as :

$$MinD \leq e(A_i) - s(A_i) \leq MaxD$$

It is obvious that for each activity A_i ($MinD \leq MaxD$). Also, in a case where $MinD = MaxD$, the activity A_i has precise duration.

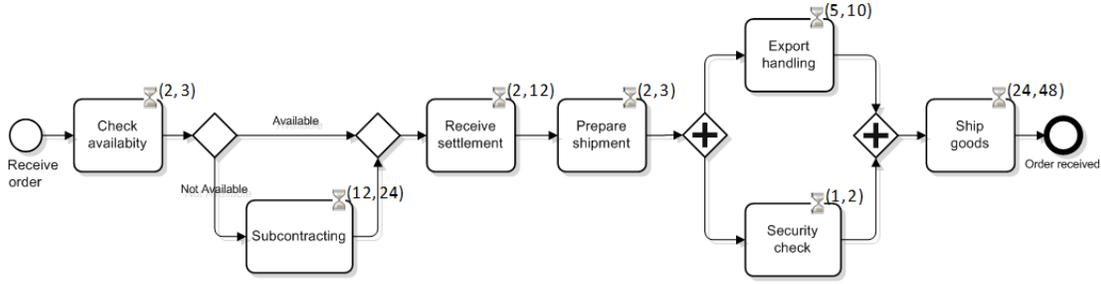


Fig. 4. The purchase order handling process enriched with proposed temporal constraints

Assumption: We assume a structured representation of process models.

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 [4] showed 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)). We formally capture the structured process models as follows.

Definition 2: Process Graph

Let Γ be a set of types of nodes. A Process Graph P is a tuple (N, E, τ, γ) , 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
- γ is the set of the temporal constraints of the process.

Actually, Γ 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 3: pre_set, post_set, pre_activity_set, post_activity_set

- $pre_set(N_j, P) = \{N_i \in N \mid \exists (N_i, N_j) \in E\}$, denotes the predecessor nodes of N_j ,
- $post_set(N_i, P) = \{N_j \in N \mid \exists (N_i, N_j) \in E\}$, denotes the successor nodes of N_i ,
- $pre_activity_set(N_j, P) = \{N_i \in N \mid \exists (N_i, N_j) \in E \wedge \tau(N_i) = A\}$, denotes the activity nodes of the predecessor nodes of N_j , and
- $post_activity_set(N_i, P) = \{N_j \in N \mid \exists (N_i, N_j) \in E \wedge \tau(N_j) = A\}$, denotes the activity nodes of the successor nodes of N_i .

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

Definition 4: parent-node, child-node, next-node

- parent-node (N_i, P) is a function that maps a node N_i to its

parent node,

- 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 node N_i belongs to a gateway (i.e. PAR, INCL or EXCL) the different nested nodes (gateways or activities) are respectively returned.

- next-node (N_i, P) is a function used only to add an order to children of a node sequence(SEQ). In other words, this function points to the next node of the sequential flow.

Let us now introduce the finish-to-start temporal constraint necessary to build timed process views. The *Temporal Dependency Finish to Start (FS)* constraint stands for a proposed BPMN extension to offer an explicit way to depict dependencies between two activities. A temporal dependency is a relationship between two activities, say A_i and A_j , in order to coordinate their starting and finishing times. The temporal dependency suggested in this paper enhances the expressiveness of previous proposals in representing such constraint. In [3], the authors proposed the Time-BPMN approach. They assigned lead and lag times to temporal dependency relations. Compared to our work, we propose a fine grained temporal dependency constraint as follows. Figure 5 shows the proposed BPMN notation to depict the proposed *Temporal Dependency Finish to Start (FS)* constraint.

Definition 5: Finish-to-Start Temporal constraint

Let $s(A)$ (rep. $e(A)$) be the starting (resp. ending) execution time of an activity A, the Finish-to-Start Temporal Dependency constraint of two activities A_i, A_j , denoted $TD(FS, A_i, A_j, MinD, MaxD)$, is defined as :

$$MinD \leq s(A_j) - e(A_i) \leq MaxD$$

The latter definition denotes that the activity A_j should starts its execution no later than $MaxD$ time units and no earlier than $MinD$ time units after the activity A_i ends.

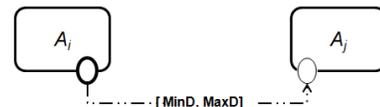


Fig. 5. The customer process view enriched with the necessary temporal constraints

In the rest of this section, we explain the steps of the ABSTRACTION and AGGREGATION processes.

A.ABSTRACTION: The ABSTRACTION process consists in making some parts of the process invisible to other external observer. Parts of the process that are not interesting for or do not contribute to the interaction with another partner are intended to be hidden. Furthermore, privacy issues would be the aim of making some parts of the process unobservable. For example, the private process of organization A (see Figure 4) contains the activity *Subcontracting*. After making this activity invisible in the view of the process (i.e. the *customer process view* in Figure 2), the customer can not see the activity *Subcontracting*. To enable timed views generation from timed private processes by abstraction and/or aggregation, timed duration constraints are propagated into Finish-to-Start temporal dependency constraints. Figure 6 shows the result of timed constraints propagation of timed duration constraints into Finish-to-Start temporal Dependency when generating the customer process view.

B.AGGREGATION: The AGGREGATION process consists in using aggregation activities for constructing views. Some executable activities of the private process can be grouped into a so-called aggregation activity. Figure 6 illustrates an exemplary application of aggregation. The aggregation activity *Deliver goods* aggregates the activities (*Prepare shipment*, *Export handling*, *Security check* and *Ship goods*).

Given a private process graph $P_{pr}(N_{pr}, E_{pr}, \tau_{pr}, \gamma_{pr})$, a source node (*src*) and a destination node (*dst*) (i.e. the node from which the abstraction/aggregation process begins/ends) where γ_{pr} is the set of duration temporal constraints $\gamma_{pr} = D_{pr}$ with $D_{pr} = \{ TC_i \mid TC_i = Duration(N_i, MinD, MaxD) \wedge \tau_{pr}(N_i) = A \}$, the result of process abstraction/aggregation is a process graph view $P_v(N_v, E_v, \tau_v, \gamma_v)$, where the set of temporal constraints γ_v is defined by the set of duration and temporal dependency constraints $\gamma_v = D_v \cup T_v$ such as $D_v = \{ TC_i \mid TC_i = Duration(N_i, MinD, MaxD) \wedge \tau_v(N_i) = A \}$ and $T_v = \{ TC_i \mid TC_i = TD(FS, N_i, N_j, MinD, MaxD) \wedge \tau_v(N_i) \in \{SE, A\} \wedge \tau_v(N_j) \in \{EE, A\} \}$.

The ABSTRACTION algorithm presented in Figure 7 proceeds by adding temporal dependency constraints Finish to Start (FS) to the temporal constraints set of the calculated view γ_v . The goal is to add a certain delay to the process view as depicted in Figure. 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.

A detailed version exhibiting the *Min_Duration* and *Max_Duration* functions can be found in [5].

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.

The AGGREGATION algorithm presented in Figure 7 calculates 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 durations 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 γ_v .

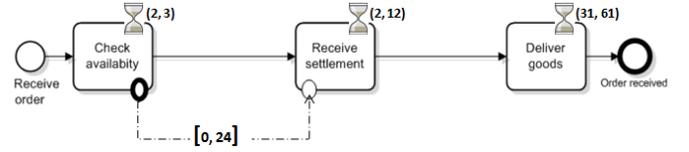


Fig. 6. The customer process view enriched with temporal constraints resulting from the application of the TAPVG approach

Consider the motivating example process presented in Figure 4. While collaborating with their potential customers, the organization A does not want to overload them with data unnecessary for their communication. For that aim, the designer can aggregate all activities related to goods shipment (i.e. *Prepare shipment*, *Export handling*, *Security check* and *Ship goods*) in a unique aggregation activity *Deliver goods*. Moreover, to preserve industrial privacy, the designer needs to omit the *Subcontracting* activity. Figure 6 depicts the result of the application of the AGGREGATION algorithm followed by the ABSTRACTION algorithm on the given example. The resulted customer process view presents the addition of a temporal constraint Finish to Start (FS) as follows: $TD(FS, Check\ availability, Receive\ settlement, MinD, MaxD)$ with $MinD = 0$ time units and $MaxD = 24$ time units. Additionally, a Duration temporal constraint is added for the aggregation activity *Deliver goods*:

$Duration(Deliver\ goods, MinD, MaxD)$ $MinD = 31$ time units ($31 = 2 + 5 + 24$) and $MaxD = 61$ time units ($61 = 3 + 10 + 48$). 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.

IV. 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 temporal constraints. Our approach allows the verification of deadlock freedom. Moreover, our work goes far beyond the simple verification of the structural properties of the generated views

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 sequence*/
21:     $src\_TC\_set \leftarrow pre\_activity\_set(src, P_{pr})$ 
22:  end if
23:  if  $src\_TC\_set = \phi$  then /* $src$  is the first activity of the private process*/
24:     $src\_TC\_set \leftarrow SE$  /*the addition of the start Event*/
25:  end if
26:  /*The calculation of the destination set  $dst\_TC\_set$  of added constraints*/
27:   $dst\_TC\_set \leftarrow \phi$ 
28:  /*the destination node  $dst$  is in a sequential process flow*/
29:  if  $\tau_{pr}(parent\_node(dst, P_{pr})) = SEQ$  then
30:    for all  $Activity A_i \in post\_activity\_set(dst, P_{pr})$  such that
31:     $parent\_node(dst, P_{pr}) = parent\_node(A_i, P_{pr})$  do /*for all activities following  $dst$  in the sequence*/
32:       $dst\_TC\_set \leftarrow A_i$ 
33:    end for
34:  end if
35:  /* $dst$  is not in a sequential flow or  $dst$  is the last activity of the sequence*/
36:  if  $dst\_TC\_set = \phi$  then
37:    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)*/
38:       $dst\_TC\_set \leftarrow A_i$ 
39:    end for
40:  end if
41:  if  $dst\_TC\_set = \phi$  then /* $dst$  is the last activity of the sequence*/
42:     $dst\_TC\_set \leftarrow post\_activity\_set(dst, P_{pr})$ 
43:  end if
44:  if  $dst\_TC\_set = \phi$  then /* $dst$  is the last activity of the process*/
45:     $dst\_TC\_set \leftarrow EE$  /*the addition of the end Event*/
46:  end if
47:  /*The calculation and the addition of the FS constraint*/
48:  for all  $Noeud A_i \in src\_TC\_set$  do
49:    for all  $Noeud A_j \in dst\_TC\_set$  do
50:       $MinD \leftarrow Min\_Duration(A_i, End, A_j, Start, P_{pr})$ 
51:       $MaxD \leftarrow Max\_Duration(A_i, End, A_j, Start, P_{pr})$ 
52:      if  $MinD \geq 0$  and  $MaxD \geq 0$  then
53:         $Add\_constraint(\gamma_v, TD(FS, A_i, A_j, MinD, MaxD))$ 
54:      end if
55:    end for
56:  end for
57: end procedure

```

Fig. 7. The Abstraction algorithm

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

```

1: procedure AGGREGATION( $src, dst, P_{pr}, Agg\_activity, \gamma_v$ )
2:   Input  $src, dst, P_{pr}, Agg\_activity$ 
3:   Input/Output  $\gamma_v$  /*temporal constraints set of the private process*/
4:   local  $MinD, MaxD$ 
5:    $MinD \leftarrow Min\_Duration(src, Start, dst, End, P_{pr})$ 
6:    $MaxD \leftarrow Max\_Duration(src, Start, dst, End, P_{pr})$ 
7:   if  $MinD \geq 0$  and  $MaxD \geq 0$  then  $Add\_constraint$ 
8:     ( $\gamma_v, Duration(Agg\_activity, MinD, MaxD)$ )
9:   end if
10: end procedure

```

Fig. 8. The Aggregation algorithm

(eg. deadlock). Precisely, we ensure 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]. Figure 9 shows the mapping onto TA of the customer process view resulting from the application of the TAPVG approach (the process diagram is presented in Figure6).

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.OrderReceived imply $t_0 \leq 100$): 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.

V. RELATED WORKS

Modeling and managing temporal requirements has long been a topic of intensive researches in business process management area. Hence, there have been several attempts to model a variety of temporal constraints using the defacto industrial standard for business process modeling, BPMN [3], [2], [1]. Several research efforts take into account the cooperation between more than one process in the inter-organizational business process field [8], [9], [10], [11]. Nevertheless, when addressing the issue of inter-organizational business process field, little consideration is given to temporal constraints associated with process views [11], [9].

In [12], [13], the authors aim at deriving a process view from a given private process. In these works, the authors focus on the setting of rules and algorithms to the construction of process views. Nevertheless, they do not consider the migration of temporal constraints associated with private processes to their corresponding process views. In the context of Inter-Organizational cooperation, the authors in [9], [10] propose an extension based on Time Petri nets for modeling and advertising temporal requirements for cooperative activities on process views. The authors demonstrate how missing deadlines while delivering the required services may cause a global failure execution, even if the business behavior complementarity of

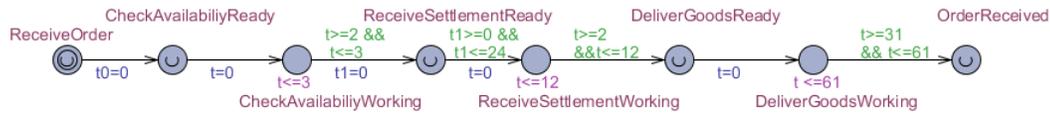


Fig. 9. The mapping of the customer process view onto Timed Automata

the involved services is ensured. This work allows modeling and propagating only one deadline constraint per workflow. Nevertheless, this work permits only to add deadline constraints between activities of the view. Eder and Tahamtan [11] use the concept of temporal plans and use timed activity graphs to model private as well as process views. By calculating the temporal execution plan of the view, the authors propagate the duration temporal constraint from private to public processes. The temporal execution plan defined by the authors allows to compute deadlines according to the start execution time of processes. Consequently, to enable for example views consistency checking, views must start executing at the same time. This is a restrictive assumption. According to our work, we aim at enabling timed views generating from timed private processes independently of starting time of processes. To do so, when driving process views, we use temporal dependency and duration constraints.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed the TAPVG approach which aims at enabling timed process views generating from timed private processes. To do so, we first propose a temporal BPMN extension for capturing temporal duration and temporal dependency constraints. Based on this extension, we proposed algorithms to derive public process views from private processes through automatic recalculation and propagation of the duration temporal constraints into temporal dependency constraints in process views. Finally, we rely on a model checking based verification approach to detect problems such as temporal conflicts. Particularly, we have used the timed automata formalism and the UPPAAL model checker.

Our verification framework aims to assist in mitigating risk and facilitate the early discovery of temporal violations during business process management life-cycle. Throughout this paper, we have used a manufacturing business process case study to show the different steps of the proposed approach.

Indeed, we plan to further investigate on this problem to allow the automation of the proposed BPMN extensions through process execution engines. In addition, we plan to extend the TAPVG framework to handle a larger set of temporal constraints beyond the *Duration* temporal constraint 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 them.

ACKNOWLEDGMENT

Part of this work has been supported by FP7-ICT IMAG-INE research and development project, co-funded by the

European Commission under the “Virtual Factories and Enterprises” (FoF-ICT- 2011.7.3, Grant Agreement No: 285132).

REFERENCES

- [1] Lanz, A., Weber, B., Reichert, M.: Workflow Time Patterns for Process-Aware Information Systems. In: Enterprise, Business-Process and Information Systems Modeling. Volume 50 of LNCS., Springer (2010) 94–107
- [2] Watahiki, K., Ishikawa, F., Hiraiishi, 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
- [3] Gagné, D., Trudel, A.: Time-BPMN. In: Proceedings of the IEEE Conference on Commerce and Enterprise Computing (CEC), IEEE Computer Society (2009) 361–367
- [4] Polyvyanyy, A., García-Bañuelos, L., Dumas, M.: Structuring Acyclic Process Models. In: Business Process Management. Volume 6336 of LNCS., Springer (2010) 276–293
- [5] Cheikhrouhou, S., Kallel, S., Guermouche, N.: The Time-Aware Automatic Process View Generation TAAVPG: Abstraction and Aggregation algorithms considering only the Duration temporal constraint. Technical report, http://www.redcad.org/projects/TR_SC001_2013.pdf. ENIS_LAAS (2013)
- [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] 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., CEUR-WS.org (2005) 13–17
- [8] Chebbi, I., Dustdar, S., Tata, S.: The view-based approach to dynamic inter-organizational workflow cooperation. *Data & Knowledge Engineering* **56**(2) (2006) 139–173
- [9] 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
- [10] 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
- [11] Eder, J., Tahamtan, A.: Temporal Consistency of View Based Inter-organizational 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
- [12] 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
- [13] Eshuis, R., Grefen, P.W.P.J.: Constructing customized process views. *Data & Knowledge Engineering* **64**(2) (2008) 419–438