# Using the RTPN model for the modelling of complex Workflow systems

Abdia Hamdani
Ibn-Khaldoun University of Tiaret.
hh_abla@yahoo.fr

Abdelkrim Abdelli
LSI laboratory, USTHB university, Algiers.
Abdelli@lsi-usthb.dz



## Abstract

Dealing with synchronization in time constrained workflow is becoming a challenging issue. In this paper, we present a modelling approach based on Petri nets formalism for timed workflow systems with complex synchronization among tasks of different privileges (Master/Slave). To this aim, we consider the concept of rendezvous already introduced in the RTPN (Time Petri Nets with rendezvous), to define a subclass of RTPNs called Time Workflow-nets with Rendezvous (RTWF-nets). We discuss how this model can cover a large range of timed synchronization patterns in a very smart and compact framework.

**Keywords** – Workflow, Real-time systems, Rendezvous, Petri nets, Patterns.

## 1 Introduction

In the late of the 70's, the research on workflow systems has started and a lot of approaches and powerful tools have been proposed and developed. Generally, *workflows* represent processes describing how and when their elementary tasks should be accomplished, thus describing the *control flow* of the workflow system. This may include different mechanisms ( e.g., sequence, choice, parallelism and synchronizations), usually called workflow *patterns* [Aal05]. For instance, synchronization in workflow system can be seen as a meeting point during the process where a set of tasks has to wait for others according to a given scheme (e.g the AND-join synchronizer pattern). Nowadays, the real challenge in workflow systems is to deal with situations where a variable number of tasks is processed under different synchronization and time constraints patterns. Indeed, adapting, replanning, and synchronizing workflows in response to an unexpected progress, delays, or technical conditions are necessary to maintain the safety of the systems. Furthermore, such requirements are becoming critical aspects in many domains as for example healthcare workflows [Car09]. For example, in transplantation surgery activity, we require the concurrent presence of the organ to be implanted, blood for the patient and the patient, that must arrive within the same hour at the hospital to avoid their functional degradation.

In this paper, we propose the use of RTPN (Time Petri Nets with Rendezvous) [Ham17], for the modelling and the analysis of complex workflow systems that include synchronization and time constraints among tasks. The $RTPN$ introduces the paradigm of rendezvous to define various synchronization schemes under different time constraint orchestrations. With its expressiveness power, RTPN provides a compact framework to represent complex workflow systems in elegant way, that could hardly be handled by other existing models. After presenting and discussing the workflow patterns provided by RTPN, we define a subclass of RTPN, called RTWFN (Time Workflow-nets with Rendezvous ), which is dedicated for the



In:Proceedings of the 3rd Edition of the International Conference on Advanced Aspects of Software Engineering (ICAASE18), Constantine, Algeria, 1,2-December-2018, published at http://ceur-ws.org

specification of workflow systems.
The remaining of this paper is organized as follows: Section 2 discusses the related works. In section 3, we present our workflow model with timed rendezvous patterns. Section 4 presents the RTPN and RTWFN models before presenting the modelling approach. In section 5, case-study examples are presented. Finally, conclusions and comments on future work are given.

## 2 Related Works

In this section, we review the key works defined in the literature that address workflow systems modelling and analysis.

**a)-Petri Nets based Models :**
Basic Petri nets have been used for the representation, the validation and the verification of business procedures in [Aal05]. In [Aal98], the authors introduce the *WorkFlow net* (WF-net) to specify the processing of a workflow. Afterwards, the WF-net has been extended to deal with data, time and other aspects. A various extensions are proposed: *(i)* the extension with time [Ada98]; *(ii)* the extension with color to model data [Rus09] ; *(iii)* the extension with hierarchy to structure large models [Aal08]; *(iv)* and finally combining some of the previous features [Fra17].

Different other models have been proposed in the literature : In [Wan08], the authors introduce the R/NT-WF Net to model workflows constrained by resources and a non-determined time. A procedure is given to compute the earliest and the latest times to start each activity. In [Bou08], a new formalism called the Time Recursive ECATNets (T-RECATNets) is proposed for modelling and analysing time constrained reconfigurable workflows. In [Ber12] component-based timed-arc Petri Nets (CTAPN) are defined to model collaborative healthcare workflows. The authors in [Cic13] consider the TSPN model for modelling and enactment of complex workflows. The authors in [Yeh05], introduce the WFCS-nets (workflow with critical sections nets), to deal with synchronization and time constraints among activities while considering critical sections. In [Aal13] [Sai17], patterns are adapted to cope with inter-organisational workflows (IOWF). An approach describing how the qualitative and quantitative analysis of the framework can be performed by using TCTL model checking is presented in [Bou14].

**-b) Others Models :**
Other formalisms have been also considered to specify workflow systems, as the use of *computational tree logic* ($CTL$) in [Rus91], the *event algebra* in [Els09], multi-agent theory in [Alf16], UML activity diagrams in [Arn16], and *State Charts* [Wen12]. However, all the previous works do not consider time constraints. To deal with the latter, the authors in [Eder15] introduce the Timed Workflow Graph (TWG) defining a graph which is composed of a set of nodes (activities) and edges (control flow). Each activity is characterized by its duration and earliest and latest ending time. However, no delay between activities is defined, thus, leading to incorrect time evaluation.

## 3 Our workflow model

Workflow modelling and analysis are very important aspects and become more complex when considering, in addition, synchronization and time constraints. We present in this section the essence of our approach by describing the different paradigms used in our modelling framework.

### 3.1 Tasks, rendezvous, locality and delay parameters

In our model, we consider a set of elementary work units called tasks, that collectively achieve the whole process. An atomic task is an activity that cannot be divided into sub-processes. Tasks can have different privileges (*Master* or *Slave*) and are associated with a time interval defining the earliest and the latest times within which they should occur. A *Rendezvous* is a time point during the workflow process where a set of tasks has to wait for others in order to perform a synchronization. A rendezvous, noted $R$, is the tuple $(T_m, T_s, (Loc^-, Loc^+), (\alpha^-, \alpha^+, \delta^-, \delta^+))$, such that:
- $T_m$ denotes the non empty set of master tasks involved in the scheme; $T_s$ is the set of slave tasks. In the sequel, we note $T_r = T_s \cup T_m$, and we assume that each task $t$ is characterized by its own time constraints, given in the form of an interval $[x(t), y(t)]$.
- $(Loc^-, Loc^+)$ denotes the localities on which refer the delay parameters. $Loc^-$ is called the *early time locality* and $Loc^+$ is the *latest time locality.*
- The delay parameters $(\alpha^-, \alpha^+, \delta^-, \delta^+)$ are positive rational numbers from $Q^+ \cup \{+\infty\}$ that specify the tolerable drifts between the synchronizing tasks in the rendezvous. The parameters $\alpha^-$and $\delta^-$ are associated with the early time locality, while $\alpha^+$and $\delta^+$ are with the latest time locality.

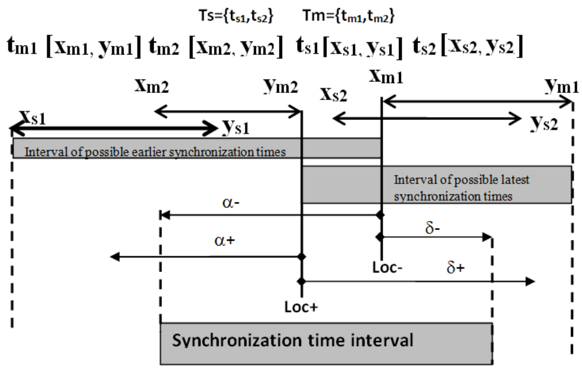


Figure 1: Localities and delays in a rendezvous

According to the type of synchronization pattern, the early time locality $Loc^-$ can refer to one of the two following dates (see Fig.1): $Loc^- := \underset{\forall t \in T_m}{MAX}\{x(t)\}$, namely at the earliest time of the last master task that started its execution; or $Loc^- := \underset{\forall t \in T_r}{MAX}\{x(t)\}$, namely at the earliest time of the last task that started its execution.

Likewise, $Loc^+$ can take two different dates: $Loc^+ := \underset{\forall t \in T_m}{MIN}\{y(t)\}$, namely at the latest time of the first master task that ends its execution; or $Loc^+ := \underset{\forall t \in T_r}{MIN}\{y(t)\}$, namely at the latest time of the first task that ends its execution.
For the sake of simplification, in the sequel the expression of the value of both localities will refer only to the set of transitions considered in the operators $MIN$ or $MAX$, namely $T_m$ or $T_r$ rather than considering the whole expression. However, whatever the synchronization pattern in the Rendezvous we consider, it should be noticed that at least one of the two localities $Loc^-$ and $Loc^+$ of the rendezvous must refer to the set of master tasks $T_m$. This is because the rendezvous must be driven by master tasks.

### 3.2 Timed Rendezvous patterns

Let $R = (T_m, T_s, (Loc^-, Loc^+), (\alpha^-, \alpha^+, \delta^-, \delta^+))$ be a rendezvous. As illustrated in Fig.1, the earliest time to hold the rendezvous must occur:
- no earlier than $\alpha^-$ time units before the occurrence of the locality $Loc^-$; and
- no later than $\delta^-$ time units after the occurrence of the locality $Loc^-$.

The latest time to hold the rendezvous must occur:
- no earlier than $\alpha^+$ time units before the occurrence of the locality $Loc^+$; and
- no later than $\delta^+$ time units after the occurrence of the locality $Loc^+$;

It is noteworthy that the parameters $\alpha^-$ and $\delta^-$ are associated with the locality $Loc^-$, while $\alpha^+$ and $\delta^+$ are associated with the locality $Loc^+$. Let be $MAX$: the maximum value between the earliest time of the last master task that started its execution and the latest time of the first master task that ends its execution. Likewise, $MIN$: the smallest value between the earliest time of the last master task that started its execution and the latest time of the first master task that ended its execution.
According to the value of the delay parameters, the two localities, we define and discuss in the sequel a panel of useful synchronization patterns that are derived from the concept of rendezvous:

**a-Cobegin rendezvous:** This regroups all the patterns such that the earliest time of the rendezvous is determined by the earliest start of one task among masters tasks or all the tasks, leading to different variants, as for example:

-1) if we have ($\alpha^- = 0$) and ($\delta^- = 0$ or $\infty$), then, the earliest time of the synchronization occurs at the earliest time of the last master task that started its execution. We call this pattern the *cobegin master Synchronized* rendezvous (see Fig 2.a)). Example: let's consider the example of the transplantation surgery activity: if the organ arrives between $[1, 3]$, the blood between $[2, 3]$ and the patient between $[2, 5]$, the earliest time the rendezvous occurs is 2.

-2) If we have($\alpha^- = \infty$) and ($\delta^- = 0$ ), then the earliest time of the synchronization occurs at $MIN$ (see Fig 2.b). This is called a *strict cobegin* rendezvous. Assuming the same example, if the organ arrives between $[1, 2]$, the blood at $[3, 3]$ and the patient between $[2, 5]$. The earliest time the rendezvous occurs is the minimum between $(3, 2) = 2$.

-3) If we have($\alpha^- = \infty$) and ($\delta^- = \infty$ ), then the earliest time of the synchronization occurs at:
- if $Loc^- = T_m$ : The earliest time of the first master task that stated its execution. We call this pattern, *cobegin master-relaxed* rendezvous (see Fig 4.g)).
- if $Loc^- = T_s$ : The earliest time of the first task that stated its execution. We call this rendezvous *cobegin-relaxed* rendezvous

**b-Coend related rendezvous:** This regroups all the patterns such that the latest time of the rendezvous is determined by the latest ending time

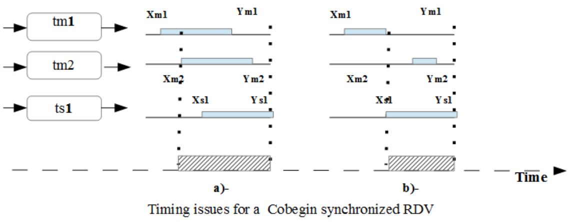


Figure 2: *Cobegin* related Rendezvous

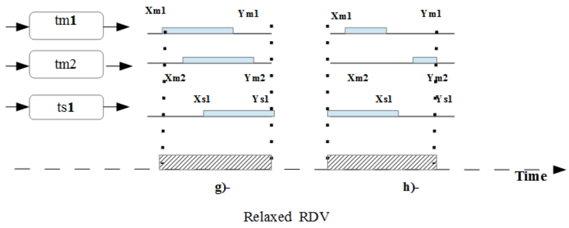


Figure 4: Relaxed rendezvous.

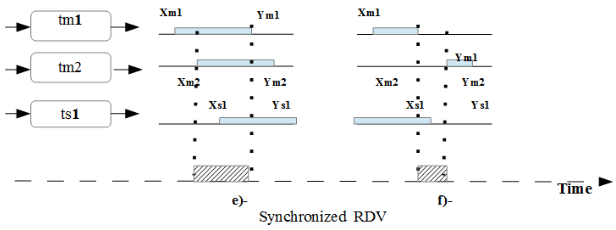


Figure 5: Fully synchronized rendezvous.

of the fist task (among masters tasks or all the tasks); this leads to different variants, for instance:

-1) If we have ($\delta^+ = 0$) and ($\alpha^+ = 0$ or $\infty$), then the latest time of the synchronization occurs at the latest time of the first master task that ends its execution. We call this pattern the *coend master Synchronized* rendezvous (see Fig 3.c). Example: Let's consider the previous example, if the organ arrives between $[1, 3]$, the blood between $[2, 3]$ and the patient between $[2, 5]$, the latest time of the rendezvous is 3.

-2) If we have($\delta^+ = \infty$) and ($\alpha^+ = 0$ or $\infty$), then the latest time of the synchronization occurs at $MAX$ (see Fig 3.d). This is called a *hard Coend* rendezvous. Example: if the organ arrives between $[1, 2]$, the blood at $[3, 3]$ and the patient between $[2, 5]$, the latest time the rendezvous occurs is the maximum between $(3, 2) = 3$.

-3) If we have ($\delta^+ = \infty$) and ($\alpha^+ = \infty$), then the latest time of the synchronization occurs at:
- (if $Loc^+ = T_m$): The latest time of the last master task that ends its execution. We call this pattern the *coend master-relaxed* rendezvous. By taking the previous example, the synchronisation occurs at 5.
- (if $Loc^+ = T_s$): The latest time of the last task that ends its execution. We call this pattern the *coend relaxed* rendezvous.

By combining the previous patterns, or by considering specific conditions, we can define new sub-patterns, as for example:

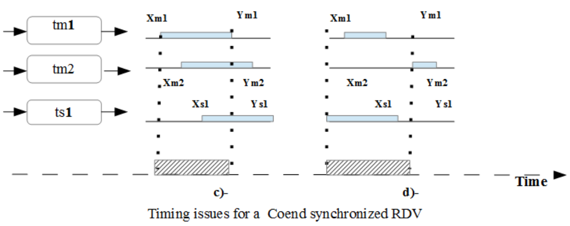


Figure 3: Coend related Rendezvous.

1. A *Fully master synchronized* rendezvous is both a *cobegin master Synchronized* and a *co-end master Synchronized* rendezvous. (see Fig 5.e,f)

2. A *Relaxed* rendezvous is both a *cobegin relaxed* and a *coend relaxed* rendezvous (see Fig 4.g,h).

3. A *Critical* rendezvous, is a rendezvous such that $Loc^+ = Loc^-$ . In other terms the earliest and the latest time of the occurrence of the rendezvous are the same. The rendezvous has to be executed in urgency and cannot be delayed once its offered.

### 3.3 Discussion

A large range of synchronization schemes defined in the literature are covered by our model, and many others. For example for the basic rules of the $TSPN$ model [Cic13], the *Synchronized* rendezvous coincides with either the $And$ or the $Pur - And$ rules (when considering a different variants of disjointed interval relations). This expresses that all tasks are present during the rendezvous. The *Relaxed* rendezvous is referred to the $Or$ rule of $TSPN$, for instance. The *Cobegin Synchronized* rendezvous covers: the $And$, $Pur - And$, $Weak - And$, $And - Master$, rules, whereas, the (Coend Synchronized) rendezvous covers: the $And$, $Pur - And$, $Strong - Or$, $StrongMaster$ rules. The *Cobegin Relaxed* rendezvous covers the : $Or$, $Strong - Or$, $Or - Master$, rules, whereas, the *Coend Relaxed* covers the $Weak - And$, $Or$, $Weak - Master$,rules. Our model covers also, the *wait* constraints of the parallel pattern [Car09]. However, our model is more expressive as it considers time constraints in form of intervals (not instants) and at different levels (not only between tasks), while assuming different privileges. For the best of our knowledge, no related works have addressed the timed synchronization between workflow tasks of different privileges. Synchronization and time constraints are generally studied separately.

# 4 Time Workflow net with Rendezvous: RTWFN

In this section, we first present the syntax and the semantics of the RTPN model as introduced in [Ham17], then we introduce a subclass of RTPN, called Workflow nets with timed rendezvous RTWFN. This model is dedicated to specify workflow systems. Finally, we present a RTWFN based modelling approach for workflow systems.

## 4.1 Time Petri net with Rendezvous: $RTPN$

A classical Petri net (PN) is a directed bipartite graph with two types of nodes, called places and transitions. The nodes are connected via directed arcs and connections between two nodes of the same type are not allowed. Places are represented by circles and transitions by rectangles. We assume that the reader is familiar with Petri nets theory. In the RTPN model, time intervals are associated with each transition thus defining a Time Petri net (TPN). The TPN is then extended with a set of synchronization rules defined by the concept of rendezvous introduced previously. We recall hereafter the syntax and the semantics of the $RTPN$ model. Formally, the syntax of the $RTPN$ model is defined as follows:

**Definition** *An $RTPN$ is given by the tuple $(P, T, B, F, M_0, I_s, RDV_s)$ where: $P$ and $T$ are respectively two non empty disjoint sets of places and transitions; ; $B$ and $F$ are respectively the backward and the forward incidence functions $B : P \times T \longrightarrow N = \{0, 1, 2, ..\}$; $F : P \times T \longrightarrow N$; $M_0$ is the initial marking function that associates with each place a number of tokens $M_0 : P \longrightarrow N$; $I_s$ is the delay interval mapping function; $I_s : T \longrightarrow Q^+ \times Q^+ \cup \{\infty\}$, where $Q^+$ is a set of null or positive rational values. We write $I_s(t) = [x_0(t), y_0(t)]$. This gives the static time interval within which the transition $t$ can fire, such that $0 \leq x_0(t) \leq y_0(t)$;*
*$RDV_s$ denotes a finite set of synchronous Rendezvous. A Rendezvous $R$ of $RDV_s$ is a synchronisation scheme that has the form $(T_m, T_s, (Loc^-, Loc^+), (\alpha^-, \alpha^+, \delta^-, \delta^+))$, where $T_m$ and $T_s$ are subsets of transitions. $T_m \cap T_s = \emptyset$ and we note $T_r = T_m \cup T_s.T_m$ is a non empty and finite set of master transitions, and $T_s$ the finite set of slave transitions. $(Loc^-, Loc^+)$ are the localities considered in the rendezvous, values of which are in $\{(T_m, T_m), (T_m, T_r), (T_r, T_m)\}$. Finally, $\alpha^-, \alpha^+, \delta^+$ and $\delta^-$ are the delay parameters that take their values in $Q^+ \cup \{+\infty\}$.*

In the $RTPN$ model, a transition can be involved in more that one rendezvous. This denotes the case where an event or a process ( modelled by the transition) is subject to different alternative synchronization schemes. The selection of the rendezvous to execute is handled in non deterministic manner. However, a priority function on the set of rendezvous can be introduced as an additional parameter to solve the non determinism. In other respects, a transition that is not involved in any rendezvous of the $RTPN$ is to progress in an asynchronous manner since it is not compelled by any synchronisation scheme. The $RTPN$ model evolves by firing a *rendezvous* at each step. This implies the firing of all its transitions providing that some conditions are satisfied. Firing a rendezvous relies on the marking and on the corresponding synchronization pattern which entails to meet the dynamic time constraints of the model. Different semantics can be considered : a monoserver semantics, namely for any marking only one instance of a transition and by extension a rendezvous can be enabled (see the paper [Ham17] for more details); or a multi-server one which considers that a transition and hence a rendezvous can be enabled more than once for a given marking (refer to the papers [Abd15] [Bouch13]).

## 4.2 Time Workflow Net with Rendezvous: RTWFN

We introduce, in the following, the RTWFN model which is a particular case of a RTPN.

**Definition** *A RTWFN noted $RT_w$ is a tuple $(RT, p_b, p_e)$, such that:*
*-RT = $(P, T, B, F, M_0, I_s, RDV_s$ is a Time Petri Net with rendezvous.*
*-$p_b$ is a special place of $P$ called the beginning place*

*of the workflow net, and we have:* $\bullet p_b = \emptyset$ *and* $M_0(p_b) \neq 0$*;*
*-*$p_e$ *is a special place of* $P$ *called the ending place of the workflow, and we have:* $p_e\bullet = \emptyset$ *and* $M_0(p_e) = 0$ *; where:* $\bullet x$ *denotes the set of input transitions connected to the place* $x$ *while* $x\bullet$*: gives the set of output transitions connected to* $x$ *.*

The place $p_b$ denotes the source of the net while the place $p_e$ the sink of the net. The RTWFN should verify that there exists a run from the initial marking including the place $p_b$ to a final marking including the place $p_e$ ; we say that the net is *strongly connected.*

### 4.3 Modeling Workflow with synchronization and time delay using RTWF-Net

The RTWFN model of the whole workflow constrained by synchronization and time delays can be obtained by the following approach:

-1)First we create the places $p_e$ and $p_b$.

-2) *A single task:* Each elementary unit:(*task*) is mapped into a transition $t \in T$ and an input place $p \in P$. For time constraints a time interval is associated with each transition's task $I(t) = [x(t), y(t)]$, thus, defining the earliest and the latest time delay of the task. If no time constraint are imposed, this means that $I(t) = [0, \infty)$, contrarily, with $I(t) = [0, 0]$ the task cannot be delayed and must occur as soon as the input place is marked (See Fig.6.(x)). If the task is the first in the process then its input place is $p_b$. If it is the last in the workflow then its output place is $p_e$.

-3) *Sequence*: In the example of Fig 6.a, tasks $t_1$ and $t_2$ are executed sequentially, representing precedence constraints of task execution in the workflow;

-4) *Choice*: In Fig 6.b, $t_1$ and $t_2$ are in conflict and can never occur both;

-5) *Concurrency*: In Fig 6.c, tasks are in concurrency; they occur in parallel and are not in conflict. Their execution can be governed by synchronization patterns that are expressed in the form of rendezvous.

### 4.4 Cases Study Examples

In this section, we present two case-studies to highlight how the RTWFN model is suitable to model worflow systems with complex synchronization patterns.

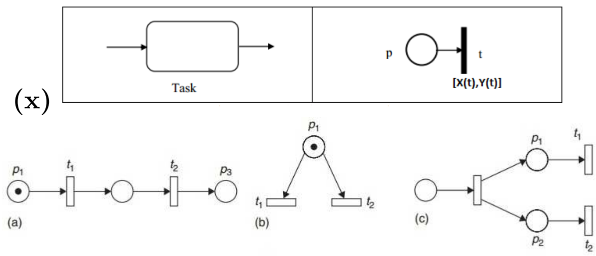


Figure 6: Modelling approach with RTWFN

#### 4.4.1 Example-1

Let's consider the example of "*An online vendor workflow*", already presented in [Bet02]. The figure 7 depicts a portion of the whole $RTWFN$ specification. In the problem description, different types of synchronization and time constraints have to be considered to ensure that all the products are delivered to the customer in a due time to better manage the warehouse resources. $A$ and $B$ correspond to the shipments that have to be made by two suppliers (of the same privilege). The task $A$, resp $B$ is denoted by the transitions $A_b$ and $A_e$(Beginning and and end of $A$) resp. the transitions $B_b$ and $B_e$(Beginning and the end of $B$). Both $A$ and $B$ must occur respectively between $[3, 7]$ and $[1, 3]$ after the the ending of the operation $OP$ that lasts between $[1, 10]$. The tasks $A$ and $B$ have a duration between $[1, 5]$. Finally, the beginning of the local delivery task $LD$ (denoted by the transition $LDb$), has to occur as soon as $A$ and $B$ complete. The final delivery task must begin after that all products are made available and none of the products must wait more than 2 time units at the warehouse. To express the previous synchronization requirements we introduce the following set of rendezvous $RDV = \{R_1, R_2\}$ such that: $R_1 = (\{t_{Ab}, t_{Bb}\}, \emptyset, (\{t_{Ab}, t_{Bb}\}, \{t_{Ab}, t_{Bb}\}), (0, 0, 0, 0))$ which means no temporal delay is considered at the beginning of $A$ and $B$ and represents a *fully synchronized* rendezvous as well as a *Critical rendezvous.* $R_1$ can fire within:
$\left[\underset{\forall t \in \{t_{Ab}, t_{Bb}\}}{MAX}\{x(t)\}, \underset{\forall t \in \{t_{Ab}, t_{Bb}\}}{MIN}\{y(t)\}\right]$ =[3,3].
$R_2 = (\{t_{Ae}, t_{Be}\}, \emptyset, (\{t_{Ae}, t_{Be}\}, \{t_{Ae}, t_{Be}\}), (0, 0, 2, 2))$
Means that the earliest time to start delivery task can be advanced not more than 2 times units, and can be delayed to no more than 2 time units. Furthermore, $R_2$ can fire within $\left[\underset{\forall t \in \{t_{Ae}, t_{Be}\}}{MAX}\{x(t)\}, \underset{\forall t \in \{t_{Ae}, t_{Be}\}}{MIN}\{y(t)\} - 2\right]$=[1,5-2]

=[1,3]. This pattern denotes a *restricted* rendezvous.

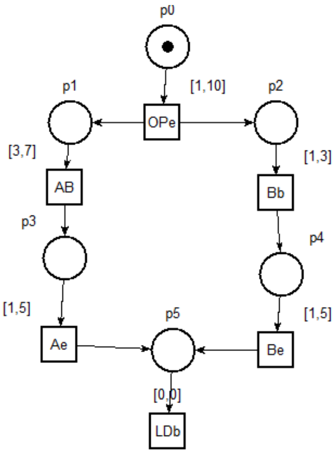


Figure 7: The RTWFN modelling the "on line vendor" workflow

### 4.4.2 Example-2

We consider here the example of *ST-segment Elevation Myocardial Infarction* (STEMI), published by the American College of Cardiology/American Heart Association in 2004 [Car09]. The associated RTWFN is depicted in Fig. 8. The problem is as follows: when a patient comes to the Emergency Department ($E.D.$) (task $T_1$), he can wait between approximatively $[2, 4]$ minutes before being handled. Once he is admitted, the patient is examined first (task $T_2$), which takes between $[5, 20]$ minutes. If the diagnosis is a (STEMI) occurrence ( transition $C1$), then a well-know set of therapy and diagnosis tasks has to be performed. Otherwise, ( transition $C2$) a further patient evaluation has to be done (choice). Since the guideline considers only (STEMI) patients, we have decided to close the flow issued from $C2$ after the task $T_3$. The flow from $C1$ is composed by three parallel sub-flows. The lowest from place $B$) refers to the main therapeutic action in presence of a *myocardial infarction*: *reperfusion* is obtained through a *fibrinolytic* therapy (transition $T_4$ which is a slave task). The flow from place $C$ refers to the complementary therapeutic action consisting of the assumption of *beta blocker drugs* (task $T_5$) (master task). The uppermost flow (from place $A$) contains the possible activities related to therapies for *ischemic discomfort*. If the presence of ischemic discomfort ($C2$) is confirmed (transition $I2$), a *nitroglycerin therapy* is provided

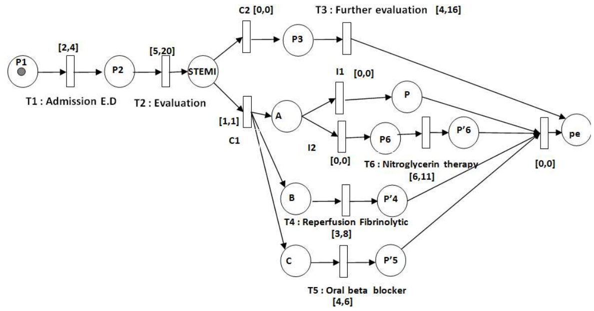


Figure 8: A "healthcare" example

(task $T_6$)(a second slave task). Otherwise ($I1$ no supplementary treatment is provided. After all these therapeutic actions, the workflow ends.

As discussed in [Car09], we want to express the fact that the synchronization of the *reperfusion* ($T_4$) and the *oral therapy* ($T_5$) neither can start more than 2 minutes before nor can start more than 1 minute after the end of the oral therapy ($T_5$). To this aim, We consider the following rendezvous in our model:

$R_1 = (\{T5\}, \{T4\}, (\{T4, T5\}, \{T5\}), (0, 0, 2, 1))$ without a *nitroglycerin therapy*: $T_6$; and $R_1$ can fire within:

$\left[\underset{\forall t \in \{T5,T4\}}{MAX}\{x(t)\}, \underset{\forall t \in \{T5\}}{MIN}\{y(t) - 2\}\right]$ =[4,6-2]=[4,4].

This denotes a *critical cobegin* rendezvous ), or:

$\left[\underset{\forall t \in \{T5,T4\}}{MAX}\{x(t)\}, \underset{\forall t \in \{T5\}}{MIN}\{y(t) + 1\}\right]$ = [4,6+1]=[4,7]

which denotes a *cobegin synchronized* rendezvous.

# 5 Conclusion

In this paper, we have presented a modelling approach for timed workflow systems with complex synchronizations. The approach is based on Petri net formalism, by a subclass of *Time Petri Nets with rendezvous* (RTPN), called *time workflow nets with rendezvous* RTWFN. The latter provides a large panel of concise synchronization patterns that can deal with any complex synchronization scheme in timed workflow systems. Further work will lead us to investigate a methodology for the analysis of the quantitative and the qualitative properties of the RTWFN model.

# References

[Aal05] W.M.P. Van der Aalst, A.H.M. ter Hofstede: YAWL: yet another workflow language. Inf. Syst. 30(4): 245-275, 2005.

[Car09] C. Combi, R. Posenato: Controllability in Temporal Conceptual Workflow Schemata. BPM 64-79, 2009.

[Ham17] A. Hamdani, A. Abdelli: Time Petri Net with Rendezvous. IEEE Codit, Barcelona, Spain 2017.

[Aal98] W.M.P. van der Aalst: The Application of Petri Nets to Workflow Management. Journal of Circuits, Systems, and Computers 8(1): 21-66 1998.

[Ada98] N.R. Adam, V. Atluri, W.K. Huang: Modeling and Analysis of Workflows Using Petri Nets. J. Intell. Inf. Syst. 10(2): 131-158 1998.

[Rus09] N. Russell, A.H.M. ter Hofstede: newYAWL: Towards Workflow 2.0. Trans. Petri Nets and Other Models of Concurrency 2: 79-97 2009.

[Aal08] J.J Bæk and L.K. Bisgaard and W.L.M.P Van der Aalst : From task descriptions via colored Petri nets towards an implementation of a new electronic patient record workflow system. STTT 10(1): 15-28 2008.

[Fra17] F.M. Barreto, S. Julia: Modeling of Video Games Using Workflow Nets and State Graphs. SBES 261-266, 2017.

[Wan08] H. Wang, Q. Zeng: Modeling and Analysis for Workflow Constrained by Resources and Nondetermined Time: An Approach Based on Petri Nets. IEEE Trans. Systems, Man, and Cybernetics, Part A 38(4): 802-817 2008.

[Bou08] K. Barkaoui, H. Boucheneb, A. Hicheur: Modelling and Analysis of Time-Constrained Flexible Workflows with Time Recursive ECATNets. WS-FM 19-36, 2008.

[Ber12] C. Bertolini, Z. Liu, J. Srba: Verification of Timed Healthcare Workflows Using Component Timed-Arc Petri Nets. FHIES 19-36, 2012.

[Cic13] F. Cicirelli, A. Furfaro, L. Nigro: Using time stream Petri nets for workflow modelling analysis and enactment. Simulation 89(1): 68-86, 2013.

[Yeh05] Y. Thabet Kotb, E. Badreddin: Synchronization among Activities in a Workflow Using Extended Workflow Petri Nets. CEC 548-551, 2005.

[Aal13] W.M.P. van der Aalst, M. Weske: Reflections on a Decade of Interorganizational Workflow Research. Seminal Contributions to Information Systems Engineering 307-313, 2013.

[Sai17] S. Boukhedouma, Z. Alimazighi, M. Oussalah: Adaptation and Evolution Frameworks for Service Based Inter-Organizational Workflows. IJEBR 13(2): 28-57, 2017.

[Bou14] H. Boucheneb, K. Barkaoui: Partial order reduction for checking soundness of time workflow nets. Inf. Sci. 282: 261-276, 2014.

[Rus91] M.Rusinkiewicz, A.P. Sheth, G. Karabatis: Specifying Interdatabase Dependencies in a Multidatabase Environment. IEEE Computer 24(12): 46-53, 1991.

[Els09] E.L. Gunter, A. Yasmeen, C.A. Gunter, A.Nguyen: Specifying and Analyzing Workflows for Automated Identification and Data Capture. HICSS 1-11, 2009.

[Alf16] J. Alfonso-Cendón, et al : Implementation of context-aware workflows with multi-agent systems. Neurocomputing 176: 91-97, 2016.

[Arn16] A.Karhof, et al: On the de-facto Standard of Event-driven Process Chains: Reviewing EPC Implementations in Process Modelling Tools. Modellierung 77-92, 2016.

[Wen12] W.W. Zhang, T. Beaubouef, and H. Ye, "State chart: A Visual Language for Workflow Specification," International Journal of Computer Theory and Engineering vol. 4, no. 6, pp. 921-925, 2012.

[Eder15] J. Eder, E. Panagos, M. Rabinovich: Time Constraints in Workflow Systems. Seminal Contributions to Information Systems Engineering 191-205, 2013.

[Abd15] A. Abdelli : Towards a General Model to Handle Multi-enabledness in Time Petri Nets. Formalisms for Reuse and Systems Integration 103-131, 2015.

[Bouch13] H. Boucheneb, D. Lime, O.H. Roux: On Multi-enabledness in Time Petri Nets. Petri Nets 130-149, 2013.

[Bet02] C. Bettini, X.S. Wang, S.Jajodia: Temporal Reasoning in Workflow Systems. Distributed and Parallel Databases 11(3): 269-306, 2002.