Constraint Analysis based on Energetic Reasoning Applied to the Problem of Real Time Scheduling of Workflow Management Systems

Flávio Félix Medeiros[1] and Stéphane Julia[2]
[1]*Institute of Computation, Federal Institute of Mato Grosso do Sul - IFMS, Hilda Streat 203, Naviraí, Brazil*
[2]*Computer Science Department, Federal University of Uberlandia - UFU, Uberlândia, Brazil*

Keywords: Workflow Management System, Petri Nets, Constraint Programming, Energetic Reasoning.

Abstract: The objective of this paper is to propose a constraint analysis applied to the problem of real time scheduling in workflow management systems. The adopted model is a p-time Workflow net with a hybrid resource allocation mechanism. The approach considers time constraint propagation techniques for the different types of routings that exist in workflow processes. Different types of resources, discrete and continuous, are then incorporated into the model and an approach based on energetic reasoning is applied. Energetic reasoning can identify unacceptable schedulings due to the energetic inability of the involved resources in the implementation of the related activities. An update of the temporal constraints is then produced in order to eliminate the dates inconsistent with the set of scheduling solutions. Considering the set of modified constraints, a specialized inference mechanism called token player is then applied, which has the purpose of obtain in real time an admissible scenario corresponding to a specific sequence of activities that respects the time constraints.

1 INTRODUCTION

The purpose of a workflow management system (Aalst and Hee, 2004) is to run workflow processes. A workflow process represents a sequence of activities of an organization that must execute a specific case in order to achieve a specific goal (treat a specific case). Over recent years, Business Process Management has become important in order to raise service quality and a company's performance (Hofstede et al., 2010).

According to (Aalst and Hee, 2004), the use of Petri nets for the modeling of workflow processes has many advantages, as in the fact of forcing a precise process definition in enterprise systems.

Some models based on Petri nets were defined exclusively for workflow representation as the workflow Nets defined by Aalst (Aalst and Hee, 2004). The workflow Net has only a *Start* place and only one *End* place. A token in a *Start* place represents a case that needs to be handled and a token in a *End* place represents a case that has been handled. Every task is associated with a transition and every condition is associated with a place. In addition, every task and condition must be on a path between the *Start* place and the *End* place.

The workflow Nets are also used as an abstraction of the workflow process that is used to check the soundness property. This property guarantees the absence of deadlocks and other anomalies that can be detected without domain knowledge (Aalst and Hee, 2004).

The scheduling problem (Lee and DiCesare, 1994) aims to organize in time a sequence of activities, taking into consideration some time constraints (timeslots) and restrictions shared resources used in the implementation of the activities. Considering the real-time system case, many scenarios (many cases in a workflow management system) can be run simultaneously and conflict for shared resources, that must be solved in real time (without backtrack mechanisms), may occur if any non-preemptive resource is called at the same time to execute activities that belong to different scenarios.

The fundamental difference between the traditional scheduling problem of production systems and the scheduling problem in workflow management systems is the nature of the resources involved in the execution of activities. For production systems, the resources represent physical equipment and are represented by simple tokens (discrete type resources) in the places. For workflow management systems (WFMS), resources can represent physical equipment as well as human resources. For example, we can allocate a nurse in a hospital to take care of several pa-

Medeiros, F. and Julia, S.
Constraint Analysis based on Energetic Reasoning Applied to the Problem of Real Time Scheduling of Workflow Management Systems.
DOI: 10.5220/0006275903730380
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 3*, pages 373-380
ISBN: 978-989-758-249-3

tients at the same time during her working day. In this situation, a nurse cannot be seen as a single discrete token. As a direct consequence, a model based on an ordinary Petri net is not appropriate to represent all the features that exist in workflow management systems.

Another problem is the uncertainty associated with the behavior of human resources. This problem makes it impossible to compute a predictable scheduling solution for processes commonly found in manufacturing systems (Lee and DiCesare, 1994).

In this paper, a method of analysis under constraints that aims to building a constraint knowledge base system that can be integrated into a real time scheduling strategy, is presented. In particular, the main approach will consider a kind of energetic reasoning applied on resource constraints that exist in workflow management systems.

2 TIME WORKFLOW NET WITH HYBRID RESOURCE ALLOCATION MECHANISM

The model proposed in this paper is the same model adopted in (de Oliveira et al., 2008). It is a p-time Petri net with hybrid resource allocation mechanisms. The control structure follows the same as that used in a classic workflow Net (Aalst and Hee, 2004), with a Start place, a End place and with the main routings that exist in workflow processes. The main definitions and concepts related to the workkflow Net are presented following in sections 2.1 and 2.2, respectively.

2.1 Workfow Nets

- It has only one source place, named *Start* and only one sink place, named *End*. These are special places such that the *Start* place has only outgoing arcs and the *End* place has only incoming arcs.
- A token in *Start* represents a case that needs to be handled and a token in *End* represents a case that has been handled.
- Every task t (transition) and condition p (place) should be on a path from *Start* place to *End* place.

The formal definition of workflow Net is presented in the following.

Definition 1 (Workflow Net). A Petri net $PN = \{P,T,F\}$ is a workflow Net if, and only if,

1. There is one source place $i \in P$ such that $\bullet i = \phi$,
2. There is one sink place $o \in P$ such that $o\bullet = \phi$,
3. Every node $x \in P \cup T$ is on a path from i to o.

A workflow Net has one input place (i) and one output place (o) due to the fact that any case handled by the process represented by the workflow Net is created when it enters the WFMS and is deleted once it is completely handled by the WFMS, i.e., the workflow Net specifies the life-cycle of a case. Finally, the third requirement in Definition 1 has been added to avoid "dangling tasks and / or conditions", i.e., tasks and contitions that do not contribute to the processing of cases (Aalst and Hee, 2004).

As previously mentioned, a workflow Net has a soundness property. This property is a correctness criterion related to its dynamic behavior.

2.2 Soundness

A workflow Net is Sound if, and only if, the following requirementes are satisfied (Aalst and Hee, 2004) (Soares Passos and Julia, 2013).

- For each token put in the *Start* place, one and only one token appears in *End* place.
- When the token appears in *End* place, all the other places are empty for this case.
- For each transition (task), it is possible to move from the initial state to a state in which that transition is enabled, i.e. there are no dead transitions.

The formal definition of soundness is presented in the following:

Definition 2 (Soundness). A process modelled by a workflow Net $PN = \{P,T,F\}$ is sound if, and only if:

1. For every state M reachable from state i, there exists a firing sequence leading from state M to state o.
   Formally,
   $$\forall_M (i \xrightarrow{*} M) \Rightarrow (M \xrightarrow{*} o)$$
2. State o is the only state reachable from state i with at least one token in place o. Formally,
   $$\forall_M (i \xrightarrow{*} M \wedge M \geq o) \Rightarrow (M = o)$$
3. There are no dead transitions in (PN,i). Formally,
   $$\forall_{t \in T} \exists_{M,M'} \, i \xrightarrow{*} M \xrightarrow{*} M'$$

An activity can be associated to a transition in a workflow Net. However, as presentend in (Leiliane et al., 2016), in order to explicitly indicate the beginning and the end of each activity in execution, two sequencial transitions plus a place to model an activiy is used. The first transition represents the beginning of the activity, the place represents the activity, and the second transition represents the end of the activity (Wang and Rosca, 2009) .

In this work, instead of associating tasks to transitions, tasks are associated with specific places. Thus, the resource allocation mechanisms can be easily viewed on tasks.

In the following the main features of this model are presented.

2.3 Routing Constraints

The *handle complaints process* presented in (Aalst and Hee, 2004) will be used to illustrate the various types of routings studied in this work. In this process, a complaint needs to be initially registered. Then the customer, who made the complaint, and the department affected by the complaint are contacted. The customer is contacted in order to obtain more information. The department is informed about the complaint and it is asked to take an initial action. These two activities should be executed in parallel (parallel routing). After that, the information is collected, and in sequence, a decision must be taken (sequential routing). Depending on the decision that was taken, a payment is made or a letter is sent to the customer (alternative routing). Finally the complaint is filed.

Figure 1 illustrates the different types of routings on the *handle complaints process*. The places E_j with j=0 to 10 represent waiting places between activities. The activities are associated with the places A_i with i=1 to 8. In particular, the places *E0* and *E10* represent the Start and the End of the process. The token in *E0* represent a case to be handled.

2.4 Time Constraints

Usually, the time required to execute an activity in a workflow process is non-deterministic. According to (de Oliveira et al., 2008), explicit time constraints existing in systems with real-time characteristics can be formally specified using a p-time Petri net model, corresponding to a static time interval associated to each place of the model.

The dynamic behavior of p-time Petri net depends on the marking of the network and also of the tokens temporal situation that is given by the visibility interval (de Oliveira et al., 2008). A visibility interval [(δ_p)min, (δ_p)max] associated with a token in a place *p* of a p-time Petri net specified the minimum date ((δ_p)min) at which a token is available in *p* to trigger an output transition of *p* (earliest start date of an activity), and the maximum date after which the token becomes unavailable (*dead*) and cannot be used to trigger any transition (latest start date of the corresponding activity).

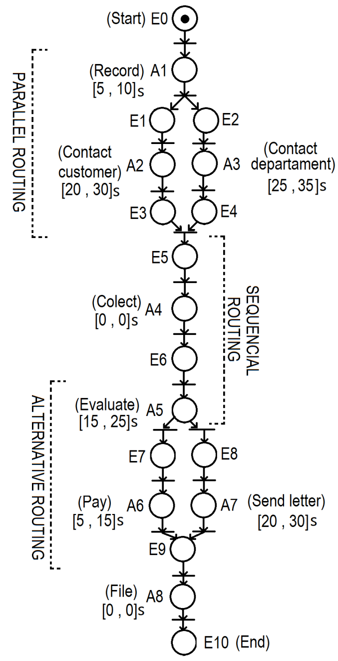

Figure 1: Handle complaints process.

2.4.1 Static Interval

The static definition of a p-time Petri net is based on static intervals that represent the permanency interval of the tokens in the places from the point of view of the activities duration. The static definition of a p-time Petri net can be shown in (de Oliveira et al., 2008).

Figure 1 shows static intervals related to the activities of the *handle complaints process*. For each activity A_i, there exists a static interval that specifies its minimum and maximum duration. In particular, the static intervals associated with the tasks *collect* in *A4* and *file* in *A8* are *[0, 0]s* because their durations hold no valeu when compared to other tasks of the *handle complaints process*.

2.4.2 Visibility Interval

According to (de Oliveira et al., 2008) the dynamic behavior of p-time Petri net depends on the marking of the net and on the time situation of the tokens, which is given by the visibility intervals whose definition is the following one:

Definition 3 (Visibilit Interval of a p-time Petri Net). A visibility interval $[(\delta_p)_{mim}, (\delta_p)_{max}]$

associated with a token in a place p of a p-time Petri net defines:

- the earliest date $(\delta)_{min}$ when the token in p becomes available for the firing of an output transition of p;
- the latest date $(\delta)_{max}$ after which the token becomes non-available (dead) and cannot be used for the firing of any transition.

2.5 Resource Constraints

Some of the resources used in workflow management systems can be considered of discrete types and can be represented by simple tokens. It is generally the resources which represent physical equipment such as a printer, for example. Other resources can be represented by continuous resources. It is generally the case of human type resources. These resources are represented by a real number that shows the availability of a human resource.

According to (David, 2010) a discrete resource allocation mechanism and a continuos resource mechanism can be defined as following:

Definition 4 (Discrete Resource Allocation Mechanism). A discrete resource mechanism can be defined by the marked ordinary Petri net model (David, 2010)

$C_{DR} = \langle A_{DR}, T_{DR}, Pre_{DR}, Pos_{DR}, M_{DR} \rangle$ with:

- $A_{DR} = \bigcup_{\alpha=1}^{N_{DR}} A_\alpha \cup \{R_D\}$ where R_D represents the discrete resource place, A_α an activity place and N_{DR} the number of activities which are connected to the discrete resource place R_D.
- $T_{DR} = \bigcup_{\alpha=1}^{N_{DR}} T_{in_\alpha} \cup \bigcup_{\alpha=1}^{N_{DR}} T_{out_\alpha}$ where T_{in_α} represents the input transtion of the activity A_α and T_{out_α} represents the output transtion of the activity A_α.
- Pre_{DR} : $A_{DR} \times T_{DR} \rightarrow \{0,1\}$ the input incidence application, such as $Pre_{DR}(R_D, T_{in_\alpha}) = 1$ and $Pre_{DR}(A_\alpha, T_{out_\alpha}) = 1$ (other combinations of place/transition are equal to zero).
- Pos_{DR} : $A_{DR} \times T_{DR} \rightarrow \{0,1\}$ the output incidence application, such as $Pos_{DR}(R_D, T_{out_\alpha}) = 1$ and $Pos_{DR}(A_\alpha, T_{in_\alpha}) = 1$ (other combinations of place/transition are equal to zero).
- M_{DR} : $R_D \rightarrow R$ the initial marking application, such as $M_{DR}(R_D) = m_D$ the number os discrete resources of the same type.

Definition 5 (Continuous Resource Allocation Mechanism). A continuous allocation mechanism can be defined by the marked hybrid Petri net model (David, 2010)

$C_CCR = \langle A_{CR}, T_{CR}, Pre_{CR}, Pos_{CR}, M_{CR} \rangle$ whith:

- $A_{CR} = \bigcup_{\alpha=1}^{N_{CR}} A_\alpha \cup \{R_C\}$ where R_C represents the continuous resource place, A_α an activity place and N_{CR} the number of activities which are connected to the continuous resource place R_C.
- $T_{CR} = \bigcup_{\alpha=1}^{N_{CR}} T_{in_\alpha} \cup \bigcup_{\alpha=1}^{N_{CR}} T_{out_\alpha}$ where T_{in_α} represents the discret input transtion of the activity A_α and T_{out_α} represents the discrete output transtion of the activity A_α.
- Pre_{CR} : $A_{CR} \times T_{CR} \rightarrow R^+$ the input incidence application, such as $Pre_{CR}(R_C, T_{in_\alpha}) = X_\alpha$ and X_α *in* R^+ and $Pre_{CR}(A_\alpha, T_{out_\alpha}) = 1$ (other combinations of place/transition are equal to zero).
- Pos_{CR} : $A_{CR} \times T_{CR} \rightarrow R^+$ the output incidence application, such as $Pos_{CR}(R_C, T_{out_\alpha}) = X_\alpha$ and $Pos_{CR}(A_\alpha, T_{in_\alpha}) = 1$ (other combinations of place/transition are equal to zero).
- M_{CR} : $R_C \rightarrow R^+$ the initial marking application, such as $M_{CR}(R_C) = m_C$ the availability (in percentage) of the continuous resource.

3 CONSTRAINT PROPAGATION TECHNIQUE

3.1 Time Constraint Propagation on Routings

The time constraint propagation technique on routings proposed in this paper was originally presented in (de Oliveira et al., 2008). The visibility intervals associated with the waiting places E_j between the activities of the workflow Net are calculated using classica techniques of constraint propagation based on graphs without circuits (de Oliveira et al., 2008), and setting the beginning date of a case as well as its maximum duration to be completed.

Figure 2 illustrates the application of the time constraint propagation mechanism over the routes of the handle complaints process, considering four cases to be treated.

In figure 2 (A), the minimum borders of visibility intervals associated with the place *E0* represent the start date of four cases to be treated by the workflow process. The maximum borders of visibility intervals associated with the place *E10* represent the expected end dates of the four cases, knowing that the maximum duration to treat each case is *105* units of time. Since the minimum and maximum durations of

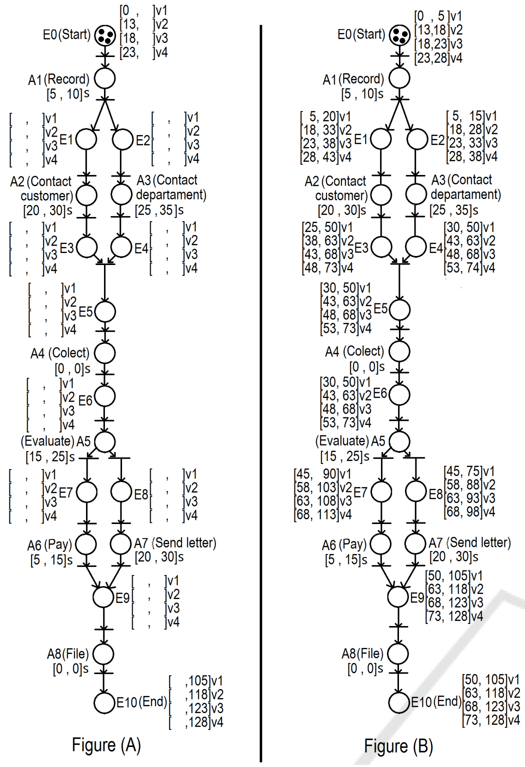

Figure 2: Example of time constraint propagation mechanism on routings.

each activity (static intervals) are known as well as the workflow process routings, we can apply constraints propagation mechanisms to obtain the waiting times between the activities (visibility intervals associated with waiting places *Ei*. Figure (2 (B)) shows the resulting visibility intervals after applying the constraint propagation techniques for each one of the four cases. For example, the visibility interval *V1* associated with the place *E7* means that the activity *pay* associated with *case 1* can be started between the earliest date *45* and the latest date *90*.

3.2 Energetic Reasoning for Resource Constraints

The time constraint propagation mechanism that considers the resources availability takes into account an approach based on a kind of energetic reasoning proposed initially by *Artigues and Lopez* (Artigues and Lopez, 2015). Such an approach was used to treat the problem of task scheduling in manufacturing systems using a model based on graphs with activities on arcs. Energetic reasoning shows the relationship that exists between the energy provided by the resources and the energy consumed by the activities.

Considering the context of workflow management systems, each activity i can be characterized by its minimum ($Pmin_i$) and maximum ($Pmax_i$) durations and must be executed within a time window [r_i, d_i] with r_i representing the earliest date of the activity and d_i representing the latest delivery date of the activity.

3.2.1 Global Consistency Study

According to (Artigues and Lopez, 2015) the maximum available energy that a resource can provide in a given time window Δ =[F_Δ, S_Δ] is defined as: $W^\Delta = A.(F_\Delta - S_\Delta)$ where A is an energy intensity value that the resource in question provides to execute the activities that use it.

Considering an activity i, that starts at date S_i and finishing at date F_i, considering also the energy intensity A_i that it requires from the resource and a given time window Δ, two situations can occur to define the energy required by i in Δ:

If $([S_i, F_i] \cap \Delta) \neq \emptyset$ then

$W_i^\Delta = a_i.[min(F_i, F_\Delta) - max(S_i, S_\Delta)]$.

If $([S_i, F_i] \cap \Delta) = \emptyset$ then $W_i^\Delta = 0$, where W_i^Δ is the energy required by i in Δ. More generally, we can say that

$W_i^\Delta = a_i.max[0, min(F_i, F_\Delta) - max(S_i, S_\Delta)]$.

Since S_i (earliest date of activity i) is a variable, W_i^Δ is also a variable, therefore, we can derive some minimum values taking into account the time interval $[r_i, d_i]$.

The minimum energy required by an activity i over the time window Δ, which is called $\underline{W_i^\Delta}$, is obtained by the position of the activity i that least overlaps the time window Δ: $\underline{W_i^\Delta} = a_i.max[0, min(Pmin_i, F_\Delta - S_\Delta, r_i + Pmin_i - S_\Delta, F_\Delta - d_i + Pmin_i)]$.

Considering W^Δ, an energy provided by a resource on a given time window Δ, and $\sum_{i=1}^{n} W_i^\Delta$, the energy required by the activities that will use it in the same time window Δ, we can derive a global consistency condition which must be respected by any scheduling at any interval Δ,

$\forall\Delta, \sum_{i=1}^{n} W_i^\Delta \leq W^\Delta$ (1)

The formula in *(1)* means that the sum of the energy spent for the activities that use the same resource in a time interval Δ has to be smaller than the total energy produced by the resource. In particular, from condition *(1)*, the following proposition can be drawn.

If $\exists\Delta$ such that $\sum_{i=1}^{n} \underline{W_i^\Delta} > W^\Delta$, then there exists no admissible scheduling.

Figure 3 illustrates a conflict situation for the shared resource *R2* (Complaints employee) of the *handle complaints process* illustrated in Figure 5(A). This resource will be involved in the execution of

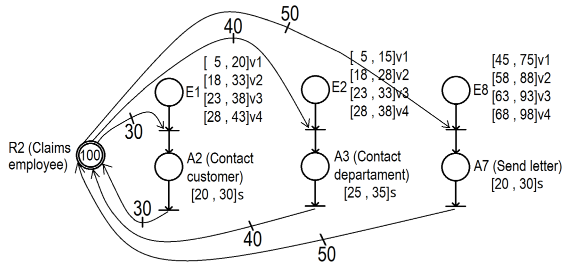

Figure 3: Continuous resource allocation mechanism.

twelve different activities. Table 1 presents these activities with their respective time constraints $[r_i, d_i]$, their minimum durations ($Pmin_i$) and their energy intensities (a_i). The time restriction r_i is given by the minimum bounds of visibility intervals, $Pmin_i$ by the minimum bounds of static intervals, d_i by the sum of minimum bounds of visibility intervals and the maximum bounds of the static intervals, and a_i by the weights associated to the archs of the continuous resource *R2*.

Table 1: Data of the activities involved with resource R2.

i	r_i	$Pmin_i$	d_i	a_i
A2 Case 1	5	20	50	30
A2 Case 2	18	20	63	30
A2 Case 3	23	20	68	30
A2 Case 4	28	20	73	30
A3 Case 1	5	25	50	40
A3 Case 2	18	25	63	40
A3 Case 3	23	25	68	40
A3 Case 4	28	25	73	40
A7 Case 1	45	20	105	40
A7 Case 2	58	20	118	40
A7 Case 3	63	20	123	40
A7 Case 4	68	20	128	40

For example, considering the activity *A3*, related to *Case 4*, an interval Δ that covers the entire execution window $[S_\Delta, F_\Delta]$ = $[r_i, d_i]$ must be chosen. In this case, Δ=$[S_\Delta = 28, F_\Delta = 73]$. Adding up the minimum energy needed for the complete execution of each of the twelve activities, within the considered Δ, we obtain $\sum_{i=1}^{n} \underline{W}_i^\Delta = 3830$. This value is less than the energy that the resource *R2* can provide during this studied interval (W_Δ=4500). This ensures that in the interval Δ, the resource *R2* can provide the minimum energy required to execute the *12* activities completely.

In the example of the *handle complaints process*, it can be verified that for the *12* activities using the resource *R2*, the global consistency condition is satisfied. A situation where condition (1) is not respected would imply in the impossibility of finding an acceptable scheduling without relaxing some of the time constraints.

Energetic reasoning also takes into account a set *S*. This set denotes constraint variables that represent the starting dates of the activities consistent with [r_i, d_i] and with the resource restrictions involved in the execution of the activities. The global consistency condition does not allow for the direct updating of the variables in set *S*. For this, a local consistency condition can be considered.

3.2.2 Local Consistency Study

When considering an activity *i*, the global consistency condition (*1*) can be rewritten as follows,

$\forall i, \forall\Delta, W_i^\Delta \leq W^\Delta - \sum_{j=1, j\neq i}^{n} W_j^\Delta$

Considering the maximum value of the right side of the condition above, we can derive the maximum value for its left side.

The term $A_i^\Delta = W^\Delta - \sum_{j=1, j\neq i}^{n} \underline{W}_j^\Delta$ is called the maximum energy available left by other activities in the interval Δ, for the execution of the activity *i*. Any starting date S_i of the activity *i*, that leads to a higher energy consumption than the maximum energy available to it, must be modified.

The constraint propagation mechanisms that are of interest in this work aim to remove inconsistent values of certain activities in order to reduce the set of possible solutions.

As previously mentioned, the set *S* denotes constraint variables, which represent the starting dates of the activities consistent with [r_i, d_i] and with the resource restrictions involved in the execution of the activities. At this point the formal definition of the set *S* can be presented as follows.

Definition 6 (Constraint Variables of the Set *S*). A starting date S_i of an activity *i* will be part of the set *S* if, and only if the following three conditions are satisfied,

1. $S_i \in [r_i, d_i]$,
2. $\forall\Delta, \sum_{i=1}^{n} W_i^\Delta \leq W^\Delta$,
3. $\forall i, \forall\Delta, W_i^\Delta \leq W^\Delta - \sum_{j=1, j\neq i}^{n} W_j^\Delta$.

According to (Artigues and Lopez, 2015), a way to locally verify that the resource used by a particular activity provides sufficient energy for its full completion taking into account the value of A_i^Δ (maximum energy available for the activity *i*) is to perform the calculation of the maximum equivalent duration $\overline{p}_i^\Delta$. This duration represents the maximum duration of the resource availability to execute the activity i respecting the minimum energy expenditure of others activities that use the same resource in the interval Δ. Thus $\overline{p}_i^{\Delta}$ is given by the ratio between A_i^{Δ} and the energy intensity that i requires from the resource: $\overline{p}_i^{\Delta} = \frac{A_i^{\Delta}}{a_i}$.

For example, considering the activity *A3* of *case 4* on figure 3 it can be verified that by adopting an interval Δ, which covers exactly the minimum duration of the activity execution, $\Delta = [S_{\Delta} = ((\delta_p)min), F_{\Delta} = ((\delta_p)min) + Pmin_i]$, that is $[S_{\Delta} = 28, F_{\Delta} = 28 + 25 = 53]$ (figure 4) and considering the maximum energy available A_i^{Δ}=970 for this activity in this interval Δ, the obtained value of $\overline{p}_i^{\Delta}$ =24.25 is less than the value of $Pmin_i$ =25. In this particular case, the value of $\overline{p}_i^{\Delta}$ =24.25 means that the resource *R2* can supply energy to the activity *A3* on *case 4* during 24.25 time units. Since the minimum duration of the activity corresponds to 25 time units, there exists a local inconsistency as the resource duration availability is less than the minimum necessary to execute the corresponding activity. In order to find a time range within the time interval $[r_i, d_i]$ where the duration $\overline{p}_i^{\Delta}$ is greater or equal to $Pmin_i$, a displacement of the interval Δ on the right can be applied.

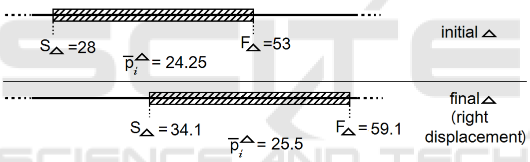

Figure 4: Displacement of the delta time interval.

The interval Δ, after the displacement on the right, in accordance to the subtraction between ($Pmin_i$) and ($\overline{p}_i^{\Delta}$) (25 - 24.25 $\approx$ 0.8), is equal to $[S_{\Delta} = 28 + 0.8, F_{\Delta}$=53+0.8] = [$S_{\Delta}$=28.8 ,$F_{\Delta}$=53.8]. By performing the calculation of the maximum equivalent duration, taking into account this new interval Δ, we obtain $\overline{p}_i^{\Delta}$=24.4. Since the maximum equivalent duration is still smaller than the minimum duration of the activity *A3* associated to *case 4* ($Pmin_i$=25), there still exists a local inconsistency corresponding to the earliest starting date of this activity. Thus, it is necessary to continue producing a right displacement of Δ until a local consistency condition is verified.

By moving the interval Δ for nine times on the right, a new inteval Δ =[S_{Δ}=34.1 ,F_{Δ}=59.1] is produced (figure 4). By performing the calculation of the maximum equivalent duration, taking into account this final interval Δ, we obtain $\overline{p}_i^{\Delta}$=25.5, with $\overline{p}_i^{\Delta} > Pmin_i$. Then, the earliest starting date of the activity *A3* associated to *case 4* becomes 34.1 $\approx$ 34, as shown in figure 5 (B).

In fact, if this activity starts between the dates 28

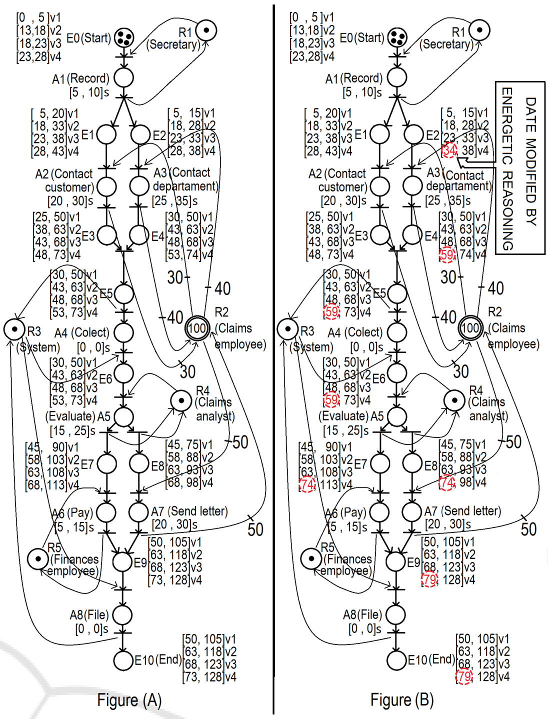

Figure 5: Application of the energy reasoning approach on the *handle complaints process*.

and 34 (visibility interval obtained after applying the time constraint propagation technique on the routings of the process and before applying energetic reasoning on the resource involved in the activity execution), we are sure that an inconsistency will happen and as a consequence a schedule that respects all the time constraints will not be found.

The moving of the latest starting dates of any activity of the workflow process can be performed in a similiar way considering left displacements of maximum bounds of Δ intervals.

After the upgrade of the visibility interval *V4* in *E2*, related to activity *A3* associated to *case 4* (figure 5 (B)), the same propagation technique must be applied to the other activities of the *handle complaints process*. The final result is show in Figure 5(B).

4 GENERAL PRINCIPLE FOR REAL TIME SCHEDULING

A possible technique for allowing the dynamic execution of a p-time Petri nets (real time scheduling) is the p-time Petri net token player algorithm presentend in (de Oliveira et al., 2008). In particular, when the model of the workflow process is based on a p-time Petri net model, the algorithm of the token player must allows to solve in real time conflicts for shared resources.

The main purpose of the scheduling technique is then to find a sequence of activities, from *Start* place to *End* place, which respect the set of time constraints given by the visibility intervals associated to each activity of the process.

The token player has a calendar of events (minimum and maximum bounds of the visibility intervals for each case) scheduled over time. Each time a minimum bound is reached, a token becomes available. If this token enables a transition and if there is no actual conflict for a shared resource, the transition is fired; otherwise the conflict is isolated and a decision-making mechanism is activated to verify if the transition can be fired at the minimum date indicated on its visibility interval. If a transition is fired, a new marking is produced and new visibility intervals are computed; otherwise, the next event of the calendar is treated. The decision-making mechanism is based on the generation of class graphs and was presented in (de Oliveira et al., 2008). If a maximum date of a visibility interval is reached, then the death of a token happens and a time constraint is violated. In this case, there is no guarantee that deadliness of the process will be respected.

5 CONCLUSIONS

The major contribution of this work concerns the improvement of a knowledge base system that defines a set of time constraints that a token player must respect during the execution of a real-time scheduling technique. The main objective is to produce for each case of a workflow Net a sequence of activities that respect a set of temporal constraints, given by a set of visibility intervals (date intervals that fix the starting dates of the activities). In particular, the filtering technique based on a kind of energetic reasoning and on global and local conditions ensures that the existing inconsistent time windows will be removed and reallocated in different dates in order to create favorable conditions for the existence of an admissible scheduling solution. Applying a kind of intelligent token player algorithm with decision making mechanism to the new knowledge base obtained after applying the energetic reasoning, the quality of the produced scheduling solutions (in particular the respect of deadlines associated to the cases of the process) will then be increased, in particular if compared with basic strategies of the type FIFO (First in First Out), which are generally the ones applied to this kind of business systems.

A prototype of the presented approach was implemented in Visual Prolog programming language.

As a future work proposal, a real time scheduling strategy based on a kind of token player similiar to the one presented in (Freitas et al., 2016) will be implemented and validated on the Colored Petri net modeling and simulation tool known as CPN Tolls (Jensen et al., 2007)

REFERENCES

Aalst, W. V. D. and Hee, K. V. (2004). *Workflow Management: Models, Methods and Systems*. MIT Press.

Artigues, C. and Lopez, P. (2015). Energetic reasoning for energy-constrained scheduling with a continuous resource. *Journal of scheduling*, 18(3):225 – 241.

David, R., A. H. (2010). *Discrete, Continuous, and Hybrid Petri Nets. 2nd edn.* Springer Publishing Company, Incoporated.

de Oliveira, F. F., Julia, S., and Valette, R. (2008). Real time scheduling of workflow management systems based on a p-time petri net model with hybrid resources. *Simulation Modelling Practice and Theory 16*, pages 462–482.

Freitas, J. C. J., Julia, S., and Rezende, L. P. (2016). Modeling a fuzzy resource allocation mechanism based on workflow nets. *18th International Conference on Enterprise Information Systems*, pages 559 – 566.

Hofstede, A., Aalst, W., Adams, M., and Russell, N. (2010). *Modern Business Process Automation: YAWL and its Support Environment*. Springer-Verlag, Berlin.

Jensen, K., Kristensen, M., and Wells, L. (2007). Coloured petri nets and cpn tools for modelling and validation of concurrent modeliling. *International Journal on Software Tools for Technology Transfer*, pages 213 – 254.

Lee, D. and DiCesare, F. (1994). Scheduling flexible manufacturing systems using petri nets and heuristic search. In *IEEE transactions on Robotics and Automation*, pages 123–132.

Leiliane, P. R., Julia, S., and Cardoso, J. (2016). Uncertain marking for dealing with partial parallelization in business processes. *In Proceedings of the 18th International Conference on Enterprise Information Systems*, pages 118–125.

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

Wang, J., T. W. and Rosca, D. (2009). Emergency response workflow resource requirements modeling and analysis. *IEEE Transactions on SMC, Part C, 39:270-283.*