Possibilistic WorkFlow Nets for Dealing with Cancellation Regions in Business Processes

Leiliane Pereira de Rezende[1,2], Stéphane Julia[1] and Janette Cardoso[2]
[1]Computing Faculty, Federal University of Uberlândia - UFU, Uberlândia - MG, Brazil
[2]DMIA Department, Institut Supérieur de l'Aéronautique et de l'Espace - ISAE, Toulouse, France

Keywords: WorkFlow Net, Possibilistic Petri Net, Cancellation Features, Reset Arcs, Process Monitoring.

Abstract: In this paper, an approach based on WorkFlow nets and possibilistic Petri nets is proposed for dealing with the cancellation features in business processes. Routing patterns existing in business processes are modeled by WorkFlow nets. Possibilistic Petri nets with uncertainty in the marking and the transition firing are used to deal with all possible markings when cancellation behaviour is considered. Combining both formalisms, a kind of possibilistic WorkFlow net is obtained. An example of a simplified version of a credit card application process is presented.

1 INTRODUCTION

Organizations are increasingly using Workflow Management Systems (WFMS) to reduce costs and improve the performance and efficiency of important business processes (BP). BP represent the sequences of activities that have to be executed within an organization to treat specific cases and to reach well defined goals (van der Aalst and van Hee, 2004). In the field of systems engineering, modeling plays a key role for understanding and controlling the behavior of the corresponding systems. WFMS are used for the modeling, analysis, enactment, and coordination of structured BP by groups of people (van der Aalst, 2000). A workflow process corresponds to the automation of a business process, in whole or part, during which documents, information or activities are passed from one participant to another for a particular form of action, according to a set of procedural rules (Members, 1994).

Many researchers, such as (van der Aalst, 1998) (van der Aalst and van Hee, 2004) (Soares Passos and Julia, 2009), have indicated the Petri net theory as an efficient tool for the modeling and analysis of WFMS's. In particular, an acyclic Petri net model used to represent a BP, the so-called WorkFlow nets (WF-nets), was proposed in (van der Aalst and van Hee, 2004). When considering the WF-nets, the main property that has to be proven to guarantee the correctness notion of the process model is the soundness property (van der Aalst and van Hee, 2004).

An important concept in WFMS is cancellation when the execution of some activities may lead to the termination of other activities in certain circumstances (van der Aalst et al., 2003). Cancellation can be triggered by either a customer request (e.g., a customer wishes to cancel a purchase) or by exceptions (e.g., an order cannot be processed due to insufficient information). In general, cancellation results in one of two outcomes: disabling some scheduled activities or stopping currently running activities (Wynn et al., 2009).

In order to model cancellation in a process model, reset nets were proposed. They are Petri nets extended with special arcs (reset arcs) that can clear the tokens in selected places, i.e., everything is removed from these places for a particular instance (Dufourd et al., 1998). Reset nets have a natural application in business process modeling, where possible cancellation of activities needs to be modeled explicitly, and in WFMS where such process models with cancellation behaviors have to be enacted correctly (Wynn et al., 2009).

The reset nets are apparently an innocent extension of Petri nets but they have rather dramatic consequences. Simple questions such as reachability become undecidable for reset nets with more than two reset arcs (Araki and Kasami, 1976) (Dufourd et al., 1998) (Dufourd et al., 1999) (Hofstede et al., 2010).

The first author received CAPES Scholarship - Proc. n°. 99999.001925/2015-06.

Rezende, L., Julia, S. and Cardoso, J.
Possibilistic WorkFlow Nets for Dealing with Cancellation Regions in Business Processes.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 2*, pages 126-133
ISBN: 978-989-758-187-8

Consequently, the soundness property is also undecidable for reset WF-net as proven in (van der Aalst et al., 2009). This shows that, although cancellation regions form a very useful modeling construct, increasing the expressive power of Petri nets, they complicate to a certain extent the process of verification of workflow models (Verbeek et al., 2007). Considering this, Wynn et al., in (Wynn et al., 2009), proposed a set of reduction rules for workflow nets with reset nets to facilitate the verification of workflow models. The inspiration for such rules came from earlier reduction rules for Petri nets without reset arcs. Typically, a reduction rule will decrease the number of elements under consideration by removing certain transitions and/or places in the net while preserving some interesting properties (Verbeek et al., 2010). However, a reset arc can never be abstracted entirely from a reset net. That is, if a net contains reset arcs, it is not possible to obtain a reduced net without any reset arc (Wynn et al., 2009). In addition, model translations normally introduce lots of "dummy" transitions that do not correspond to real events.

Note that it is far from trivial to express the desired behavior without reset arcs given that, to remove them, all possible markings should be considered, making the process model completely unreadable and intractable (van der Aalst et al., 2009). Taking this into account, an approach based on WF-nets and on possibilistic Petri nets is proposed to deal with all possible markings when the reset arcs are disregarded through the use of pseudo-firings. In particular, a kind of possibilistic WorkFlow net will be defined to consider, even without being explicitly enumerated, all the possible cancellation situations during real time execution of the process model. In addition, such an approach will preserve the decidability of the good properties of the model, such as the soundness property, given that the reset arcs are not consider in the process model.

The remainder of this paper is as follow: in section 2, the definition of WF-nets and soundness property are provided. In section 3, an overview of possibilistic Petri nets is given. In section 4, the possibilistic WorkFlow net is presented and a simplified version of a credit card application process illustrates the approach. Finally, section 5 concludes this work with a short summary, an assessment based on the approach presented and an outlook on future work proposals.

2 WORKFLOW NET

A Petri net that models a workflow process is called a WorkFlow net (WF-net) (van der Aalst and van Hee, 2004). A WF-net needs to possess the following properties (van der Aalst, 1998):

- It has only one source place, named *Start* and only one sink place, named *End*. These are special places, such that the place *Start* has only outgoing arcs and the place *End* 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 activity t (transition) and condition p (place) should be on a path from place *Start* to place *End*.

As previously mentioned, an activity can be associated to a transition in a WF-net. However, in order to explicitly indicate the beginning and the end of each activity in execution, two sequential transitions plus a place to model an activity is used. The first transition represents the beginning of the activity, the place the activity, and the second transition represents the end of the activity (Wang et al., 2009).

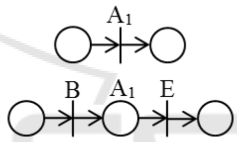


Figure 1: WorkFlow net model of an activity.

As shown in figure 1, transition B represents the beginning of an activity execution; E represents the end of the activity execution. Place A_1 represents the activity in execution. From a reachability analysis perspective, figure 1 can be reduced to a single transition, which represents the entire activity execution as a single logic unit.

Soundness is a correctness criterion defined for WF-nets and is related to its dynamics. A WF-net is sound if, and only if, the following three requirements are satisfied (van der Aalst and van Hee, 2004):

- For each token put in the place *Start*, one and only one token appears in the place *End*.
- When the token appears in the place *End*, all the other places are empty for this case.
- For each transition (activity), 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.

A method for the qualitative analysis of WF-nets (soundness verification) based on the proof trees of linear logic is presented in (Soares Passos and Julia, 2009) and another based on a reachability graph is presented in (van der Aalst et al., 2011).

3 POSSIBILISTIC PETRI NET

Possibilistic Petri nets are derived from Object Petri nets (Sibertin-Blanc, 2001). As characterized in the approach presented in (Cardoso, 1999), a possibilistic Petri net is a model where a marked place corresponds to a possible partial state, a transition to a possible state change, and a firing sequence to a possible behavior. The main advantage in working with possibilistic Petri nets is that they allow for the updating of a system state at a supervisory level with ill-known information without necessarily reaching inconsistent states.

A possibilistic Petri net model associates a possibility distribution $\Pi_o(p)$ to the location of an object o, p being a place of the net. $\Pi_o(p) = 1$ represents the fact that p is a possible location of o, and $\Pi_o(p) = 0$ expresses the certainty that o is not present in place p. Conventionally, a marking in a possibilistic Petri net is then a mapping:

$$M : O \times P \longrightarrow \{0,1\}$$

where O is a set of objects and P a set of places. If $M(o, p) = 1$, there exists a possibility of there being the object o in place p. On the contrary, if $M(o, p) = 0$, there exists no possibility of there being o in p. A marking M of the net allows one to represent:

- *A certain marking*: each token is located in only one place (well-known state). Then $M(o, p) = 1$ and $\forall p_i \neq p, M(o, p_i) = 0$.
- *An uncertain marking*: each token location has a possibility distribution over a set of places. It cannot be asserted that a token is in a given place, but only that it is in a place among a given set of places. For example, if there exists a possibility at a certain time of having the same object o in two different places, p_1 and p_2, then $M(o, p_1) = M(o, p_2) = 1$.

A possibilistic marking will correspond in practice to knowledge concerning a situation at a given time.

In a possibilistic Petri net, the firing (certain or uncertain) of a transition t is decomposed into two steps:

- *Beginning of a firing*: objects are put into output places of t but are not removed from its input places.
- *End of a firing*: that can be a firing cancellation (tokens are removed from the output places of t) or a firing achievement (tokens are removed from the input places of t).

A certain firing consists of a beginning of a firing and an immediate firing achievement. An uncertain firing (or a pseudo-firing) that will increase the uncertainty of the marking can be considered only as the beginning of a firing (there is no information to confirm whether the normal event associated with the transition has actually occurred or not). To a certain extent, pseudo-firing is a way of realizing forward deduction.

The interpretation of a possibilistic Petri net is defined by attaching to each transition an authorization function $\eta_{x_1,\ldots,x_n}$ defined as follows:

$$\eta_{x_1,\ldots,x_n} : T \longrightarrow \{False, Uncertain, True\}$$

where $x_1, \ldots, x_n$ are the variables associated with the incoming arcs of transition t (when considering the underlying Object Petri net).

If $o_1, \ldots, o_n$ is a possible substitution for $x_1, \ldots, x_n$ for the firing of t, then several situations can be considered:

- t is not enabled by the marking but the associated interpretation is true; an inconsistent situation occurs and a special treatment process of the net is activated;
- t is enabled by a certain marking and the interpretation is true; then a classical firing (with certainty) of an object Petri net occurs;
- t is enabled by a certain marking and the interpretation is uncertain; then the transition is pseudo-fired and the imprecision is increased;
- t is enabled by an uncertain marking; if the interpretation is uncertain, t is pseudo-fired;
- t is enabled by an uncertain marking and the interpretation is true: a recovery algorithm, presented in (Cardoso et al., 1989), is called and a new computation of the possibility distribution of the objects involved in the uncertain marking is realized in order to go back to a certain marking.

Concepts concerning possibilistic Petri nets will be illustrated through a practical example in section 4.

4 POSSIBILISTIC WORKFLOW NET FOR CANCELLATION

As pointed out in the introduction, cancellation is an important concept in WFMS's, which captures the interference of an activity in the execution of others in certain circumstances. In order to elaborate such a concept, in (Dufourd et al., 1998), reset arcs were introduced into the process model, thus allowing for an increase to its expressive power and to cancel the execution of the activities that belong to cancellation region. However, reachability and, consequently, the soundness property are undecidable for Petri nets (or WF-nets) with reset arcs.

To consider the concept of cancellation, such that the reachability and soundness property continue decidable, a model of the process based on WF-nets and on possibilistic Petri nets is proposed. Such a model, through the use of pseudo-firings, will then be able to deal with several markings that must be considered when the reset arcs are not considered.

The notions of cancellation activity and cancellation case can be generalized to the notion of the cancellation region, whereby an arbitrary region of a workflow specification can be subjected to a cancellation action (Wynn, 2006). In the model proposed in this paper, if an activity belongs to a cancellation region, its beginning and end transition will have a certain and uncertain interpretation attached to each. However, if an activity does not belong to a cancellation region, its beginning and end transition will only have a certain interpretation attached to each. The certain interpretation is related to the beginning and end conditions of an activity and the uncertain interpretation to a cancellation event. Note that if the process model has more than one cancellation region, the uncertain interpretation will be a disjunction of the cancellation events related to each region.

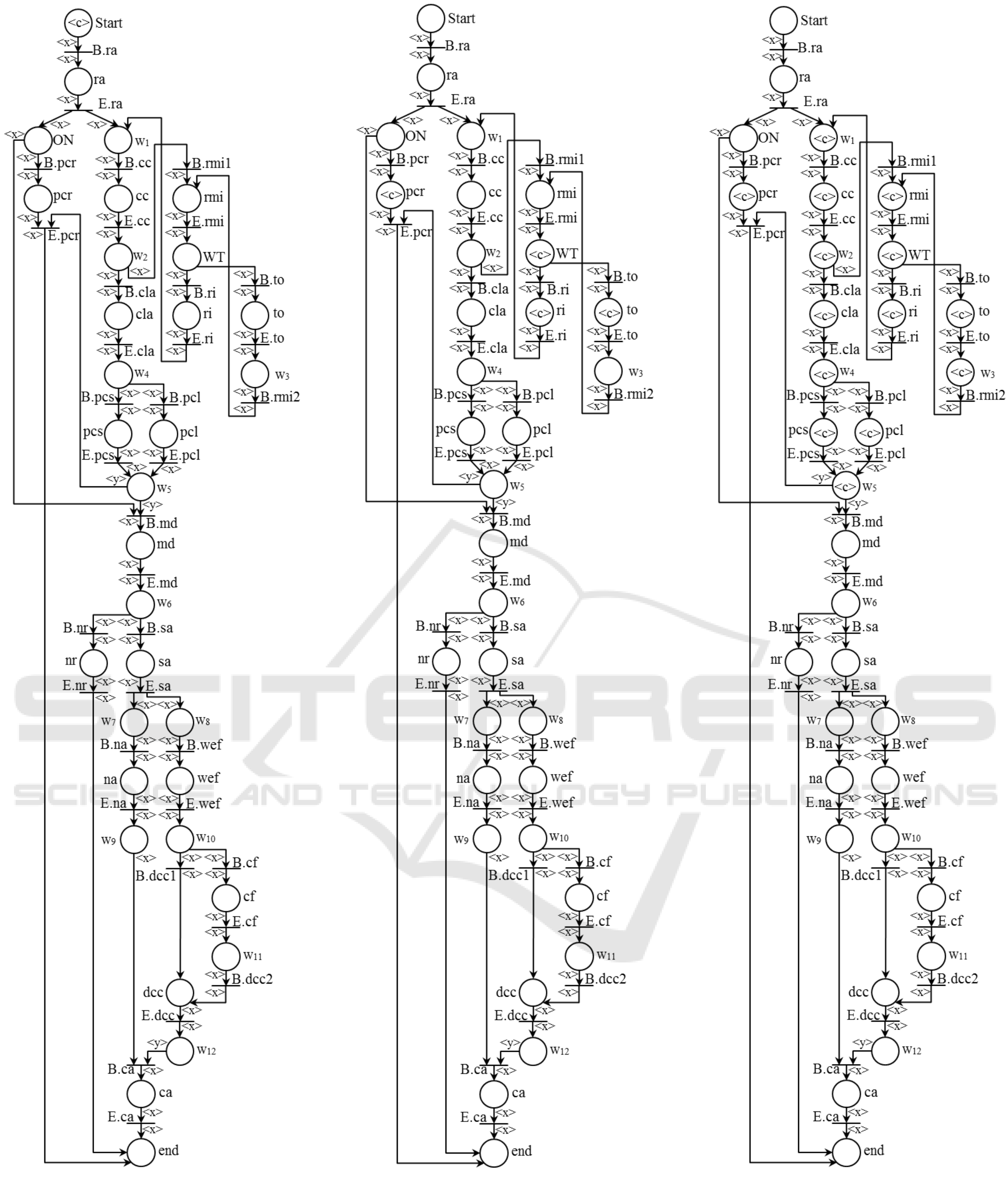
To illustrate the approach, a simplified version of a credit card application process, presented in (Wynn et al., 2009), will be used. The process starts when an applicant submits a credit card application (with the proposed amount). Upon receiving an application (ra), a credit clerk checks whether the submitted application is complete (cc). If not, the clerk requests additional information from the applicant (rmi) and waits (WT) until this information is received (ri) before proceeding. At the same time, a timer is set (to) so that if a certain period elapses before requested information is received, another request for information is sent again. For a complete application, the clerk first checks the requested loan amount (cla). It is then followed by additional checks to validate the applicants income and credit history. Different checks are performed depending on whether the requested loan is large (pcl) or small (pcs). The validated application is then passed on to a manager to make a decision (md). In the case of an acceptance, the credit card approval activity can start (sa). The applicant is notified of the decision (na) and, at the same time, he/she is asked for his/her preference on any extra features (wef). The applicant can choose extra features such as rewards program or secondary cardholders (cf) before a credit card is produced and delivered (dcc). This indicates the completion of the approval activity (ca) and the process ends. For a rejected application, the applicant is notified of the rejection (nr) and the process ends. An interesting feature of this process is that an applicant can request to cancel an ongoing application (ON) at any time after it was received (ra) and before the manager makes a decision (md), i.e., the activities *cc*, *rmi*, *ri*, *to*, *cla*, *pcs* and *pcl* belong to a cancellation region and the activity *pcr* is responsible for capturing the withdrawal of an ongoing application.

The possibilistic WF-net with objects in Figure 2(a) depicts the credit card application process. The symbol $< c >$ is an object belonging to the class "Credit", as well as variables x and y, and all the model's places. Each transition has an interpretation and an action attached to it defined by the designer. The interpretation is used to manage the occurrence of each event in the system by imposing restrictions on the firing of transitions. An action is an application that involves some specific methods applied on the attributes of the formal variables associated with the incoming arcs, allowing for the modification of some specific attributes of the object $< c >$. However, in the process model presented in this article, the actions will not be described given that they do not interfere in the understanding of the approach.

Knowing that the activities *cc*, *rmi*, *ri*, *to*, *cla*, *pcs* and *pcl* belong to a cancellation region, their beginning and end transitions will have a certain and uncertain interpretation attached to each of them. All the other activities (*ra*, *pcr*, *md*, *nr*, *sa*, *na*, *wef*, *dcc*, *cf* and *ca*) will have only a certain interpretation. Table 1 shows the authorization functions (η) for each transition belonging to the possibilistic WF-net shown in Figure 2(a). The certain and uncertain interpretations attached to each transition by the authorization function are respectively represented by the columns with the subscription "*true if*" and "*uncertain if*". The false interpretation is not represented in Table 1, but it is evaluated as true when the certain and uncertain interpretations are evaluated as false at the same time or, otherwise, it is evaluated as false. Finally, if a transition does not have an uncertain interpretation attached to it, the space concerning the uncertain interpretation in Table 1 is empty.

The conditions used in the interpretation of the transitions correspond to the following interpretations:

- "*begRA*", "*begCC*", "*begRMI*", "*begRI*", "*begTO*", "*begCLA*", "*begPCS*", "*begPCL*", "*begMD*", "*begNR*", "*begSA*", "*begNA*", "*begWEF*", "*begDCC*", "*begCF*" and "*begCA*" represent, respectively, the beginning of the execution of the activities *ra*, *cc*, *rmi*, *ri*, *to*, *cla*, *pcs*, *pcl*, *md*, *nr*, *sa*, *na*, *wef*, *dcc*, *cf* and *ca*;
- "*endRA*", "*endCC*", "*endRMI*", "*endRI*",



(a) Credit card application process modeled by a possibilistic WF-net.

(b) The uncertain marking obtained after the pseudo-firing of *B.ri* and *B.to*.

(c) The uncertain marking obtained after the pseudo-firing of the transitions.

Figure 2.

"*endTO*", "*endCLA*", "*endPCS*", "*endPCL*", "*endMD*", "*endNR*", "*endSA*", "*endNA*", "*endWEF*", "*endDCC*", "*endCF*" and "*endCA*" represent, respectively, the end of the execution of the activities *ra*, *pcr*, *cc*, *rmi*, *ri*, *to*, *cla*, *pcs*, *pcl*, *md*, *nr*, *sa*, *na*, *wef*, *dcc*, *cf* and *ca*;

- "*cancel*" represents the request of an applicant to cancel an ongoing application.

Considering the process model represented in Fig-

Table 1: Authorization functions of the transitions.

	true if	***uncertain if***
$\eta_x(B.ra) =$	$x.begRA$	
$\eta_x(E.ra) =$	$x.endRA$	
$\eta_x(B.pcr) =$	$cancel$	
$\eta_{xy}(E.pcr) =$	$cancel$	
$\eta_x(B.cc) =$	$x.begCC \wedge \neg cancel$	$cancel$
$\eta_x(E.cc) =$	$x.endCC \wedge \neg cancel$	$cancel$
$\eta_x(B.rmi1) =$	$x.begRMI \wedge \neg cancel$	$cancel$
$\eta_x(B.rmi2) =$	$x.begRMI \wedge \neg cancel$	$cancel$
$\eta_x(E.rmi) =$	$x.endRMI \wedge \neg cancel$	$cancel$
$\eta_x(B.ri) =$	$x.begRI \wedge \neg cancel$	$cancel$
$\eta_x(E.ri) =$	$x.endRI \wedge \neg cancel$	$cancel$
$\eta_x(B.to) =$	$x.begTO \wedge \neg cancel$	$cancel$
$\eta_x(E.to) =$	$x.endTO \wedge \neg cancel$	$cancel$
$\eta_x(B.cla) =$	$x.begCLA \wedge \neg cancel$	$cancel$
$\eta_x(E.cla) =$	$x.endCLA \wedge \neg cancel$	$cancel$
$\eta_x(B.pcs) =$	$x.begPCS \wedge \neg cancel$	$cancel$
$\eta_x(E.pcs) =$	$x.endPCS \wedge \neg cancel$	$cancel$
$\eta_x(B.pcl) =$	$x.begPCL \wedge \neg cancel$	$cancel$
$\eta_x(E.pcl) =$	$x.endPCL \wedge \neg cancel$	$cancel$
$\eta_{xy}(B.md) =$	$x.begMD \wedge \neg cancel$	
$\eta_x(E.md) =$	$x.endMD$	
$\eta_x(B.nr) =$	$x.begNR$	
$\eta_x(E.nr) =$	$x.endNR$	
$\eta_x(B.sa) =$	$x.begSA$	
$\eta_x(E.sa) =$	$x.endSA$	
$\eta_x(B.na) =$	$x.begNA$	
$\eta_x(E.na) =$	$x.endNA$	
$\eta_x(B.wef) =$	$x.begWEF$	
$\eta_x(E.wef) =$	$x.endWEF$	
$\eta_x(B.dcc1) =$	$x.begDCC$	
$\eta_x(B.dcc2) =$	$x.begDCC$	
$\eta_x(E.dcc) =$	$x.endDCC$	
$\eta_x(B.cf) =$	$x.begCF$	
$\eta_x(E.cf) =$	$x.endCF$	
$\eta_{xy}(B.ca) =$	$x.begCA$	
$\eta_x(E.ca) =$	$x.endCA$	

ure 2(a), an object in the place *md* indicates that the application was validated and passed on to a manager to make a decision, thus disabling any request on the part of the applicant to cancel it. However, an object in the place *pcr* indicates that the application received a request for withdrawal and, consequently, all activities that belong to the cancellation region must be stopped if they are currently executing or disabled if they are scheduled.

If the applicant does not request the cancellation of the application, all the transition firings will be certain and all the markings will be precise. However, if the applicant makes a request for cancellation, some pseudo-firing will have to occur until that the transition $E.pcr$ is enabled by a uncertain marking and the process can be canceled correctly.

The general behavior of a WFMS based on possibilistic WF-net models will be based on the possibilistic *token player*, given by the activity diagram in Figure 3.

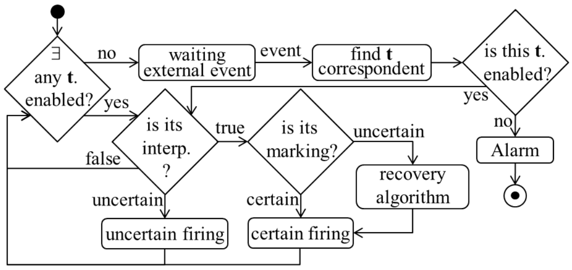


Figure 3: Possibilistic token player algorithm.

To illustrate a possible cancellation request, let us assume that the activities *ra*, *cc* and *rmi* have already been executed, i.e., the transitions $B.ra$, $E.ra$, $B.cc$, $E.cc$, $B.rmi$ and $E.rmi$ have already been fired with certainty. If the applicant requests the cancellation of the application, i.e., the condition "*cancel*" is true, the following scenario will occur:

- the transition $B.pcr$ is enabled by a certain making and its interpretation is true ($\eta_{<c>}(B.pcr) = true$). Then, $B.pcr$ is fired with certainty, i.e. the object $< c >$ is removed from the place ON, a new object $< c >$ is produced in the place pcr and the actions associated to the transition (when they exist) are executed;
- the authorization functions ($\eta_{<c>}$) attached to transitions $B.cc$, $E.cc$, $B.cla$, $E.cla$, $B.pcs$, $E.pcs$, $B.pcl$, $E.pcl$, $B.rmi1$, $E.rmi$, $B.ri$, $E.ri$, $B.to$, $E.to$ and $B.rmi2$ are all evaluated as uncertain. However, only the transitions $B.ri$ and $B.to$ can be pseudo-fired as they are the only ones enabled by the object $< c >$. Then, both $B.ri$ and $B.to$ are pseudo-fired and, consequently, copies of the object $< c >$ are produced in the places *ri* and *to*, respectively. Note that the object $< c >$ is not removed from the place WT (uncertain marking) (Figure 2(b));
- considering the new marking, the transitions $E.ri$ and $E.to$ are enabled by an uncertain marking and the interpretation attached to them is uncertain ($\eta_{<c>}(E.ri) = \eta_{<c>}(E.to) = uncertain$). Then, they are pseudo-fired and copies of the object $< c >$ are produced in the places w_1 and w_3, respectively;
- considering the evolution of the marking, the following transitions $B.cc$, $B.rmi2$, $E.cc$, $B.cla$, $E.cla$, $B.pcs$, $B.pcl$ and $E.pcs$ will be pseudo-fired given that the interpretation attached to them is uncertain, therefore, they will be enabled by an uncertain marking. Consequently, copies of the object $< c >$ will be produced in the places *cc*, *rmi*, w_2, *cla*, w_4, *pcs*, *pcl* and w_5, respectively;

- the transitions $B.rmi1$, $E.rmi$ and $E.pcl$ are enabled by an uncertain marking and the interpretation attached to them is uncertain ($\eta_{<c>}(B.rmi1) = \eta_{<c>}(E.rmi) = \eta_{<c>}(E.pcl) = uncertain$). However, such transitions cannot be fired due to the existence of the object $< c >$ in their output places (rmi, WT and w_5, respectively) which, through their definition, prohibit the pseudo-firing as explained in (Cardoso et al., 1989);
- the actual marking of the process model is shown in Figure 2(c). For this marking, the transition $E.pcr$ is enabled by an uncertain marking and the interpretation attached to it is true ($\eta_{<cc>}(E.pcr) = true$). This situation occurs because the applicant requested the cancellation of the application and all activities between *ra* and *md* must be canceled or disabled. Consequently, to go back to the certain marking, the recovery algorithm, presented in (Cardoso et al., 1989), is called. Such an algorithm will achieve the pseudo-firing of the transitions $E.pcs$, $B.pcs$, $E.cla$, $B.cla$, $E.cc$, $B.cc$, $E.ri$ and $B.ri$ and cancel the pseudo-firing of the transitions $B.pcl$, $B.to$, $E.to$ and $B.rmi2$. After the execution of the recovery algorithm, the transition $E.pcr$ can be fired with certainty given that it is enabled by a certain marking and its interpretation is true. Therefore, allowing for the execution of the actions attached to it and thus, finalize the cancellation request and the workflow process (only the place *end* is marked at the end of the cancellation). Note that no action attached to a transition pseudo-fired is executed, even if its firing is achieved.

Through the use of pseudo-firings, a firing sequence is defined, which will transform the marking until the object $< c >$ is localized in the place w_5. Then, the process can be finalized with success through the certain firing of the transition $E.pcr$ after the execution of the recovery algorithm. The existence of this firing sequence is guaranteed by the soundness property, which ensures that there does not exist dead transitions in this WF-net.

5 CONCLUSIONS

In this article, a possibilistic WorkFlow net model was presented with the purpose of dealing with cancellation features in business processes. Combining the routing structure of WF-nets and uncertain reasoning of possibilistic Petri nets, the authors presented an approach that is able to treat the cancellation events through of the use of pseudo-firing, which give a description of a set of possible markings that are all reachable markings of the underlying WF-net. Such an approach was applied to an example of a simplified version of a credit card application process.

Other studies that deal with the problem of cancellation use reset arcs in the process model. Such arcs have as a consequence the loss of decidability over some important structural properties, such as reachability. To avoid this loss, other works proposed techniques for reducing the size of the net and consequently the quantity of reset arcs. However, a reset arc can never be abstracted entirely from a reset net, consequently, the structural properties continue being undecidable. In addition, lots of “dummy” transitions that do not correspond to real events can be introduced to the process model. Comparing these studies with the approach presented in this paper, the main advantage is that, as the reset arcs are not used in the process model, the decidability of the structural properties of the model, such as soundness property, are guaranteed. Furthermore, all the scenarios that correspond to a cancellation situation are considered during the real time execution, even without being explicitly enumerating in the process model.

As a future work proposal, the quality of this approach should be explicitly validated through a kind of experimental approach that allows for the programming of transition pseudo firing. It would seem that the CPN Tools software resources (Beaudouin-Lafon et al., 2001), developed by the computing science group of Aarhus University in Denmark, allows in particular for the use of complex function calculus associated with the model’s arcs. This should be able to program in a simple way some of the basic behaviors of a possibilistic token player implementing cancellation scenarios. In addition, it will be interesting to model and test a larger business process with more than one cancellation region.

ACKNOWLEDGEMENTS

The authors would like to thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Fundação de Amparo a Pesquisa do Estado de Minas Gerais (FAPEMIG) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) for financial support.

REFERENCES

Araki, T. and Kasami, T. (1976). Some decision problems related to the reachability problem for petri nets. *Theoretical Computer Science*, 3:85 – 104.

Beaudouin-Lafon, M., Mackay, W., Jensen, M., Andersen, P., Janecek, P., Lassen, M., Lund, K., Mortensen, K., Munck, S., Ratzer, A., Ravn, K., Christensen, S., and Jensen, K. (2001). Cpn/tools: A tool for editing and simulating coloured petri nets etaps tool demonstration related to tacas. In *Tools and Algorithms for the Construction and Analysis of Systems*, volume 2031, pages 574–577. Springer Berlin Heidelberg.

Cardoso, J. (1999). Time fuzzy petri nets. In *Fuzziness in Petri Nets*, pages 115 – 145. Springer.

Cardoso, J., Valette, R., and Dubois, D. (1989). Petri nets with uncertain markings. In *Applications and Theory of Petri Nets*, volume 483, pages 64 – 78.

Dufourd, C., Finkel, A., and Schnoebelen, P. (1998). Reset nets between decidability and undecidability. In *Proceedings of the 25th International Colloquium on Automata, Languages and Programming*, volume 1443, pages 103–115.

Dufourd, C., Jančar, P., and Schnoebelen, P. (1999). Boundedness of reset p/t nets. In *International Colloquium on Automata, Languages and Programming*, volume 1644, pages 301–310.

Hofstede, A. H. M. t., van der Aalst, W. M. P., Adams, M., and Russell, N., editors (2010). *Modern Business Process Automation - YAWL and its Support Environment*. Springer Science & Business Media.

Members, W. M. C. (1994). Glossary – a workflow management coalition specification. Technical report, Coalition, Workflow Management.

Sibertin-Blanc, C. (2001). Cooperative objects: Principles, use and implementation. In *Concurrent Object-Oriented Programming and Petri Nets*, volume 2001, pages 216–246.

Soares Passos, L. and Julia, S. (2009). Qualitative analysis of workflow nets using linear logic: Soundness verification. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 2843 –2847.

van der Aalst, W. M. P. (1998). The application of petri nets to workflow management. *Journal of Circuits Systems and Computers*, 8:21–66.

van der Aalst, W. M. P. (2000). Workflow verification: Finding control-flow errors using petri-net-based techniques. In *Business Process Management, Models, Techniques, and Empirical Studies*, pages 161–183.

van der Aalst, W. M. P., Hofstede, A. H. M. t., Kiepuszewski, B., and Barros, A. P. (2003). Workflow patterns. *Distrib. Parallel Databases*, 14:5–51.

van der Aalst, W. M. P. and van Hee, K. (2004). *Workflow Management: Models, Methods, and Systems*. MIT Press.

van der Aalst, W. M. P., van Hee, K., Hofstede, A. H. M. t., Sidorova, N., Verbeek, H., Voorhoeve, M., and Wynn, M. (2009). Soundness of workflow nets with reset arcs. In *Transactions on Petri Nets and Other Models of Concurrency III*, volume 5800, pages 50–70.

van der Aalst, W. M. P., van Hee, K. M., Hofstede, A. H. M. t., Sidorova, N., Verbeek, H. M. W., Voorhoeve, M., and Wynn, M. T. (2011). Soundness of workflow nets: Classification, decidability, and analysis. *Form. Asp. Comput.*, 23:333–363.

Verbeek, H., van der Aalst, W. M. P., and Hofstede, A. H. M. t. (2007). Verifying workflows with cancellation regions and or-joins: An approach based on relaxed soundness and invariants. *The Computer Journal*, 50:294–314.

Verbeek, H., Wynn, M., van der Aalst, W. M. P., and Hofstede, A. H. M. t. (2010). Reduction rules for reset/inhibitor nets. *Journal of Computer and System Sciences*, 76:125 – 143.

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

Wynn, M., Verbeek, H., van der Aalst, W. M. P., Hofstede, A. H. M. t., and Edmond, D. (2009). Soundness-preserving reduction rules for reset workflow nets. *Information Sciences*, 179:769–790.

Wynn, M. T. K. (2006). *Semantics, Verification, and Implementation of Workflows with Cancellation Regions and OR-joins*. PhD thesis, Queensland University of Technology.