Uncertain Marking for Dealing with Partial Parallelization 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, Deviations, Process Monitoring.

Abstract: In this paper, an approach based on WorkFlow nets and possibilistic Petri nets is proposed for dealing with flow deviations 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 considered to express in a more realistic way the ordering of human activities during real time execution of the process model. Combining both formalisms, a kind of possibilistic WorkFlow net is obtained. An example of flow deviations due to human behavior at a process monitoring level is presented.

1 INTRODUCTION

An organization produces value for its customers by executing various business processes (BP) which 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 Hee, 2004). Due to complexity and variety of BP, contemporary organizations use information technology to support activities which may include the automation of their processes.

A workflow process corresponds to the automation of a BP, in whole or part, during which documents, information or tasks are passed from one participant to another for a particular form of action, according to a set of procedural rules. A Workflow Management System (WFMS) is a system that completely defines, manages and executes workflow processes through the execution of software whose sequence of activities is driven by a computer representation of the workflow process logic (Members, 1994).

Many papers have already considered Petri net theory as an efficient tool for the modeling and analysis of WFMS (van der Aalst, 1998) (van der Aalst and Hee, 2004) (Soares Passos and Julia, 2009). The WorkFlow nets (WF-nets) (acyclic Petri net models used to represent BP) are defined in (van der Aalst and Hee, 2004).

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

Soundness property is an important criterion which needs to be satisfied when treating workflow processes. In fact, good properties of well-defined formal models such as WF-nets can easily be proved when BP are following a rigid structure that does not allow for deviations from the process description during real time execution. Although the explicit ordering of activities makes the coordination of activities relatively easy to understand, even for a non-technical stakeholder, it was shown that BP do not easily map to a rigid modeling structure (Kapuruge et al., 2010). Consequently, they are implemented such that they "fit the system", which can cause various problems (Pesic, 2008). First, considering that some activities executed in a BP depend on human resources which do not necessarily respect the rigorous definition of workflow processes, manual interventions become increasingly frequent as human employees attempt to manipulate workflow inputs and outputs to conform with changes in workplace practices. Second, these manual interventions reduce productivity and increase processing time. Since it is undertaken in an ad-hoc manner, manual handling incurs an added penalty: the corrective actions undertaken are not added to organizational memory and, as a consequence, natural process evolution is not incorporated into future iterations of the process (Adams, 2010).

Attempts at incorporating a certain level of flexibility in process definition have already been proposed by several authors.

Adams et al., in (Adams et al., 2005) and (Adams et al., 2006), defined the worklet approach as an extensible repertoire of self-contained sub-processes aligned to each task, from which a dynamic runtime selection is made depending on the context of the particular work instance. In addition, in (Adams, 2007) and (Adams et al., 2007), an extensible repertoire of self-contained exception-handling processes called “exlets” was incorporated in the worklet approach. However some good properties can be lost due to the dynamic changes.

In (Pesic and van der Aalst, 2006) and (Pesic et al., 2007), a declarative constraint-based approach that allows a process instance to deviate from the original behavior at runtime by adding/removing or softening/hardening the constraints and their mapping to process definitions is proposed. In (van der Aalst et al., 2009), an approach based on constraints that implicitly specifies the execution procedure, i.e., anything is possible as long as it is not explicitly forbidden is proposed. The problem with these approaches is that the sequencing of single activities specified for a certain process type is only limited to the satisfaction of predetermined constraints.

In (Mohammed et al., 2007), two process models coexist during the monitoring of the BP. The first one corresponds to the expected behavior of the process and the second one, based on the visible actions of human actors, is built dynamically. The two process models are then permanently compared and analyzed in order to detect possible deviations. The problem in dealing with two models is that the monitoring activity can easily be overloaded implying a decrease in the system’s performance.

In (Zazworka et al., 2009), the deviations and nonconformities between a planned and executed process are detected during the development of the software. In (Thompson and Torabi, 2009), a kind of declarative implicit model, essentially based on rules, is used to detect inconsistencies (non-conformant states in the process) and to accept possible deviations (nonconformant transitions between activities). The limitation of the methodology is that the specified rules only detect the non-conformance of the process but do not help in an explicit manner for the real time process monitoring.

Some authors, such as (Rozinat and van der Aalst, 2008)(Munoz-Gama, 2010)(van der Aalst et al., 2012), evaluated the level of conformance between the process model and the execution log of the process. For this, they developed metrics for measuring the relationship between predefined process models and actual results presented in the form of event logs. The problem with this approach is that the verification is carried out after the process execution.

A very promising alternative for dealing with flexibility in BP seems to be those approaches based on uncertain knowledge as that presented in (Cmpan and Oquendo, 2000). The model of the process is then given through fuzzy sets and possibilistic distributions.The advantage of fuzzy and possibility reasoning is the ability to naturally represent uncertain and imprecise information that exists when activities are executed by human employees.

One of the first studies which combines fuzzy and possibilistic representation of information with the precise structure of a Petri net when considering discrete event systems is that described in (Cardoso et al., 1999). The main feature of possibilistic/fuzzy Petri nets is to allow one to reason about the aspects of uncertainty and change in dynamic discrete event systems. Most of the examples presented by the authors of possibilistic Petri net were applied to flexible manufacturing systems.

In this paper, an approach based on WF-nets and possibilistic Petri nets is proposed to deal with flexibility in BP. In particular, a kind of possibilistic WorkFlow net will be defined to treat deviations incurred by manual interventions of the human employees during real time execution of the process model.

The remainder of this paper is as follow: in section 2, the definition of WF-nets and soundness correctness criterion are provided. In section 3, an overview of possibilistic Petri nets is given. In section 4, the possibilistic WorkFlow net is presented and a sample business process model 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 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 task t (transition) and condition p (place) should be on a path from place *Start* to place *End*.

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

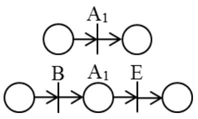


Figure 1: WorkFlow net model of a task.

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

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 T 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 $o_1,...,o_n$ is a possible substitution for $x_1,...,x_n$ for firing 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 THE PARTIAL PARALLELIZATION OF A SEQUENTIAL PROCESS

If a Petri net is used as a model for Business Processes in a WFMS, transitions will represent the state changes of the process. In particular, each event occurring during the execution of the process (beginning and ending of activities) will be associated with a transition as a boolean variable. Such a variable will be essentially seen as an external value corresponding to a message received from an activity (or send to an activity). Possibly, internal values depending on certain token attributes will enable some transitions too.

Petri net models can be directly executed using a specialized inference mechanism called "*token player* algorithm" that allows for a simplified monitoring of the represented process model. A classical "*token player* algorithm", as the one defined in (Cardoso and Valette, 1997), is only based on events under those conditions normally expected. If an unexpected event occurs, the process stops or needs to be repaired manually.

As pointed out in the introduction, the explicit ordering of activities makes the coordination of activities relatively easy to understand even for a non-technical stakeholder, however, due to rigid structure of workflow processes, several manual interventions have become increasingly frequent as human employees attempt to manipulate workflow inputs and outputs to conform with changes in workplace practices. From this, some deviations may occur between the model of the process and the real process execution. A classic deviation may be the initialization of an activity before the end of a previous one due to a deadline for example.

If inevitable system changes are handled haphazardly without being added to organizational memory in order that they be considered part of the evolutionary process, they can lead to major work disruptions increasing dissatisfaction to a point at which the entire system implementation is considered as a failure. In order to handle the changes during real process execution and register them in the organizational memory, a model of the process based on the routing structure of WF-nets and on uncertain marking along with the firing of possibilistic Petri nets is proposed for dealing with the deviations incurred by manual interventions by human employees during its real time execution.

A simple ordering process, presented in (Weske, 2007), will be used to illustrate the approach. This process features a sequence of activities. Firstly, the order is stored. After, the inventory is checked, the invoice is sent, the funds are received and the shipment is prepared. Finally, the goods are shipped and the process terminates with the archiving of the order. The possibilistic WF-net with objects in Figure 2 represents such a model.

$< c >$ is an object belonging to the class "Order", as well as variable x 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. The action is an application that involves the attributes or methods of formal variables associated with incoming arcs allowing for the modification of some specific attributes. However, in the model presented, the actions are not represented given that they do not interfere in understanding the approach. The true and uncertain interpretations are represented, respectively, by the letters T and U. Note that when the representation of the uncertain interpretation is omitted in the model, such interpretation is always evaluated as false. On the other hand, the false interpretation is evaluated as true when both the true interpretation and the uncertain interpretation are evaluated as false. The element τ represents the current time.

The conditions correspond to the folloing interpre-

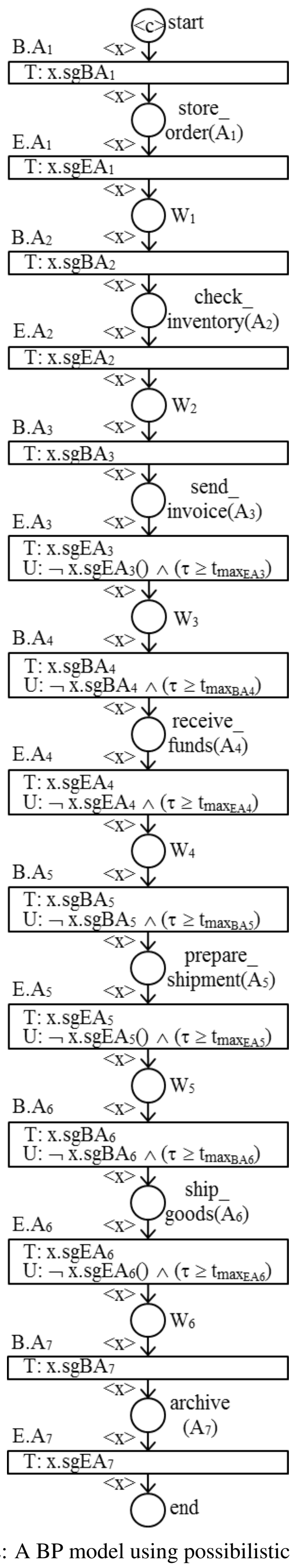


Figure 2: A BP model using possibilistic WF-net.

tations:

- "*sgBA*1", "*sgBA*2", "*sgBA*3", "*sgBA*4", "*sgBA*5", "*sgBA*6" and "*sgBA*7" represent, respectively, the beginning of the tasks "store order", "check inventory", "send invoice", "receive funds", "prepare shipment", "ship goods" and "archive";
- "*sgEA*1", "*sgEA*2", "*sgEA*3", "*sgEA*4", "*sgEA*5", "*sgEA*6" and "*sgEA*7" represent, respectively, the end of the tasks "store order", "check inventory", "send invoice", "receive funds", "prepare shipment", "ship goods" and "archive";
- $t_{max_{BA4}}$, $t_{max_{BA5}}$ and $t_{max_{BA6}}$ are maximum points in the timescale associated to a specific activity which indicate the maximum time allowed for the initialization of a task;
- $t_{max_{EA3}}$, $t_{max_{EA4}}$, $t_{max_{EA5}}$ and $t_{max_{EA6}}$ are maximum points in the timescale associated to a specific activity, which indicates the maximum time allowed to begin the conclusion of a task.

Due to the somewhat cautious policy realized by the BP, the normal expected behavior of the process is to execute all activities in sequence such that for an activity to start, it must wait for the end of the previous one. For example, the activity "prepare shipment" (A_5) starts only after receiving the funds (activity A_4). In consequence, BP instances based on this process model might suffer from long processing times, resulting in insufficient customer satisfaction.

An abnormal behavior will happen if, disregarding the rigid sequencing of the activities but considering a maximum time allowed to start or finish an activity, the activities A_5 and A_6, for example, start before the end of the activity A_4, allowing a lower processing time and sufficient customer satisfaction. In this case, some pseudo-firing will have to occur and the imprecision linked to some objects will increase. In doing so, some of the sequential activities, specified through the rigid structure of the sequential WF-net of figure 2, will be partially executed in parallel.

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

To illustrate a possible abnormal behavior, the following values are considered: $t_{max_{EA3}} = 24$, $t_{max_{BA4}} = 25$, $t_{max_{EA4}} = 49$, $t_{max_{BA5}} = 50$, $t_{max_{EA5}} = 74$, $t_{max_{BA6}} = 75$ and $t_{max_{EA6}} = 90$. In addition, it is assumed that the conditions become true on the following dates: $sgBA_1$ at $\tau = 0$, $sgEA_1$ at $\tau = 4$, $sgBA_2$ at $\tau = 7$, $sgEA_2$ at $\tau = 19$, $sgBA_3$ at $\tau = 20$, $sgEA_3$ at $\tau = 23$, $sgBA_4$ at $\tau = 24$, $sgEA_4$ at $\tau = 85$, $sgBA_5$ at $\tau = 86$, $sgEA_5$ at

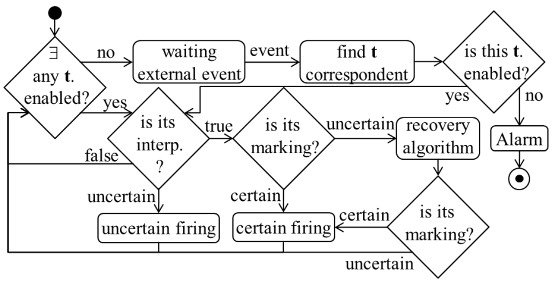
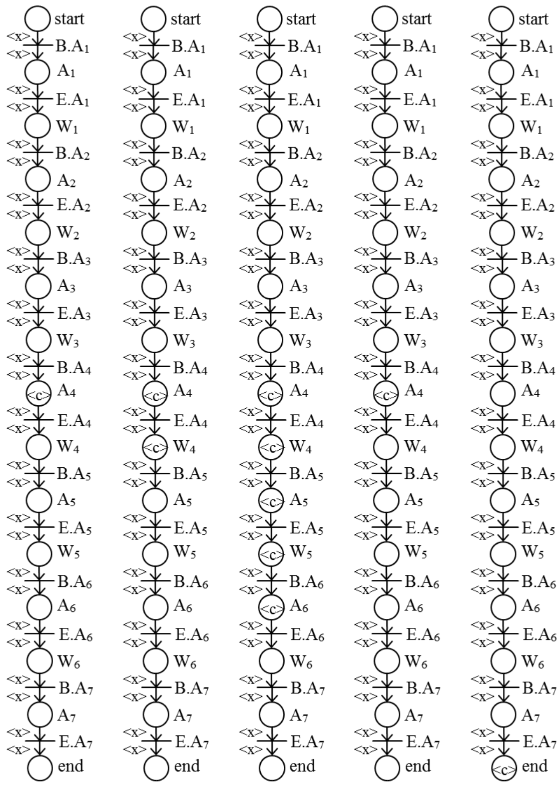


Figure 3: Possibilistic token player algorithm.

$\tau = 87$, $sgBA_6$ at $\tau = 88$, $sgEA_6$ at $\tau = 89$, $sgBA_7$ at $\tau = 90$ and $sgEA_7$ at $\tau = 100$.

Taking into consideration the values defined above (the maximum points and the conditions), the following scenario will occur:

- at the following times $\tau = 0$, $\tau = 4$, $\tau = 7$, $\tau = 19$, $\tau = 20$, $\tau = 23$ and $\tau = 24$, the transitions $B.A_1$, $E.A_1$, $B.A_2$, $E.A_2$, $B.A_3$, $E.A_3$ and $B.A_4$ are respectively fired with certainty given that they are enabled by a certain marking and the interpretation attached to them is true ($\eta_{<c>}(t) = true$). The resulting state of the process after this firing sequence is shown in Figure 4(a).
- at current time $\tau = 49$, the transition $E.A_4$ is enabled by a certain making and its interpretation is uncertain ($\eta_{<c>}(E.A_4) = uncertain$). Then, $E.A_4$ is pseudo-fired (Figure 4(b));
- at the following times $\tau = 50$, $\tau = 74$ and $\tau = 75$, the transitions $B.A_5$, $E.A_5$ and $B.A_6$ are respectively pseudo-fired given that they are enabled by uncertain markings and the interpretation attached to them is uncertain ($\eta_{<c>}(t) = uncertain$). The resulting state of the process after this firing sequence is shown in Figure 4(c);
- at current time $\tau = 85$, the transition $E.A_4$ is enabled by an uncertain marking and its interpretation becomes true ($\eta_{<c>}(E.A_4) = true$). This situation occurs because the notification of the receipt of the funds arrived late and the activities A_5 and A_6 started before it ended, so as not to increase the processing time of the process such that customer satisfaction is not affected. Consequently, the recovery algorithm described in Algorithm 1 is called to cancel some pseudo-fired transitions. After the execution of the recovery algorithm, the transition $E.A_4$ can be fired with certainty given that it is enabled by a certain marking (Figure 4(d)) and its interpretation is true. It is important to note that if the marking is not certain after the processing of the algorithm, the transition $E.A_4$ cannot be fired with certainty and the process instance must wait for a new event in order that the process state can go back to a certain marking;
- finally, at the following times $\tau = 86$, $\tau = 87$, $\tau = 88$, $\tau = 89$, $\tau = 90$ and $\tau = 100$, the transitions $B.A_5$, $E.A_5$, $B.A_6$, $E.A_6$, $B.A_7$ and $E.A_7$ are respectively fired with certainty given that they are enabled by certain markings and the interpretation attached to them is true ($\eta_{<c>}(t) = true$). The resulting state of the process, after this firing sequence, is shown in Figure 4(e).

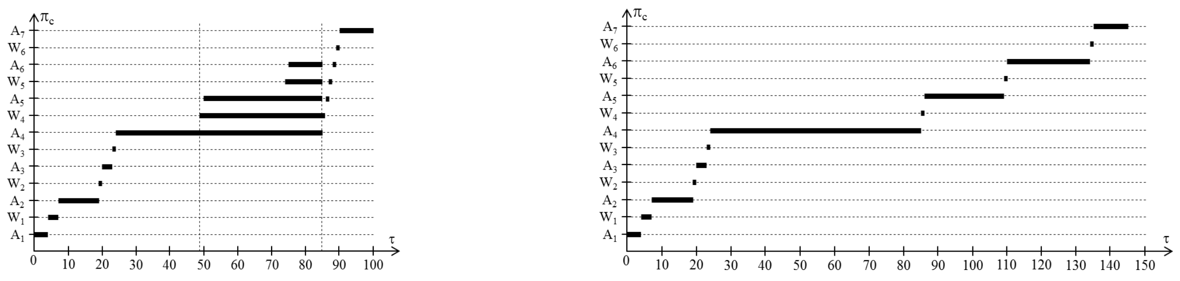


(a) $\tau = 24$ (b) $\tau = 49$ (c) $\tau = 75$ (d) $\tau = 85$ (e) $\tau = 100$

Figure 4: Simulation results of the scenario.

Figure 5(a) depicts the possibility distributions of instance $< c >$ as functions of time relative to the scenario described above(the black lines represent a possibility equal to 1 and the bright lines a possibility equal to 0). Note that the activities A_5 and A_6 after the execution of the recovery algorithm do not spend all the processing time predefined because they have already been initialized. In addition, observing the possibility distributions of instance $< c >$, it is easy to verify that the activities A_4, A_5 and A_6 were pseudo-parallelized during the time period between 49 and 85 even with the fact that in the process model they are strictly described in a pure sequential way.

In order to show the difference in the processing time spent when the process model is strictly executed



(a) Pseudo-parallelized (b) Strictly sequential

Figure 5: Possibility distributions of object location $< c >$.

```
Algorithm 1: Recovery algorithm.
Inputs:
    WN: WF-net with an uncertain marking
    t: transition
Output:
    WN: WF-net with a new marking
1.  if $t$ was pseudo-fired
2.  then $LP \leftarrow (t\bullet)$
3.          cancel the firing of $t$
2.  else $LP \leftarrow \phi$
4.  while $LP \neq \phi$ do
5.      $p \leftarrow$ LL[0]
6.      while $\exists\, t_j \in (p\bullet) \mid t_j$ was pseudo-fired do
7.          $LP \leftarrow LP \cup (t_j\bullet)$
8.          cancel the firing of $t_j$
```

sequentially, Figure 5(b) depicts the possibility distributions of instance $< c >$ as functions of time considering the same processing time that each task spent during its execution in the scenario described above. Comparing both figures (Figure 5(a) and 5(b)), it is easy to see that the processing time spent by the first is 33% lower than the second one.

Considering the change patterns suitable for providing flexibility in time execution proposed by Weber et al. in (Weber et al., 2008), the proposed approach incorporates, during real time execution, the adaptation pattern "Parallelize Process Fragments".

5 CONCLUSIONS

In this article, a possibilistic WorkFlow net model was presented with the purpose of dealing with deviation situations in BP. Combining the routing structure of WF-nets and the uncertain reasoning of possibilistic Petri nets, the authors presented an approach that can deal with deviation situations that can be reached during the real time execution when human behaviour is considered. Such an approach was applied to an example of a simple ordering process.

Other studies that deal with the problem of deviations in workflow processes create new structures during the execution process, which can lose the good properties of the process or only detect the deviations after the conclusion of the process. Comparing these studies with the approach presented in this paper, the main advantage is that the good properties of the model will be preserved because the structure in the process (the different routing patterns) will not be altered during the real time execution. Some deviations will be allowed and recorded in the organizational memory for being posteriorly considered in the process evolution. However, the allowed deviations will be limited due to the possibilistic behavior.

As a future work proposal, it will be interesting to deal with fuzzy time information in an explicit way in the structure of a WF-net as the one presented in (Cardoso, 1999). The use of fuzzy dates can allow a flexible control of a time schedule describing the causality relations between the events and improving the updating process. Such an approach should be interesting at the monitoring level in order to diminish the imprecision of human intervention.

In addition, in order to validate experimentally such an approach, the model of the possibilistic WorkFlow net will have to be implemented on a Petri net software tool allowing for the programming of transition pseudo firing. It seems that the CPN Tools software resources (Beaudouin-Lafon et al., 2001) should be able to implement in a simple way some of the basic behaviors of a possibilistic token player given that they combine advanced interaction techniques into a consistent interface for editing, simulating, and analyzing Coloured Petri Nets. Such an implementation should prove the gain in the overall process performance.

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 Cientfico e Tecnolgico (CNPq) for financial support.

REFERENCES

Adams, M. (2007). *Facilitating Dynamic Flexibility and Exception Handling for Workflows*. PhD thesis, Queensland University of Technology.

Adams, M. (2010). Chapter 4: Dynamic workflow. In *Modern Business Process Automation: YAWL and its Support Environment*, pages 123–145.

Adams, M., Hofstede, A. H. M. t., Edmond, D., and van der Aalst, W. M. P. (2005). Facilitating flexibility and dynamic exception handling in workflows through worklets. In *CAiSE*, volume 161, pages 45–50.

Adams, M., Hofstede, A. H. M. t., Edmond, D., and van der Aalst, W. M. P. (2006). Worklets: A service-oriented implementation of dynamic flexibility in workflows. In *Proceedings of the OTM*, volume 4275, pages 291–308.

Adams, M., Hofstede, A. H. M. t., van der Aalst, W. M. P., and Edmond, D. (2007). Dynamic, extensible and context-aware exception handling for workflows. In *OTM*, volume 4803, pages 95–112.

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.

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

Cardoso, J. and Valette, R. (1997). *Redes de Petri*. DAUSFC.

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.

Cardoso, J., Valette, R., and Dubois, D. (1999). Possibilistic petri nets. *IEEE Transactions on SMC, Part B*, 29:573 –582.

Cmpan, S. and Oquendo, F. (2000). Dealing with software process deviations using fuzzy logic based monitoring. *SIGAPP Appl. Comput. Rev.*, 8:3–13.

Kapuruge, M., Han, J., and Colman, A. W. (2010). Support for business process flexibility in service compositions: An evaluative survey. In *Australian Software Engineering Conference*, pages 97–106.

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

Mohammed, K., Redouane, L., and Bernard, C. (2007). A deviation-tolerant approach to software process evolution. In *IWPSE*, pages 75–78.

Munoz-Gama, J. (2010). Algorithms for process conformance and process refinement. Master's thesis, Universitat Politècnica de Catalunya.

Pesic, M. (2008). *Constraint-based Workflow Management Systems: Shifting Control to Users*. PhD thesis, Eindhoven University of Technology.

Pesic, M., Schonenberg, M., Sidorova, N., and van der Aalst, W. M. P. (2007). Constraint-based workflow models: Change made easy. In *OTM*, volume 4803, pages 77–94.

Pesic, M. and van der Aalst, W. M. P. (2006). A declarative approach for flexible business processes management. In *BPM Workshops*, volume 4103, pages 169–180.

Rozinat, A. and van der Aalst, W. M. P. (2008). Conformance checking of processes based on monitoring real behavior. *Information Systems*, 33:64–95.

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 *IEEE SMC*, pages 2843 –2847.

Thompson, S. and Torabi, T. (2009). An observational approach to practical process non-conformance detection. In *ICADIWT*, pages 62–67.

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., Adriansyah, A., and van Dongen, B. (2012). Replaying history on process models for conformance checking and performance analysis. *WIREs: Data Mining and Knowledge Discovery*, 2:182–192.

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

van der Aalst, W. M. P., Pesic, M., and Schonenberg, H. (2009). Declarative workflows: Balancing between flexibility and support. *Computer Science - Research and Development*, 23:99–113.

van der Aalst, W. M. P., van Hee, K. M., ter Hofstede, A. H. M., 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.

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.

Weber, B., Reichert, M., and Rinderle-Ma, S. (2008). Change patterns and change support features - enhancing flexibility in process-aware information systems. *Data and Knowledge Engineering*, 66:438–466.

Weske, M. (2007). *Business Process Management: Concepts, Languages, Architectures*. Springer-Verlag New York, Inc.

Zazworka, N., Basili, V. R., and Shull, F. (2009). Tool supported detection and judgment of nonconformance in process execution. In *ESEM*, pages 312–323. ACM / IEEE Computer Society.