# Verifying Workflows with Cancellation Regions and OR-joins: An Approach Based on Reset Nets and Reachability Analysis

M.T. Wynn[1], W.M.P. van der Aalst[1,2], A.H.M. ter Hofstede[1] and D. Edmond[1]

1. Queensland University of Technology, Australia
{*m.wynn,d.edmond,a.terhofstede*}*@qut.edu.au*
2. Eindhoven University of Technology, The Netherlands
{*w.m.p.v.d.aalst*}*@tm.tue.nl*

**Abstract.** When dealing with complex business processes (e.g., in the context of a workflow implementation or the configuration of some process-aware information system), it is important but sometimes difficult to determine whether a process contains any errors. The concepts such as cancellation and OR-joins occur naturally in business scenarios but the presence of these features in process models poses new challenges for verification. We take on the challenge of finding new verification techniques for workflows with cancellation regions and OR-joins. The proposed approach relies on reset nets and reachability analysis. We present these techniques in the context of workflow language YAWL that provides direct support for these features. We have extended the graphical editor of YAWL with these diagnostic features.
**Keywords**: Workflow verification, Cancellation, OR-joins, Reset nets, YAWL.

## 1 Introduction

Given that deployed workflows may execute for a long time and may take many actions that cannot be undone in a simple manner, it is desirable to detect errors at design time. Workflow verification is concerned with determining, *in advance*, whether a workflow exhibits certain desirable behaviours. In [9], verification of workflow nets is discussed in detail and Petri net analysis techniques are used to detect whether a workflow net is sound or not. Unfortunately, these results are not straight-forwardly transferable to situations where languages are involved that use concepts not easily expressed through Petri nets (e.g., cancellation and OR-joins).

*Cancellation* captures the interference of an activity in the execution of others in some circumstances. An *OR-join* is used in situations when we need to model "wait and see" behaviour for synchronisation. The OR-join and cancellation are two of the workflow patterns described in [4]. The workflow language YAWL provides direct support for all but one of these patterns [3] and in this paper, verification techniques are proposed in the context of this language. Due to limited space in this paper, we focus on the correctness notions for YAWL workflows and provide a brief discussion of our verification approach. A more complete discussion can be found in [11].

## 2 Correctness notions for YAWL workflows

The workflow language YAWL is a general and powerful language grounded in workflow patterns and in Petri nets [3]. The introduction of new concepts such as cancellation regions or OR-joins in workflows requires the adaptation of existing verification techniques to determine the correctness of a workflow. In addition, it leads to new properties that need to be analysed. In this paper, we propose four desirable properties for YAWL workflows: *soundness*, *weak soundness*, *irreducible cancellation regions*, and *immutable OR-joins*. Using the notions of coverability and reachability, we will demonstrate how these properties are formulated and algorithmic approaches are derived.

A YAWL net is formally defined as an eYAWL-net and it is represented by the tuple $(C, \mathbf{i}, \mathbf{o}, T, F, split, join, rem, nofi)$ where $C$ is a set of conditions, $T$ is a set of tasks, $i$ and $o$ are unique input and output conditions, $F$ is the flow relation, $split$ and $join$ specify the split and join behaviours of each task, $rem$ specifies the cancellation region for a task and $nofi$ specifies the multiplicity of each task. Formal definitions and notations for YAWL can be found in [3]. In Figure 1, we present a YAWL net which describes the “lifecycle” of a student who is required to take an exam and in parallel may already book a flight to go on holidays after passing the exam. In this “holiday scenario”, a student decides to reward himself/herself by going on holidays if he/she passes the exam and cancel the plans if he/she fails the exam. One of the fundamental properties

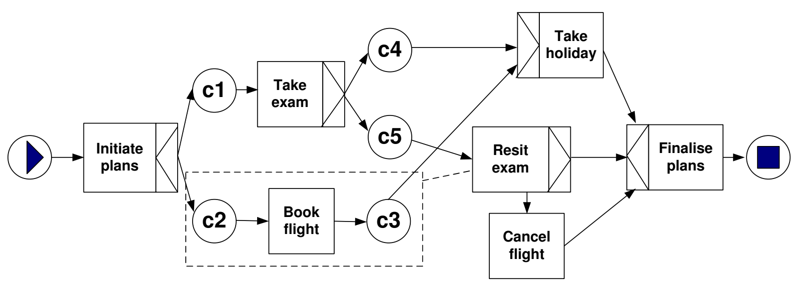


**Fig. 1.** Holiday scenario

of workflow is the soundness property and the soundness definition for YAWL is based on the definition for WF-nets [1].

**Definition 1 (Soundness).** *Let $N$ be an eYAWL-net and $M_i, M_o$ be the initial and end markings. $N$ is sound iff: 1)* ***option to complete:*** *for every marking $M$ reachable from $M_i$, there exists an occurrence sequence leading from $M$ to $M_o$, and 2)* ***proper completion:*** *the marking $M_o$ is the only marking reachable from $M_i$ with at least one token in condition $o$, and 3)* ***no dead tasks:*** *for every task $t \in T$, there is a marking $M$ reachable from $M_i$ such that $t$ is enabled at $M$.*

The concepts of reachability and coverability are defined using the YAWL semantics as defined in [3, 12]. To detect the soundness property, all reachable markings need to be generated and it is not possible to generate reachable markings for a YAWL specification with infinite state space. Therefore, we propose a weaker property called the weak soundness property that describes the minimal requirements for the soundness property and that can be used for a YAWL specification with an infinite state space.

**Definition 2 (Weak soundness).** *Let $N$ be an eYAWL-net and $M_i, M_o$ be the initial and end markings. $N$ satisfies the weak soundness property iff: 1)* ***weak option to complete:*** *$M_o$ is coverable from $M_i$, and 2)* ***proper completion:*** *there is no marking $M$ coverable from $M_i$ such that $M > M_o$, and 3)****no dead transitions:*** *for every task $t \in T$, there is a marking $M$ coverable from $M_i$ such that $t$ is enabled at $M$.*

Reducible elements in the cancellation region of a task represent elements that can never be active and therefore, can never be cancelled by the task. A net has the irreducible cancellation regions property if all elements in the cancellation regions are necessary and cannot be reduced.

**Definition 3 (Irreducible cancellation regions).** *Let $N$ be an eYAWL-net. $N$ has a reducible element $x$, if there is a task $t$ such that $x \in rem(t)$ and $x$ can never be cancelled when $t$ is being executed. $N$ satisfies the* ***irreducible cancellation regions*** *property iff for all $x \in ran\ (rem)$, $x$ is not a reducible cancellation element.*

Non-local OR-join semantics in YAWL results in expensive runtime analysis. It is therefore desirable to determine in advance whether a more appropriate join structure could be found for a task modelled as an OR-join in a YAWL net.

**Definition 4 (Immutable OR-joins).** *Let $N$ be an eYAWL-net and $t$ be an OR-join task in $N$. OR-join task $t$ is convertible to an XOR-join if only one condition in the input set of $t$ is always marked in the enabling markings of $t$ or to an AND-join if all conditions in the input set of $t$ are always marked in the enabling markings of $t$. $N$ satisfies the* ***immutable OR-joins*** *property iff for all $t \in T$, $join(t) = OR$ implies that $t$ is not a convertible OR-join.*

In this section, we have presented the definitions of four structural properties for YAWL workflows. For verification purposes, YAWL specifications are divided into those with OR-joins and those without OR-joins. This distinction is necessary as a different verification technique is needed in each case. In the next two sections, we briefly describe how to detect these properties for YAWL nets with and without OR-joins.

## 3 Verifying YAWL nets without OR-joins

We propose to transform an eYAWL-net (without OR-joins) into an RWF-net (a subclass of reset nets) to exploit the analysis techniques available for reset nets. This is achieved by first abstracting from multiple instances and hierarchy in YAWL and then applying the *transE2WF* function to transform an eYAWL-net into an RWF-net [12]. Figure 2 shows the RWF-net corresponding to the YAWL net in Figure 1. We have formulated the three criteria of the weak soundness property for an RWF-net using the notion of coverability. As coverability is decidable for a reset net using backwards firing rule [5–8], the three criteria of the weak soundness property are decidable. The **Coverable** procedure described in [12] is used to determine whether a marking is coverable

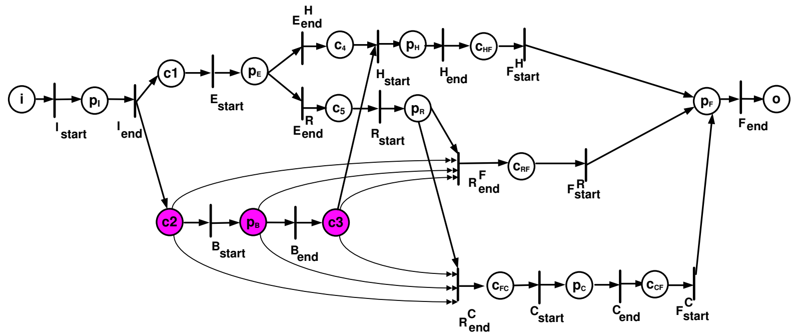


**Fig. 2.** Holiday scenario - RWF-net (Double-headed reset arcs from $c2$, $c3$ and $p_B$ to $R^F_{end}$ and $R^C_{end}$)

from the initial marking in a reset net. We exploit these results to propose an algorithmic approach for deciding the weak soundness property and the irreducible cancellation regions property of an eYAWL-net without OR-joins.

**Observation 1 (Weak soundness is decidable)** *Given an eYAWL-net without OR-joins, 1) the weak option to complete can be decided by testing whether $M_o$ is coverable from $M_i$ in the corresponding RWF-net, 2) proper completion can be decided by testing whether $o + p$ is not coverable from $M_i$ in the corresponding RWF-net for all $p \in P$, and 3) no dead transitions can be decided by testing whether $p_t$ is coverable from $M_i$ in the corresponding RWF-net for all $t \in T$.*

**Observation 2 (Irreducible cancellation regions is decidable)** *Given an eYAWL-net without OR-joins, 1) where a condition $c$ is reducible in a cancellation region of $t$ can be decided by testing whether $c + p_t$ is coverable from $M_i$ in the corresponding RWF-net, and 2) where a task $tx$ is reducible in a cancellation region of $t$ can be decided by testing whether $p_{tx} + p_t$ is coverable from $M_i$ in the corresponding RWF-net.*

As reachability is not decidable for reset nets [6] and its applicability is limited to reset nets with finite state space. As the soundness property definition relies on reachability results, the soundness property is only decidable for an RWF-net with a finite state space. For an eYAWL-net without OR-joins with a finite state space, it is possible to decide the soundness property by generating a reachability graph for the corresponding RWF-net.

**Observation 3 (Soundness is decidable)** *Given an eYAWL-net without OR-joins and a finite reachability graph, the soundness property can be decided by testing the three criteria on the corresponding RWF-netthrough its reachability graph.*

## 4 Verifying YAWL nets with OR-joins

Due to the non-local semantics of an OR-join [12], a net with OR-joins cannot be mapped directly onto a reset net. Hence, we propose to translate all OR-joins into XOR-joins first. The treatment of OR-joins in the YAWL net as XOR-joins is considered optimistic as it assumes an OR-join can be enabled if there is at least one token in its preset. After replacing all OR-joins with XOR-joins, it is now possible to transform the YAWL net into an RWF-net using the *transE2WF* function.

**Observation 4** *Given an eYAWL-net $N$ with OR-joins, let $N'$ be the corresponding eYAWL-net without OR-joins where all OR-joins in $N$ have been replaced by XOR-joins and $RN$ be the equivalent RWF-net for $N'$. The following holds: 1) if $RN$ does not have weak option to complete then $N$ does not have weak option to complete, 2) if $RN$ has dead transitions then $N$ has dead transitions, and 3) if $RN$ has proper completion, then $N$ has proper completion.*

For a YAWL net with OR-joins that has a finite state space, we propose to create a reachability graph by taking into account OR-join semantics and using enabling and firing rules as defined in [3, 12].

**Observation 5** *Given an eYAWL-net with OR-joins and a finite reachability graph, soundness, irreducible cancellation regions and immutable OR-joins are decidable.*

## 5 Verification in YAWL

We have extended the YAWL editor to support the verification approach presented in this paper. The holiday scenario as modelled in Figure 1 satisfies both weak soundness and soundness properties. Figure 3 describes a slightly modified version that have neither the weak soundness nor the soundness property. There are two differences: $c3$ is not in the cancellation region of *Resit exam*, and *Cancel flight* is now an AND-join task. Consider the case where the student has failed the exam and has to resit, after booking the flights. The way this process is now modelled, it is possible for task *Finalise Plans* to be executed, without performing task *Cancel Flight* first. A token is left in condition $c3$ when a token is put into the output condition $o$ which signals the end of the process. Therefore, the model does not satisfy the proper completion criterion. This example highlights how subtle differences in modelling business processes can adversely affect the correctness of a YAWL specification.

## 6 Conclusion

We have proposed four structural properties for workflows with cancellation regions and OR-joins together with new verification techniques based on reset nets and reachability analysis. The only other approach for YAWL verification can be found in [10]. The proposed approach transforms YAWL nets into Petri nets with inhibitor arcs to decide the relaxed soundness property. The use of inhibitor arcs instead of reset arcs means that this approach cannot detect problems in certain specifications with cancellation features. For example, this approach cannot detect problems in the erroneous holiday scenario described in Figure 3. On the other hand, approximation of OR-join semantics enables the verification of nets with OR-joins using invariants.

**Acknowledgements.** We like to thank Lindsay Bradford and Lachlan Aldred for their assistance with the integration of the verification techniques in the YAWL editor.

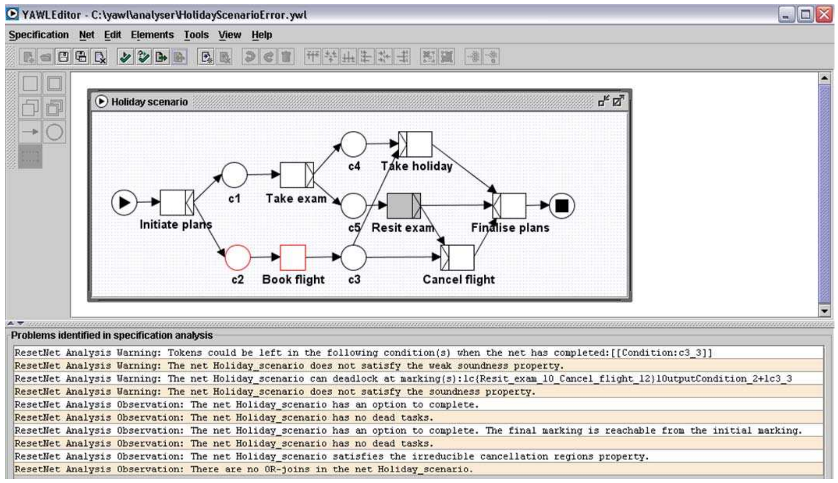


**Fig. 3.** Holiday Scenario with errors

## References


1. W.M.P van der Aalst. Verification of workflow nets. *Proceedings of Application and Theory of Petri Nets*, volume 1248 of *LNCS*, pages 407–426, 1997. Springer-Verlag.
2. W.M.P. van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.
3. W.M.P. van der Aalst and A.H.M. ter Hofstede. YAWL: Yet Another Workflow Language. *Information Systems*, 30(4):245–275, June 2005.
4. W.M.P van der Aalst, A.H.M. ter Hofstede, B.Kiepuszewski, and A.P.Barros. Workflow Patterns. *Distributed and Parallel Databases*, 14:5–51, 2003.
5. C. Dufourd, A. Finkel, and Ph. Schnoebelen. Reset Nets Between Decidability and Undecidability. *Proceedings of the 25th International Colloquium on Automata, Languages and Programming*, volume 1443 of *LNCS*, pages 103–115, 1998. Springer-Verlag.
6. C. Dufourd, P. Jančar, and Ph. Schnoebelen. Boundedness of Reset P/T Nets. *Lectures on Concurrency and Petri Nets*, volume 1644 of *LNCS*, pages 301–310, 1999. Springer-Verlag.
7. A. Finkel, J.-F. Raskin, M. Samuelides, and L. van Begin. Monotonic Extensions of Petri Nets: Forward and Backward Search Revisited. *Electronic Notes in Theoretical Computer Science*, 68(6):1–22, 2002.
8. A. Finkel and Ph. Schnoebelen. Well-structured Transition Systems everywhere! *Theoretical Computer Science*, 256(1–2):63–92, 2001.
9. H.M.W. Verbeek. *Verification of WF-nets*. PhD thesis, Eindhoven University of Technology, The Netherlands, June 2004.
10. H.M.W. Verbeek, W.M.P. van der Aalst, and A.H.M ter Hofstede. Verifying Workflows with Cancellation Regions and OR-joins: An Approach Based on Invariants. Technical Report WP 156, Eindhoven University of Technology, The Netherlands, 2006.
11. M.T. Wynn, W.M.P. van der Aalst, A.H.M. ter Hofstede, and D. Edmond. Verifying workflows with Cancellation Regions and OR-joins: An Approach Based on Reset nets and Reachability Analysis (Revised version). Technical report BPM-06-16, bpmcenter.org, 2006.
12. M.T. Wynn, D. Edmond, W.M.P. van der Aalst, and A.H.M. ter Hofstede. Achieving a General, Formal and Decidable Approach to the OR-join in Workflow using Reset nets. *Proceedings of Application and Theory of Petri nets*, volume 3536 of *LNCS*, pages 423–443, 2005. Springer-Verlag.