# Verification of Multiple Input/Multiple Output Business Processes

Costin Bădică
University of Craiova
Faculty of Automation, Computers and Electronics
Software Engineering Department
Lapus 5, Craiova, RO-1100, Romania
c_badica@hotmail.com

Chris Fox
University of Essex
Computer Science Department
Colchester CO4 3SQ, United Kingdom
foxcj@essex.ac.uk

## Abstract

*In many business process modelling situations using Petri nets, the resulting model does not have a single input place and a single output place. Therefore, the correctness of the model cannot be assessed within the existing frameworks, which are devised for workflow nets — a particular class of Petri nets with a single input place and a single output place. Moreover, the existing approaches for tackling this problem are rather simplistic and they do not work even for simple examples. This paper shows that, by an appropriate reduction of a multiple input/multiple output Petri net, it is possible to use the existing techniques to check the correctness of the original process. The approach is demonstrated with an appropriate example.*

## 1 Overview

This paper introduces a new generalized notion of soundness for Petri nets modelling certain kinds of business processes. There are many kinds of processes. They may be classified according to various properties, for example: the activities, and/or their inputs (outputs) may be continuous or discrete; they may be with or without loops; they may have different numbers of inputs and outputs; and each input (output) may be activated either at most once, or several times.

Processes with single inputs, single outputs, with single activation, are known as *workflows*. They might be used to model an order request, or a bespoke manufacturing process, for example. Their properties have been quite well analyzed, at least when modelled by *WF-nets* [1]. In particular, there is a notion of *soundness* for WF-nets [1].

Clearly, not all processes are workflow processes. We are interested in generalizing the notion of soundness for WF-nets so that can be applied to processes with multiple-inputs and multiple-outputs. Rather than considering all such processes, we focus on instantiations of processes where the inputs are activated at most once, and where we expect outputs to be activated at most once.

Given a particular pattern of activation on the inputs, we wish to determine whether the process, or some part of it, can be interpreted as a sound WF-net. To this end, we consider various ways in which these processes can be mapped directly to WF-nets.

In some cases, it is conceivable that such processes could be modelled directly as workflows. However, we are particulary interested in analyzing given process models that have been expresses as multiple input/output processes in, for example, IDEF0-IDEF3 [7]. Arguably, remodelling them as workflows amounts to a re-engineering activity. The suggested notion of soundness may then assist in correctly reformulating such processes as workflows.

## 2 Introduction

The last few years have shown an increase in the interest of applying information technology to business process management. In our opinion this state of affairs is caused by two factors: i) the need of a better integration between business processes and information systems; ii) the high complexity of the organizations and their underlying business processes. Consequently, formal modelling should be an essential activity for a better management of business processes.

The spread of a special class of software systems called workflow systems can be seen as a tentative solution for the first problem. Therefore, one of the goals of business process modelling is the implementation of a workflow system for enacting, controlling and coordinating its constituent activities.

Business processes are very complex discrete event dynamic systems involving active and passive participants, activities and goals. Because it has been claimed that any discrete event system can be modelled using a Petri net, it follows that Petri nets are also suitable for modelling business processes.

These facts are probably an explanation why in the last few years Petri nets have became an important technique for the modelling and analysis of business processes and workflows. This subject received a lot of attention in the literature, highlighting many arguments in support or against the use of Petri nets for business and workflow process modelling ([2],[4]).

## 3 Related work

The seminal paper [1] introduced WF-nets — a class of Petri nets for workflow modelling. WF-nets take into account an essential feature of workflows; they are case-based. Therefore, the WF-net for a particular workflow has a unique input place (i.e. a place with no in-coming arcs) and a unique output place (i.e. a place with no out-coming arcs). The assignment of a token to the input place indicates that the case may start, and the assignment of a token to the output place indicates that the case has ended. WF-nets have proven to be a useful tool for workflow analysis and special properties have been devised for them in order to asses the correctness of the underlying workflows ([1], [5], [9]).

However, modelling business processes with a Petri net frequently results in a model which does not have a single input place and a single output place. Such situations occur naturally when modelling business processes that are driven by some input data or objects. The modelling can be done either straight into Petri nets or using a high level formalism like Event Driven Process Chains (EPC hereafter, [3], [5]) or Hybrid IDEF0-IDEF3 ([7]) that is then mapped into Petri nets.

Typical examples of this problem are the business processes of the service industry. A case starts when a client submits a request for a service to be delivered. Usually the request is input via a form that asks the client to fill-in the request attributes. For example a request for a holiday package reservation might ask the client to fill-in a hotel reservation, a car reservation, a train ticket a.o. The client does not have to fill-in all these attributes in order for the service to be delivered, but filling in some attributes triggers some corresponding process activities. Moreover, some request attributes might be exclusive. Examples are the values of the delivery option for a book purchased from an e-shop.

A common practise when modelling with Petri nets is to interpret tokens as process objects that are consumed or produced by the process ([8]). From this it follows that multiple input and multiple output (MIMO hereafter) Petri nets naturally occur when modelling processes that are driven by the presence of the input data. Inputs correspond to places with an empty set of input transitions and outputs correspond to places with an empty set of output transitions.

MIMO processes are also obtained when the modelling is done with EPC ([5]) or Hybrid IDEF0-IDEF3 ([7]). In [5] it is shown that a business process modelled with the EPC notation may have MIMO events. Moreover, [6] shows that there is no simple and general way to transform a MIMO Petri net obtained from such an EPC model into a WF-net. A similar situation occurs when translating a Hybrid IDEF0-IDEF3 model to a Petri net ([7]).

In this paper we define Generalized Workflow Nets (GWF-nets), a generalization of WF-nets to the MIMO case. Then we show how to extract one or more WF-nets from a GWF-net. The idea is to consider that the original Petri net model describes a superposition of process definitions, instead a single process definition. Then, starting from a given pattern of input data availability we extract a sub-net from the original GWF-net such that it can then be easily translated into a WF-net by adding a unique input place and a unique output place. One advantage of this approach is that the resulting WF-net is useful for checking the correctness of the underlying process definition using the techniques for WF-nets analysis.

The paper is structured as follows. In section 4 we introduce a motivating example and we show that it cannot be approached satisfactory using existing techniques. In section 5 we define GWF-nets, and show that defining the correctness of GWF-nets can be reduced to the case of ordinary WF-nets. Section 6 concludes the paper and points to future work. We assume that the reader is familiar with the basic definitions and notations of Petri nets, as introduced for example in [10], chapter 2.

## 4 A Motivating Example

Consider the example of a hypothetical business process for material acquisition described in ([7]). This process takes material requests and produces purchase orders and payment authorizations. This process contains a sub-process for handling the material requests that takes material requests and produces validated requests. The company has a list of available authorized suppliers, but it must be prepared to find and handle new potential suppliers. Thus, there is an additional input to model new supplier requirements and an additional output to produce new supplier packages (see figure 1).

Double vertical lines indicate XOR connectors and single vertical lines indicate AND connectors. Activity $a_2$ may take $t_1$ or $t_3$ and may produce either $o_1$ or $t_2$. Activity $a_4$ takes both $o_1$ and $i_2$ to produce $o_2$. More details about this example, the modelling language and the mapping to Petri nets can be found in [7].

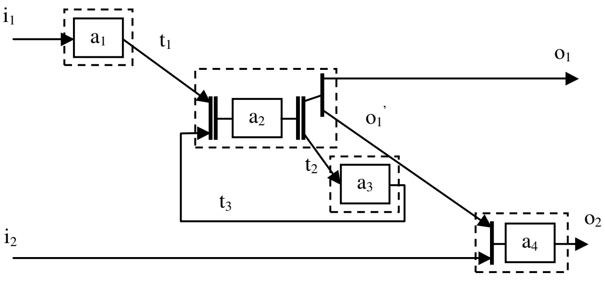


**Figure 1.** A business process for handling material requests, modelled using Hybrid IDEF0-IDEF3

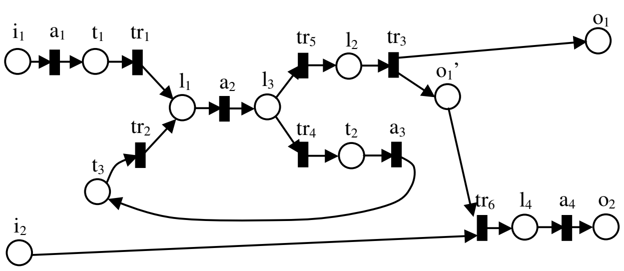


**Figure 2.** A business process for handling material requests, modelled as a Petri net

The result of translating the process from figure 1 into a Petri net is shown in figure 2. The translation mapped the flows in the original model to places $i_1, i_2, o_1, o'_1, o_2, t_1, t_2$ and $t_3$, and the activities in the original model to transitions $a_1$, $a_2$, $a_3$ and $a_4$. The mapping of connectors produces additional places $l_1$, $l_2$, $l_3$ and $l_4$, and transitions $tr_1$, $tr_2$, $tr_3, tr_4, tr_5$ and $tr_6$. The meanings of places and transitions corresponding to flows and activities in the original model are shown in table 1.

Intuitively, the process for handling material requests should be executed in two situations: when both inputs are provided or when only the material request input is provided. In the first case, the process ends by producing both a validated request and a new supplier package, while in the second case it ends by producing just a validated request. Note that according to these intuitions, the process behaves correctly in both cases.

In what follows, let us see what happens when we are checking the correctness of this process by applying the soundness theory of WF-nets. In order to make the presentation self-contained, we first provide the definitions of WF-nets, and the appropriate soundness property.

**Definition 1 (WF-net)** *A Petri net* $N = (P, T, F)$ *is a* WF-net *or* workflow net *if and only if:*

| Name | Description |
|---|---|
| $a_1$ | Log Material Request |
| $a_2$ | Validate Material Request |
| $a_3$ | Resolve Request Problems |
| $a_4$ | Develop New Supplier Specification |
| $i_1$ | Material Request |
| $i_2$ | New Supplier Requirement |
| $o_1, o'_1$ | Validated Request |
| $o_2$ | New Supplier Package |
| $t_1$ | Logged Request |
| $t_2$ | Request Errors |
| $t_3$ | Request Updates |

**Table 1. Places and transitions of the Petri net from figure 2**

*i) There is one source place* $i \in P$ *with no in-coming arcs, i.e.* $\bullet i = \emptyset$.

*ii) There is one sink place* $o \in P$ *with no out-coming arcs, i.e.* $o\bullet = \emptyset$.

*iii) Each node* $n \in P \cup T$ *is on a path from* $i$ *to* $o$.

Paper [1] defines the soundness property as a minimal correctness requirement for a workflow modelled as a WF-net. Intuitively, a workflow is sound if: it always has the option to complete a case; there are no residual tokens; there are neither deadlocks or livelocks; and there are no dead transitions.

**Definition 2 (soundness of a WF-net)** *A process modelled by a WF-net* $N = (P, T, F)$ *is* sound *if and only if:*

*i) For each state* $M$ *reachable from the initial state* $i$ *there exists a firing sequence leading from state* $M$ *to state* $o$*, i.e.* $\forall M\ (i \xrightarrow{*} M) \Rightarrow (M \xrightarrow{*} o)$.

*ii) State* $o$ *is the only state reachable from state* $i$ *with at least one token in place* $o$*, i.e.* $\forall M\ (i \xrightarrow{*} M) \wedge (M \geq o) \Rightarrow (M = o)$.

*iii) There are no dead transitions in* $(N, i)$*, i.e.* $\forall t \exists M, M'$ *s.t.* $i \xrightarrow{*} M \xrightarrow{t} M'$.

Because the Petri net shown in figure 2 is not a WF-net, we cannot check the soundness property in a straightforward way.

In the literature ([3]) it is suggested that such a net can be easily extended with an initialization and/or a termination part such that the first two requirements of definition 1 are satisfied. However, no explicit indication of how to achieve this is given in [3]. This problem is signalled again

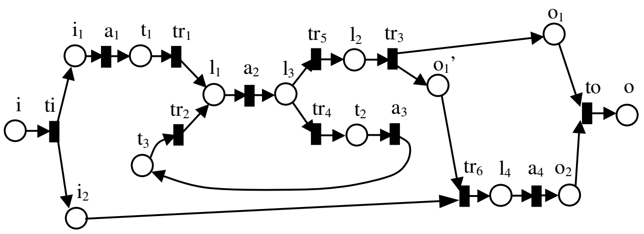


**Figure 3.** A WF-net obtained from the Petri net shown in figure 2 — first solution

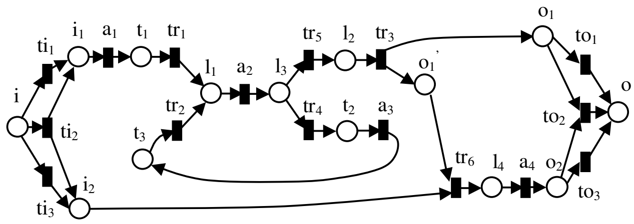


**Figure 4.** A WF-net obtained from the Petri net shown in figure 2 — second solution

in [5] and a solution is also outlined there. The author of the latter paper suggests adding a new start place and a new sink place. These are then connected to Petri net modules which initialize or clean up the places representing the inputs and the outputs of the original net in the right way. In [6], page 42, the same author states that this connection is not trivial, but depends on the relation of the corresponding inputs, and outputs, respectively, in the original net. One way to determine this relation is to track the paths starting from the different inputs or outputs until they join. The connection of the new place with the primary places would then be a Petri net module that corresponds to the connector complementing the one that was found ([6])[1].

Applying this procedure to the Petri net in figure 2, we obtain the WF-net shown in figure 3. It is easy to see that this WF-net is sound. Note, however, that the behavior for the case when only input $i_1$ is provided is not captured by this net, although it was considered correct according to our intuitions. Indeed, in the event the process started and transition $ti$ fired, both $i_1$ and $i_2$ will be marked, indicating that both the material request and the new supplier requirement were provided.

According to [6], the general case may be more difficult because there could be more than two different inputs or outputs with paths possibly meeting in various connectors of different type. Therefore, the suggestion is to link different start places and end places by Petri net modules corresponding to an OR-split or to an OR-join. Using this technique we obtain the WF-net from figure 4.

However, it is not difficult to see that the WF-net shown in figure 4 is not sound. Consider for example the following sequence of transitions: $i \stackrel{ti_1}{\to} i_1 \stackrel{a_1}{\to} t_1 \stackrel{tr_1}{\to} l_1 \stackrel{a_2}{\to} l_3 \stackrel{tr_5}{\to} l_2 \stackrel{tr_3}{\to} o_1 + o'_1 \stackrel{to_1}{\to} o + o'_1$. This shows that when $o$ is marked there is a residual token in $o'_1$.

This example highlights the fact that the existing techniques for applying the soundness criteria for WF-nets to the MIMO case are not satisfactory and, therefore, that more investigation is needed. Some new results are shown in the next section.

[1]This construction was originally suggested for modelling with EPC, but can be naturally extended for modelling with the Hybrid IDEF0-IDEF3 notation introduced in [7]

## 5 Generalized WF-nets

The concept of WF-net can be generalized to the MIMO case as follows.

**Definition 3 (GWF-net)** *A Petri net* $N = (P, T, F)$ *is an* $(m, n)$ GWF-net *or* generalized workflow net *if and only if:*

*i) There are* $m$ *source places* $i_1, \ldots, i_m \in P$ *with no in-coming arcs, i.e.* $\bullet i_k = \emptyset$ *for all* $1 \leq k \leq m$.

*ii) There are* $n$ *sink places* $o_1, \ldots, o_n \in P$ *with no out-coming arcs, i.e.* $o_l \bullet = \emptyset$ *for all* $1 \leq l \leq n$.

*iii) Each node* $x \in P \cup T$ *is on a path from* $i_k$ *to* $o_l$ *for some k and l such that* $1 \leq k \leq m$ *and* $1 \leq l \leq n$.

If $N$ is an $(m, n)$ GWF-net, we define $in(N) = \{i_1, \ldots, i_m\}$ and $out(N) = \{o_1, \ldots, o_n\}$. Note that the Petri net from figure 2 is a $(2, 2)$ GWF-net with $in(N) = \{i_1, i_2\}$ and $out(N) = \{o_1, o_2\}$.

The following proposition states some syntactic properties of GWF-nets and shows how we can attach a WF-net to a given GWF-net. It follows in a straightforward way from definitions 1 and 3.

**Proposition 1 (properties of GWF-nets)** *Let* $N = (P, T, F)$ *be a Petri net.*

*i) If* $N$ *is a GWF-net then* $in(N)$ *are the only source places of* $N$.

*ii) If* $N$ *is a GWF-net then* $out(N)$ *are the only sink places of* $N$.

*iii) If $N$ is a GWF-net then the net $N' = (P', T', F')$, defined by $P' = P \cup \{i, o\}$, $T' = T \cup \{ti, to\}$, $F' = P \cup \{(ti, i_k) | i_k \in in(N)\} \cup \{(i, ti)\} \cup \{(o_p, to) | o_p \in out(N)\} \cup \{(to, o)\}$, is a WF-net. $N'$ is called the* WF-net attached to $N$.

In order to define a correctness criterion for a GWF-net we must consider pairs composed of a pattern of provided inputs and a pattern of produced outputs.

**Definition 4 (input/output patterns)** *Let $N = (P, T, F)$ be a GWF-net. An* input-output pattern *is a pair of sets of places $(I, O)$ such that $I \subseteq in(N)$ and $O \subseteq out(N)$. $I$ is called an* input pattern *and $O$ is called an* output pattern.

Note that an input pattern $I$ induces a marking $M_I$ defined as follows:

$$M_I(p) = \begin{cases} 1, & p \in I \\ 0, & p \notin I \end{cases}$$

Remember that our analysis was focused on what happens when not all the inputs are provided to a GWF-net, and when, therefore, we cannot proceed by checking the soundness of the WF-net attached to the original GWF-net. Moreover, missing input data (i.e. places in $in(N) \setminus I$ that do not contain tokens) will generate dead transitions in $(N, M_I)$ (i.e. transitions of $N$ that are not enabled in any marking reachable from $M_I$). The intuition behind dead transitions is that they represent work that will never be executed if the inputs in $in(N) \setminus I$ were not provided, and therefore they should be discarded by the reduction process.

Fortunately, we can reduce the GWF-net by removing dead transitions in $(N, M_I)$, and other related nodes, resulting either in an empty net or in a smaller GWF-net. The soundness property can then be applied to the WF-net corresponding to the reduced net. The policy for node reduction was devised in order to ensure that the resulting net satisfies the connectivity conditions (iii) in definitions 1 and 3.

**Definition 5 (reducing a GWF-net)** *Let $N = (P, T, F)$ be a GWF-net and let $I$ be an input pattern. Let $T_D$ be the set of all dead transitions of $(N, M_I)$. Let $X = \{x \in P \cup T | x$ is on a path from a node in $in(N)$ to a node in $out(N)$ that does not contain any nodes from $T_D\}$. The net $N' = (P', T', F')$ defined by $P' = P \cap X$, $T' = T \cap X$, $F' = F \cap ((P' \times T') \cup (T' \times P'))$ is called the* reduction of $N$ with respect to $I$ *and it is denoted by $red(N, I)$.*

The net obtained by reducing the original GWF-net plays an important role in our analysis. It has the nice property that it is also a GWF-net, and its input places are among the elements of the input pattern that has been used for reduction.

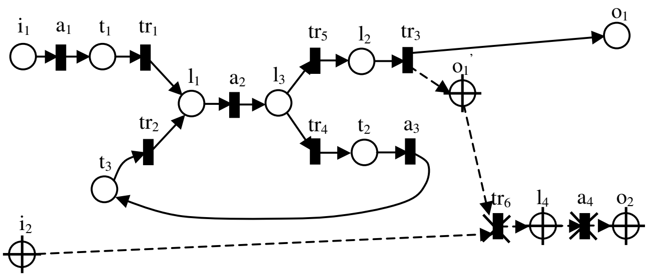


**Figure 5.** Reducing the net shown in figure 2 with respect to $\{i_1\}$

**Proposition 2 (reducing a GWF-net yields a GWF-net)** *Let $N = (P, T, F)$ be a GWF-net, $I$ an input pattern and let $N' = red(N, I)$. If $P' \cup T' \neq \emptyset$ then $N'$ is a GWF-net with $in(N') \subseteq I$.*

We are now ready to define a correctness criterion for GWF-nets. It is called *MIMO soundness* and its definition is based on the classical soundness property defined for WF-nets.

**Definition 6 (MIMO soundness)** *Let $N = (P, T, F)$ be a GWF-net, $I$ an input pattern and let $N' = red(N, I)$ such that $P' \cup T' \neq \emptyset$. The triple $(in(N'), out(N'), N)$ is called* MIMO sound *if and only if the WF-net attached to $N'$ is sound.*

Let us apply the theory developed in this section to the net $N$ from figure 2 and the input pattern $I = \{i_1\}$. The reduction process is detailed in figure 5. Transitions marked with an 'x' are dead transitions, nodes marked with a '+' are removed according to definition 5. The arcs drawn with dotted lines are also removed according to definition 5. Note that the WF-net attached to $red(N, I)$ is sound. This shows that the triple $(\{i_1\}, \{o_1\}, N)$ is MIMO sound.

## 6 Conclusions

In this paper we have shown that the soundness theory developed for WF-nets can also be successfully applied to MIMO business processes modelled as GWF-nets. The key point is the reduction of the original GWF-net to an appropriate sub-net that can be checked within the existing soundness framework. Interesting problems for future investigation are: (1) decidability/complexity analysis of the reduction process; (2) development of algorithms for computing all the input/output patterns for which a MIMO business process has the property of MIMO soundness and (3) searching for conditions that guarantee the MIMO soundness property. We also intend to experiment with our technique on more and significantly larger process models, as soon as such models become available.

## References


[1] Aalst, W.M.P. van der: Verification of Workflow Nets. In: Azéma, P., Balbo, G. (eds.): Application and Theory of Petri Nets 1997, 18th International Conference, ICATPN '97, Toulouse, France, June 23–27, 1997, Proceedings, Lecture Notes in Computer Science, Vol. 1248, Springer-Verlag (1997) pp.407–426.

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

[3] Aalst, W.M.P. van der: Formalization and Verification of Event-driven Process Chains, Comp.Sc.Rep. 98/01, Eindhoven University of Technology, (1998).

[4] Eshuis, R., Wieringa, R.: Comparing Petri net and activity diagram variants for workflow modelling - a quest for reactive Petri nets. In: Ehrig, H., Reisig, W., Rozenberg, G., Weber, H. (eds.): Petri Net Technology for Communication Based Systems, Lecture Notes in Comp. Sc., Vol. 2472, Springer-Verlag, (2002) pp.321–351.

[5] Dehnert, J., Rittgen, P.: Relaxed Soundness of Business Processes, Proc.CAISE'01, (2001).

[6] Dehnert, J.: A Methodology for Workflow Modeling. From business process modeling towards sound workflow specification, PhD Thesis D83, Tech.Univ.Berlin, (2003).

[7] Bădică, C., Bădică, A., Liţoiu, V.: A New Formal IDEF-based Modelling of Business Processes, In: Manolopoulos, Y., Spirakis, P. (eds.): Proc. of the $1^{st}$ Balkan Conference in Informatics, Thessaloniki, Greece, (2003), pp.535–549.

[8] Uthman, C.V.: Improving the Use of Petri Nets for Business Process Modeling, (Draft Status Paper 99/09/22), (1999).

[9] Hee, K.M. van, Sidorova, N., Voorhoeve, M.: Soundness and Separability of Workflow Nets in the Stepwise Refinement Approach. In: Aalst, W.M.P. van der, Best, E. (eds.): Lecture Notes in Comp. Sc., Vol. 2679, Springer-Verlag (2003) pp.337–356.

[10] Desel, J., Esparza, J.: Free Choice Petri Nets, Cambridge Tracts in Theoretical Computer Science, 40, Cambridge University Press (1995).