Information Systems 47 (2015) 197–219

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

The relationship between workflow graphs and free-choice workflow nets

Cédric Favre [a], Dirk Fahland [b], Hagen Völzer [a,*]

[a] IBM Research, Zurich, Switzerland
[b] Eindhoven University of Technology, The Netherlands

ARTICLE INFO

Available online 27 December 2013

Keywords:
Workflow graphs
Petri nets
Free choice
Inclusive Or-join

ABSTRACT

Workflow graphs represent the main control-flow constructs of industrial process modeling languages such as BPMN, EPC and UML activity diagrams, whereas free-choice workflow nets are a well understood class of Petri nets that possesses many efficient analysis techniques. In this paper, we provide new results on the translation between workflow graphs and free-choice workflow nets.

We distinguish workflow graphs with and without inclusive Or-logic. For workflow graphs without inclusive logic, we show that workflow graphs and free-choice workflow nets are essentially the same thing. More precisely, each workflow graph and each free-choice workflow net can be brought into an equivalent normal form such that the normal forms are, in some sense, isomorphic. This result gives rise to a translation from arbitrary free-choice workflow nets to workflow graphs.

For workflow graphs with inclusive logic, we provide various techniques to replace inclusive Or-joins by subgraphs without inclusive logic, thus giving rise to translations from workflow graphs to free-choice nets. Additionally, we characterize the applicability of these replacements. Finally, we also display a simple workflow graph with an inclusive Or-join, which, in some sense, cannot be replaced. This shows a limitation of translating inclusive logic into free-choice nets and illustrates also a difficulty of translating inclusive logic into general Petri nets.

1. Introduction

Different BPM tools, execution engines, and scientific analysis techniques are based on different modeling languages for business processes. This generates a general interest in translating models from one language into another. In particular, business processes are in practice often modeled in industrial languages such as BPMN, EPCs, and UML activity diagrams whereas many analysis techniques, such as control-flow analysis, cost estimation, performance analysis, and process mining are based on Petri nets.

A particular appealing and well understood class of Petri nets are free-choice workflow nets. While they are expressive enough to model the most important control-flow patterns, they rule out some behavioral patterns that are often undesired, such as race conditions, where some choices in a process may become dependent on the ordering of concurrent events. These restrictions make free-choice Petri nets easier to understand and analyze (cf. discussion in [1]). In fact, for free-choice Petri nets, various analysis problems can be solved in polynomial time, which are NP-hard for general Petri nets [2–4].

The gap between industrial languages and Petri nets so far has been bridged in one direction, viz. by translating

* Corresponding author.
E-mail addresses: ced@zurich.ibm.com (C. Favre), d.fahland@tue.nl (D. Fahland), hvo@zurich.ibm.com (H. Völzer).

0306-4379/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.is.2013.12.004

a model in an industrial language to a Petri net e.g., [2,5]. However, such a translation becomes less effective if the input model contains not only exclusive but also inclusive alternative branching. Existing translations to Petri nets make the inclusive logic explicit, which causes an exponential blow-up in the Petri net, which in turn affects running times of analysis techniques.

Furthermore, results of algorithms working on Petri nets currently cannot be translated easily to an industrial language for the lack of a well-understood translation mechanism from Petri nets to industrial languages.

By extending existing translation mechanisms to also translate inclusive branching without exponential blow-up, and to translate from Petri nets to industrial languages, a number of interesting use cases could be enabled. Results of various Petri-net based techniques such as process discovery or process model repair could be easily translated to industrial languages. Process analysis techniques such as verification, simulation, or performance analysis could be applicable to a larger class of industrial languages. Techniques for relating different process models to each other e.g., process model comparison, alignment or querying from a repository, could become easier to apply to process models of different meta-models.

We formally study the problem as the relation between *workflow graphs* and Petri nets. The main control flow of a BPMN, EPC or UML activity diagram can be captured as a workflow graph. A workflow graph may contain exclusive or inclusive alternative branching as well as parallel branching of control flow. In this paper, we present new results on the translation between workflow graphs and Petri nets, in particular free-choice Petri nets.

The requirements of a translation between a workflow graph and a Petri net can vary for different use cases. To obtain general, yet useful results, we take the following requirements into account:

- A model and its translation must have *equivalent* behavior. Many notions of equivalence exist [6]. The adequacy of an equivalence for the translation depends on the use case. We will present the equivalences we use later in the paper. Note that this requirement may by itself not be challenging. For example, one can easily 'unfold' an acyclic workflow graph into its finite full behavior (i.e., computation tree) and then encode this 'unfolding' as a Petri net. Such a construction would preserve, depending on its precise execution, many popular behavioral equivalences, such as trace equivalence and bisimulation. However, the obtained translation is in general exponentially larger than the original workflow graph. Therefore,
- the size of the translated model must be manageable. An exponential blowup is usually not acceptable. We are not aware of any general translation from all workflow graphs with inclusive logic into Petri nets that preserves the behavior and does not incur an exponential blowup. Furthermore,
- the translation must preserve the structure of the original model as much as possible. This is important if we want to map analysis results between the original model and its translation. For example, in order to return to the user of an analysis technique the results in terms of the original process model or, when monitoring or administrating a process, to understand a trace or a state of the running process in terms of the original process model.

We present the following results. We first consider the simpler case of workflow graphs without inclusive logic. Although it is known that workflow graph without inclusive logic are tightly related to free-choice nets, only a translation from workflow graphs to free-choice nets, but not a reverse translation from free-choice nets to workflow graphs was published so far. We present such a reverse translation. Moreover, we show that workflow graphs and free-choice workflow nets are essentially the same thing. More precisely, each workflow graph and each free-choice workflow net can be brought into an equivalent normal form such that the normal forms are, in some sense, isomorphic. (The workflow graph is isomorphic to the graph of conflict clusters of its corresponding workflow net.) This means that, when being in normal form, the workflow graph representation and the free-choice net representation can be used completely interchangeably in every use case.

In the second part, we study workflow graphs with inclusive logic. The inclusive branching forks or joins a variable set of threads, thereby supporting various workflow patterns [7]. The inclusive Or-join (IOR-join), which has a *non-local* semantics, is difficult to translate to Petri nets because the semantics of a Petri net transition is *local*. That is, the enablement and effect of a transition in a Petri net relate only to its adjacent places—a small part of the state of the Petri net—whereas the enablement of an IOR-join may depend on the entire state of the process model.

We show that, in many cases, the IOR-join can be replaced with free-choice constructs, i.e., with a combination of exclusive and parallel joins. However, we will also display a simple workflow graph in which, in some formal sense, an IOR-join cannot be replaced. This will reveal an intrinsic limitation on the replaceability of IOR-joins and hence the translatability of the workflow graph of general process models into Petri nets. This also suggests that the expressiveness of IOR-joins extends beyond free-choice nets.

The remainder of this paper is structured as follows. In Section 2, we introduce the notions of a workflow net and workflow graph. In Section 3, we present the translation between workflow nets and workflow graphs without inclusive logic. In Section 4, we present our results on the translation of workflow graphs with inclusive logic.

2. Foundations

In this section, we define the necessary fundamental notions, which include workflow nets, workflow graphs, and their semantics.

2.1. Workflow nets

A *Petri net* $N=(P,T,F)$ consists of a finite set P of *places*, a finite set T of *transitions*, $P \cap T = \varnothing$, and *arcs* $F \subseteq (P \times T) \cup (T \times P)$. For any *node* $x \in P \cup T$, we write