Expert Systems with Applications 41 (2014) 904–918

Contents lists available at ScienceDirect

Expert Systems with Applications

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

On verification of nested workflows with extra constraints: From theory to practice

Roman Barták *, Vladimír Rovenský

Charles University in Prague, Faculty of Mathematics and Physics, Malostranské nám. 25, 118 00 Praha 1, Czech Republic

ARTICLE INFO

Keywords:
Workflow
Verification
Complexity
Constraint satisfaction
Temporal consistency

ABSTRACT

Workflows are used to formally describe processes of various types such as business and manufacturing processes. One of the critical tasks of workflow management is automated discovery of possible flaws in the workflow – workflow verification. In this paper, we formalize the problem of workflow verification as the problem of verifying that there exists a feasible process for each task in the workflow. This problem is tractable for nested workflows that are the workflows with a hierarchical structure similar to hierarchical task networks in planning. However, we show that if extra synchronization, precedence, or causal constraints are added to the nested structure, the workflow verification problem becomes NP-complete. We present a workflow verification algorithm for nested workflows with extra constraints that is based on constraint satisfaction techniques and exploits an incremental temporal reasoning algorithm. We then experimentally demonstrate efficiency of the proposed techniques on randomly generated workflows with various structures and sizes. The paper is concluded by notes on exploiting the presented techniques in the application FlowOpt for modeling, optimizing, visualizing, and analyzing production workflows.

1. Introduction

Workflow management is a technology for automating work with processes. It is frequently applied to business processes and project management, but it is useful also for description of manufacturing processes. Briefly speaking, workflow is a formal description of a process typically as a set of interconnected activities. There exist many formal models to describe workflows (van der Aalst & Hofstede, 2005) that include decision points and conditions for process splitting as well as loops to describe repetition of activities. In this paper we adapted the idea of *Nested Temporal Networks with Alternatives* (Nested TNA) (Barták & Čepek, 2008), in particular the form used in the manufacturing scheduling system FlowOpt (Barták et al., 2011). A Temporal Network with Alternatives (TNA) (Barták & Čepek, 2007) is a directed acyclic graph with parallel and alternative splitting and joining of processes that is also known as AND-split and OR-split (and AND-join, OR-join) in traditional workflow management systems (Bae, Bae, Kang, & Kim, 2004; van der Aalst & Hofstede, 2000). Hence a TNA can describe alternative processes in the style similar to scheduling workflows with optional activities introduced by Beck and Fox (2000) and used also in Extended Resource Constrained Project Scheduling Problems (Kuster, Jannach, & Friedrich, 2007). A Nested TNA requires a specific structure of the underlying task graph where the split operations have corresponding join operations. This structure was informally proposed by Beck and Fox (2000) and then formalized by Barták and Čepek (2008). The nested structure is obtained when seeing the workflow as a hierarchical structure of tasks that are decomposed to sub-tasks until the primitive activities are obtained. Fig. 1 gives an example of a nested workflow with three types of decompositions: parallel decomposition (*Legs*), serial decomposition (*Seat*), and alternative decomposition (*Back Support*). The figure also highlights the hierarchical (tree) structure of the nested workflow. Such a structure is quite typical for real-life workflows (Bae et al., 2004) as many workflows are obtained by decompositions of tasks. The structure is also close to the idea of hierarchical task networks leading to Temporal Planning Networks (Kim, Williams, & Abramson, 2001).

In this paper we formally introduce nested workflows. A minor difference from the formal definition of a Nested TNA is that a nested workflow is obtained by decompositions of tasks while a Nested TNA as defined in (Barták & Čepek, 2008) is obtained by decompositions of arcs. The decomposition of arcs makes it easier to see a Nested TNA as a special form of a TNA, while the decomposition of tasks is more natural from the user perspective (Barták et al., 2011). The structural properties of both approaches are identical. In particular, the nested structure guarantees that the workflow is valid in terms of possibility to select for each involved task a

* Corresponding author. Tel.: +420 221 914 242; fax: +420 221 914 323.
E-mail addresses: bartak@ktiml.mff.cuni.cz (R. Barták), v.rovensky@lit.cz (V. Rovenský).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.08.021

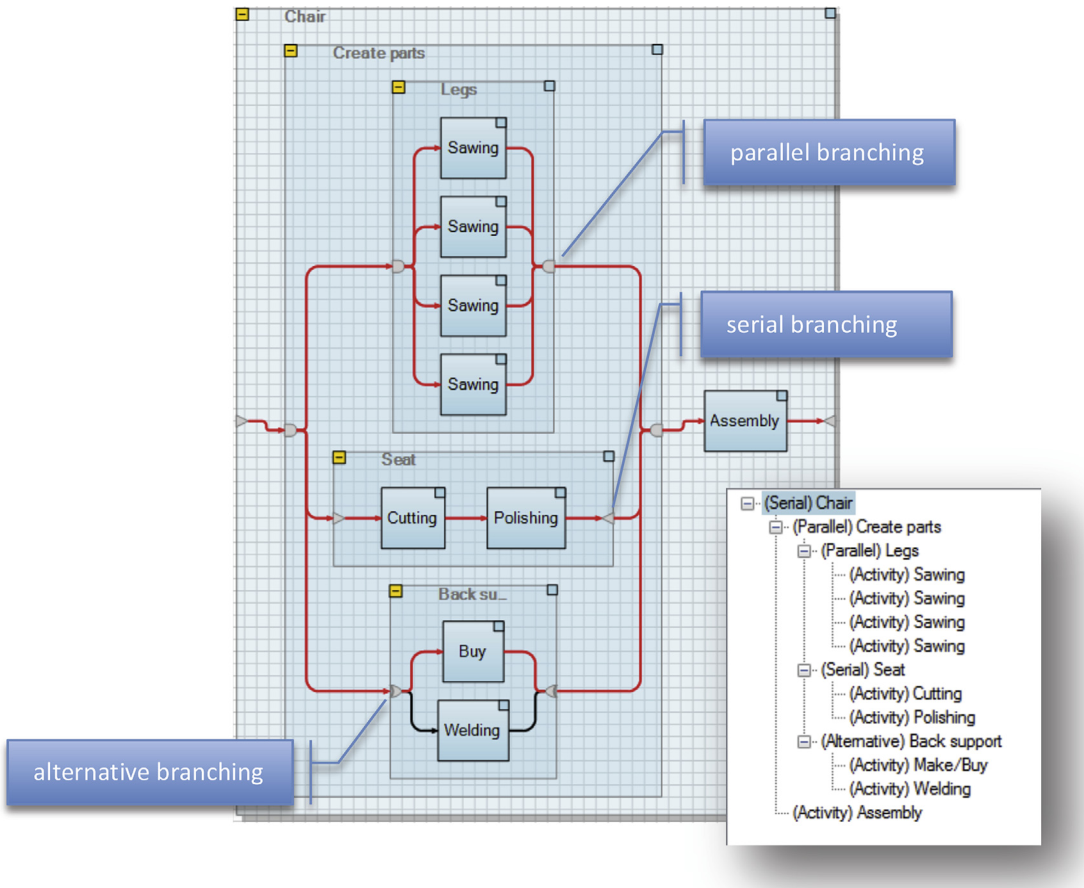

Fig. 1. Nested workflow as it is visualized in the FlowOpt Workflow Editor (from top to down there are parallel, serial, and alternative decompositions).

process that contains the task (Barták & Čepek, 2008). As a consequence, the pure nested workflows do not need to be verified, as they are always valid.

To make the nested workflows more flexible for practical applications, we suggest supporting extra synchronization, precedence, and causal constraints that can be added between any pair of tasks in the workflow (Bae et al., 2004; Kuster et al., 2007). It can be easily shown that adding extra constraints can introduce flaws to the workflow. For example adding a mutex (mutual exclusion) constraint between activities *Buy* and *Cutting* to the workflow in Fig. 1 causes the activity *Buy* to be inapplicable (*Cutting* must be part of any process selected from the workflow). This raises the question if such flaws in the design of a workflow can be detected automatically before the workflow is used for example to generate a manufacturing schedule. This is exactly the task of *workflow verification* as we define it in this paper and it should be an integral part of workflow management systems (Giro, 2007).

Workflow verification has been studied for some time. Various methods of verification have been proposed, e.g., using Petri Nets (van der Aalst & Hofstede, 2000), graph reductions (Sadiq & Orlowska, 2000), or logic-based verification (Bi & Zhao, 2004). These methods deal with complex workflow structures that are used for example to model business processes. Differently from TNAs these workflows may contain cycles and various conditions for selection of alternatives. The focus of verification methods for such workflows is on structural properties of the workflows. These methods are not applicable to extra constraints that go beyond the core workflow structure. The focus of this work is on a novel verification method that covers both the core workflow constraints (its structure) as well as the extra constrains.

We will show that the extra constraints make the problem of selecting a process containing a given task NP-complete (Barták, 2012). Hence the verification problem for such workflows is also NP-complete, as it requires verifying that a feasible process exists for each task. To verify the nested workflows with extra constraints we propose an algorithm based on modeling the verification problem as a set of constraint satisfaction problems (Barták & Rovenský, 2013) and using a specific search procedure employing incremental temporal filtering (Planken, 2008). As an attempt to improve practical efficiency of the verification algorithm we also propose collapsing some tasks in the workflow before the constraint-based algorithm is started. This decreases the size of the search space with a small overhead for identifying which tasks can be collapsed.

The paper summarizes results published in (Barták, 2012) and (Barták & Rovenský, 2013), adds a new proposition about NP-completeness when extra equivalence constraints are used, and provides a more extensive experimental evaluation. The paper is organized as follows. We will first formally define the nested workflows with extra constraints and introduce the notions of a process and a feasible process. Then, we will prove that the decision problem of existence of a feasible process in a nested workflow with extra constraints is NP-complete. In Section 2, we will specify the workflow verification problem and formulate it as a constraint satisfaction problem. After that we will present a verification algorithm that exploits the problem formulation as a constraint satisfaction problem and we will also introduce the conditions under which some tasks can be collapsed. We will then present a detailed experimental characterization of the proposed techniques that shows how these workflow verification techniques work on randomly generated workflows of different structures and size. In