Data & Knowledge Engineering 95 (2015) 39–65

Contents lists available at ScienceDirect

Data & Knowledge Engineering

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

Editorial

Stepwise structural verification of cyclic workflow models with acyclic decomposition and reduction of loops

Yongsun Choi [a,*], Pauline Kongsuwan [b], Cheol Min Joo [c], J. Leon Zhao [d]

[a] Department of Industrial and Management Engineering, Inje University, South Korea
[b] Department of Information and Communication Systems, Inje University, South Korea
[c] Department of Industrial and Management Engineering, Dongseo University, South Korea
[d] Department of Information Systems, City University of Hong Kong, China

ARTICLE INFO

Article history:
Received 30 March 2012
Received in revised form 14 October 2014
Accepted 24 November 2014
Available online 12 December 2014

Keywords:
Workflow management
Methodologies and tools
Structural verification
Acyclic decomposition
Loop reduction

ABSTRACT

Existence of cycles (or loops) is one of the main sources that make the analysis of workflow models difficult. Several approaches of structural verification exist in the literature, but how to verify cyclic workflow models efficiently in a comprehensible form remains an open research question. Thus, a novel structural verification approach for cyclic workflow models by means of acyclic decomposition and reduction of loops is introduced in this paper with the following contributions. First, acyclic decomposition of natural loops, further enhanced by reduction of nested loops, enables existing verification techniques, normally dealing with acyclic models, to handle workflow models with natural loops. Second, instantiation of an irreducible loop into natural loops, altogether with reduction of concurrent loop entries, enables the proposed approach to handle workflow models with irreducible loops. Last, diagnostic information, provided by the proposed approach, helps stakeholders correct and improve their workflow models. Two examples are provided to show that the proposed approach is systematic and practical. In addition, a prototype of the proposed approach is developed. Its execution result shows that, while providing diagnostic information, the proposed approach can handle workflow models with arbitrary cycles effectively.

1. Introduction

Over the last two decades, workflow management systems (WfMSs) have been used in various domains to achieve higher efficiency, agile adaptation, and systematic management of business processes [1–3]. Workflow models should be correctly defined before they are deployed in the WfMSs to avoid runtime errors and costly maintenance delays [3–17]. Empirical studies show that the models from practice have considerable error ratios, i.e. 5–30% having control-flow problems [18–20]. In addition, to correct and further improve the models, communication among different stakeholders is required [4,11,15,16]. Vanhatalo et al. [15] suggested two criteria of acceptable verification methodologies to eliminate enactment errors: (i) not to delay the process of constructing a model (to make it possible to integrate control-flow analysis tightly with modeling tools), and (ii) to produce useful diagnostic information in order to locate and fix errors in the model. In summary, it is essential to ensure the correctness of the workflow models in a prompt and comprehensible way before deployment.

Control-flow verification, or structural correctness, has been mainly focused in the study of workflow analysis [3,5–7,10–15,17,21]. Further analysis with respect to logical, temporal, and performance aspects, for instance, can be applied after structural conflicts are removed [13,16,22]. However, although workflow verification is important, few commercial WfMSs provide formal verification tools,

* Corresponding author at: Department of Industrial and Management Engineering, Inje University 197 Inje-ro Gimhae, GyeongNam 621-749, Republic of Korea.
E-mail addresses: yschoi@inje.edu (Y. Choi), p_kongsuwan@yahoo.com (P. Kongsuwan), cmjoo@dongseo.ac.kr (C.M. Joo), jlzhao@cityu.edu.hk (J.L. Zhao).

http://dx.doi.org/10.1016/j.datak.2014.11.003
0169-023X/

i.e. most commercial WfMSs use vendor-specific ad-hoc modeling techniques without theoretical framework for their representation, analysis, and manipulation [4,7,9]. Process simulation, supported by some WfMSs, may be used to find a useful insight into process behavior, but it cannot handle interrelationship among process components [9].

Another issue in the control-flow verification is existence of loops [20,23–26]. In a workflow model, some activities may need to be re-executed until a certain condition is met. This requirement forms a structure called *loop*, which is classified into *reducible loop* and *irreducible loop* by a number of loop entries. Control-flow instances of a loop can be various since activities could be executed multiple times, possibly in different paths in each cycle. An irreducible loop having multiple entries, its control-flow instance is further varied by which loop entries are activated at the first visit and during cycling. Consequently, it makes the loop difficult to analyze. The problem is more complex if the loop is inside or contains another loop. Additionally, arbitrary cycles in particular may cause ambiguity when executed in the context of some modeling languages, as shown in [27] for example. To avoid such ambiguity, some WfMSs strictly restrict the syntax of their modeling languages. As a result, the expressive power of their languages is reduced [28].

To handle aforementioned issues, this paper proposes a stepwise structural verification approach for workflow models with arbitrary cycles, by means of acyclic decomposition and loop reduction. Major features of the proposed approach are as follows:

- *Acyclic decomposition of natural loops, further enhanced by reduction of nested loops* enables the proposed approach to handle workflow models having natural loops by using existing verification approaches that can only deal with acyclic workflow models;
- *Instantiation of an irreducible loop into several natural loops, altogether with reduction of concurrent loop entries* enables the proposed approach to handle workflow models having irreducible loops; and
- *Providing diagnostic information* helps process designers and process analysts correct or further improve the structure of their workflow models.

The rest of the paper is organized as follows. Section 2 provides a review of existing workflow verification approaches by focusing on directed graphs and control-flow perspective. Section 3 presents background information on workflow graphs and concepts related to loops, which are well known in control-flow analysis and graph theory [18,23,29]. Section 4 fully describes the proposed approach by dividing into three functions: compound-gateway splitting, natural-loop handling, and irreducible-loop handling. Section 5 shows how the proposed approach works in detail through two examples. Section 6 discusses computational complexity of the proposed approach and execution results derived from its prototype system, a Windows-based application named *gProAnalyzer*. Finally, Section 7 summarizes contributions of this paper and provides suggestions on future research.

2. Related work

In the literature of workflow modeling and analysis, various modeling languages and verification approaches exist. Good modeling languages should have clear semantics so that analysis techniques (based on their own languages) can be developed. An example of such languages is Petri net [6,30]. It is a graphical language, which can support the primitives identified by Workflow Management Coalition (WfMC) [31]. Its semantics is formally defined. Its analysis techniques and tools are also abundant. However, its description of a real process tends to be complex and extremely large [6], and possibly induce state-explosion problem [32]. To reduce reachable states during the verification, partial-order reduction, reduction rules, and model-slicing algorithm, for example, are added [30,33–37]. Approaches transforming a process modeled in other languages to Petri net for the sake of verification have also been proposed [17,20,34,38–42]; but, two-way mapping issues of model equivalence and reflection of verification results in the original models might occur [28].

Another formal technique widely used for the analysis of state space is model checking [43]. Its state space is a directed graph, which represents the control flow of a process model and can be constructed by breadth-first or depth-first search (from the graph theory). It verifies correctness of the graph (e.g. safety and liveness properties) by using a model checker (e.g. Spin and NuSMV) together with temporal logic (e.g. LTL and CTL) or propositional logic, providing both syntax and semantics for the constraints. Techniques such as modeling slicing and decomposition are applied to reduce state space [44,45]. However, a recent approach by Bartak and Rovensky [45], for example, still report only one error (if found) per model to avoid generating the entire state space.

Since the control flow of a process can be represented as a directed graph, graph-based approaches have been proposed for structural verification [11,12,14,15,21,46,47]. Sadiq and Orlowska [14] proposed five rules to reduce an acyclic directed graph iteratively until the graph is empty or no rules can be applied. If the final graph is empty, it means that the graph has no error; otherwise, the graph contains at least one structural conflict. Lin et al. [21] criticized the previous approach for its incompleteness and proposed two additional rules. However, both approaches are limited to acyclic models and they were criticized by Liu and Kumar [48] that they cannot provide causes of detected errors and help for further improvement of the model. Choi and Zhao [11] proposed an approach combining loop decomposition and inline-block reduction with pattern-based verification rules for nested cyclic directed graphs. This research was criticized by Bi and Zhao [10] for not providing scalability, usability, and computational efficiency of the prototype system.

Zerguini [46] proposed reduction algorithm to transform a free-choice WF-net (a subclass of Petri net) to a hierarchical net based on reducible regions (defined in the graph theory) without changing the behavior of the original net. Soundness of each region and the final net are verified separately by using Petri-net-based techniques. Hauser et al. [12] proposed an approach imposing hierarchical structure to a cyclic workflow graph by using region-growing rules, adapted from reduction rules in [14,21]. The model is semantically sound if and only if it is region-reducible. This approach can detect and localize structural conflicts, can support overlapped structure, but cannot support synchronized conditions. Vanhatalo et al. [15] proposed a decomposition method based on single-entry and single-exit (SESE) regions; and combined reduction rules and heuristic rules to speed up the verification process of a workflow