Reliability Engineering and System Safety 145 (2016) 351–365

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

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

Restructuring of workflows to minimise errors via stochastic model checking: An automated evolutionary approach

L.T. Herbert [a], Z.N.L. Hansen [b,*]

[a] The Technical University of Denmark, DTU Compute, 2800 Lyngby, Denmark
[b] The Technical University of Denmark, DTU Management Engineering, 2800 Lyngby, Denmark

ARTICLE INFO

Available online 15 July 2015

Keywords:
Reliability analysis and risk assessment methods
Safety assessment software tools
Consequence modelling and management
Enterprise risk management
Safety management and decision making

ABSTRACT

This paper presents a framework for the automated restructuring of stochastic workflows to reduce the impact of faults. The framework allows for the modelling of workflows by means of a formalised subset of the BPMN workflow language. We extend this modelling formalism to describe faults and incorporate an intention preserving stochastic semantics able to model both probabilistic- and non-deterministic behaviour.

Stochastic model checking techniques are employed to generate the state-space of a given workflow. Possible improvements obtained by restructuring are measured by employing the framework's capacity for tracking real-valued quantities associated with states and transitions of the workflow. The space of possible restructurings of a workflow is explored by means of an evolutionary algorithm, where the goals for improvement are defined in terms of optimising quantities, typically employed to model resources, associated with a workflow.

The approach is fully automated and only the modelling of the production workflows, potential faults and the expression of the goals require manual input. We present the design of a software tool implementing this framework and explore the practical utility of this approach through an industrial case study in which the risk of production failures and their impact are reduced by restructuring the workflow.

1. Introduction

Today, a wide range of industries employ documented workflows and seek to ensure their efficiency and safety [5]. There is substantial evidence [6–8] that being able to determine safety and efficiency properties of a workflow and their exact quantitative values early in the design phase can result in smoother integration, accurate provisioning to meet service level agreements, and cost savings.

For the purpose of this paper a production *workflow* is understood to be a well defined sequence of production steps such as cutting, forming or moulding a product, or conducting a quality control test. In this paper, *goals* for these workflows are to meet or exceed targets with regard to quantitative goals such as efficiency, cost, hygiene or similar and qualitative goals such as ensuring that a product has been through certain production steps in a specific order. The design of such processes is today a predominantly manual activity, in which process maps are drawn and analysed by hand, and improved configurations are found by a process of trial and error. This approach takes considerable effort and arriving at an optimal practice is frequently costly and time consuming. There is therefore a need for a more efficient approach.

1.1. Related work

In recent years the *business process modelling and notation* (BPMN) language [1] has emerged as the dominant notation for the description of workflows, especially at the level of domain analysis and high-level systems design [9]. While being a widely used standard in practice BPMN appears not to be implementable due to the numerous details of its execution semantics which are expressed with insufficient precision [1]. Indeed the BPMN standard explicitly states that "*the execution semantics are described informally (textually), and this is based on prior research involving the formalisation of execution semantics using mathematical formalisms.*" [1, p. 445]. The shortcomings of these semantics have been

Abbreviation: BPD, business process diagram [1]; BPMN, business process model and notation [1]; COWS, calculus for orchestration of web services [2]; CSP, communicating sequential processes [3]; PCTL, probabilistic computation tree logic [4]; SBOAT, stochastic BPMN optimisation and analysis tool

* Corresponding author. Tel.: +45 45 25 44 40; fax: +45 45 93 34 35.
E-mail address: znlh@dtu.dk (Z.N.L. Hansen).

http://dx.doi.org/10.1016/j.ress.2015.07.002
0951-8320/

observed, for all BPMN versions, by numerous authors [10–16] (this list is not exhaustive).

Fundamentally BPMN lacks a general notion of state and as a consequence, the specification of relevant data-dependent execution conditions are only poorly supported. For example, data objects, which are associated with activities or sequence flows, are used only informally, in particular with underspecified assumptions on the input/output selection at task nodes or sequence flows. This additionally makes any sort of quantitative formal analysis of standard BPMN impossible. However, the standardisation of BPMN has lead to its widespread use by business practitioners and consequently there have been a number of attempts to provide a formalisation of its semantics.

Wong and Gibbons [17] take an abstraction refinement approach to the formalisation of a subset of BPMN models as communicating sequential processes (CSP) [3] process-algebraic expressions. Their work expresses the syntax of BPMN in the *Z* notation [18]. They proceed to define a semantic function which takes a syntactic description of a BPMN diagram and returns a CSP process that models the behaviour of that diagram as the parallel composition of CSP processes corresponding to the states of the diagram. The semantics they impose abstracts the internal flow of individual task states, and models the sequence of task initialisations and terminations within a workflow. This work presents a thorough approach to the formal translation of BPMN to CSP. However, the specific semantic interpretation forced upon BPMN models and the resulting CSP models produced are very circuitous. While they are theoretically amenable to formal analysis, in practice the verbosity of the generated CSP models makes it computationally expensive to perform analysis. Probabilistic properties and rewards are not supported, and while variants of CSP exist that support these features, considerable reworking of their approach would be needed to account for these.

Analysis of BPMN models extended with stochastic properties has seen limited development with only two approaches identified for dealing with models which exhibit both probabilistic and non-deterministic transitions. Prandi et al. [2] have identified the model checking tool PRISM [4] as ideally suited to the analysis of stochastic PRISM workflows. This effort involves conversion of PRISM models into a model expressed in the COWS [19], which in turn is converted into a model that can be analysed using PRISM. This approach adds unnecessary complexity in that it is possible to convert the notation of BPMN directly into the PRISM modelling language, and then allow PRISM to impose a semantic interpretation without the additional semantic restrictions of going via COWS. Further, the translations from PRISM to COWS and in particular from COWS to PRISM are loosely defined and, in the form described by the authors, not amenable to algorithmic translation. This approach does allow the use of rewards. Consequently, the PRISM model checker is potentially able to perform analysis of both quantitative and stochastic properties of a workflow. However, details of how such properties would be included in the source BPMN models is not described.

1.2. Contribution

This paper develops a framework which allows for the automated restructuring of workflows so as to minimise the impact of errors on, and improve the performance of, a production workflow. This framework consists of

1. A workflows modelling language constructed as formalisation and extension of the *business process modelling and notation* (BPMN) language (Section 2).
2. Algorithms for formal verification, fault-tree generation and automated improvement of workflows (Sections 3 and 4).
3. An efficient scalable software implementation (Section 5).

The central contribution of this paper is an tractable environment and evolutionary algorithm that explores the space of possible improved workflows. Central to this is the determination of the expected mean values of quantities of interest at points of failure by means of stochastic model checking. This allows for the expression of temporal state-space queries identifying the errors in workflows execution and their associated impact on production, typically measured in cost and/or time.

A software tool, Stochastic BPMN Optimisation and Analysis Tool (SBOAT), is presented which implements our approach. Using this we explore the practical utility of this work by means of an industrial case study which focuses on a manufacturing workflow. The case company is a manufacturer of baked goods based in Denmark. We discuss both the degree of improvement achieved by use of this evolutionary approach and the current limitations of this approach.

2. Modelling workflows

We present a framework which uses an extended version of the BPMN language to automatically restructure of workflows so as to minimise the impact of errors on a production workflow. This modelling language is intended to a visual language allowing for the modelling workflows in a business context by a business analyst. This section describes the theoretical underpinnings of the language and the software implementation of the presented framework allows for model construction through a simple drag-and-drop editor.

2.1. Core BPMN

For the purposes of this paper a subset of BPMN [1] is chosen based on elements which are most commonly used in the full BPMN language, alongside considerations of which elements in the full language can be decomposed into combinations of simpler elements [20]. Often called the *core* subset of BPMN, it consists of the eight elements which are found to be most commonly used in a large survey of real-world BPMN usage [10]; it should be noted that more than 70% of models surveyed consisted only of these elements. The graphical elements of core BPMN are shown in Fig. 1 and described below.

The process of modelling a workflow in BPMN involves composing a number of BPMN elements into a business process diagram (BPD). The intention is that a business process diagram captures the complete workflow of a business process, with separate sub-components of a workflow organised into separate pools which communicate via message passing.

Definition 1. (Core BPD). A core BPD is an extended process graph tuple $BPD = (\mathbf{N}, \mathcal{F}, \mathbf{P}, \text{pool}, \mathbf{L}, \text{lab})$ where $\mathbf{N} \subseteq \mathbf{T} \cup \mathbf{E} \cup \mathbf{G}$ is a set of nodes composed of the following disjoint sets:

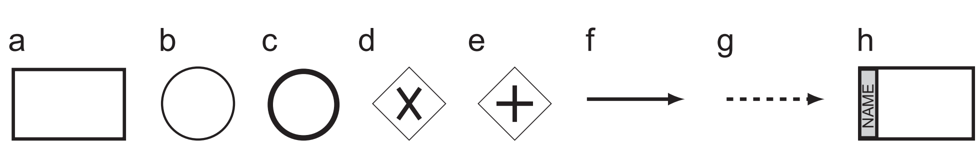

Fig. 1. Core BPMN elements (a) Task, (b) Start, (c) End, (d) Decision Gateway, (e) Parallel Gateway, (f) Sequence Flow, (g) Message Flow and (h) Pool.