Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 64 (2015) 643 – 650

Procedia Computer Science

Conference on ENTERprise Information Systems / International Conference on Project MANagement / Conference on Health and Social Care Information Systems and Technologies, CENTERIS / ProjMAN / HCist 2015 October 7-9, 2015

Reliability of BPMN Business Processes

Ana Respício[ab], Dulce Domingos[ac*]

[a]Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, Campo Grande,1749-016 Lisboa, Portugal
[b]Centro de Matemática, Aplicações Fundamentais e Investigação Operacional, Faculdade de Ciências, Universidade de Lisboa, Campo Grande,1749-016 Lisboa, Portugal
[c]Lasige, Faculdade de Ciências, Universidade de Lisboa, Campo Grande,1749-016 Lisboa, Portugal

Abstract

Organizations increasingly use business processes and the Business Process Model and Notation (BPMN) to model them. Taking into account the relevance of Quality of Service (QoS) aspects in business processes, we can find in the literature some proposals that already calculate reliability of structured workflows and service compositions, as well as some proposals that extend BPMN to include cost, availability, and reliability, among others QoS aspects.
In this paper we focus on reliability and we calculate the reliability of the overall BPMN process starting from the reliability value of its activities. We use the Stochastic Workflow Reduction method, which applies a set of reductions rules to process blocks. To accomplish this, we extend BPMN with reliability information and we identify the BPMN process blocks for which we can apply a reduction. We apply our approach to a use case concerning a simplified paper reviewing process. In addition, we identify the limitations of our proposal, which are intrinsic to the BPMN non-block structure. To the best of our knowledge, this is the first proposal to compute the reliability of BPMN business processes.

Peer-review under responsibility of SciKA - Association for Promotion and Dissemination of Scientific Knowledge

Keywords: BPMN, reliability, business process, QoS

* Corresponding author. Tel.: +351 217 500 524; fax: +351 217 500 084
E-mail address: mddomingos@fc.ul.pt

Peer-review under responsibility of SciKA - Association for Promotion and Dissemination of Scientific Knowledge
doi:10.1016/j.procs.2015.08.578

1. Introduction

Nowadays, organizations increasingly use business processes for capturing, managing, and optimizing their activities. In Portugal, according to the results of the Business Process Management (BPM) Observatory survey, 83% of 371 biggest Portuguese organizations have some kind of BPM related procedures [7]. Since its release in 2004, BPMN (Business Process Model and Notation) [13] has become the de-facto business process modelling language standard [6]. Despite BPMN2.0 not comprising Quality of Service (QoS) aspects, we can find some works that extend it with reliability, performance, cost, and availability among others QoS aspects [1, 3, 9, 10, 16]. Considering reliability, there are some works on reliability calculation of structured workflows [2] and service compositions [4, 5, 11]. However, BPMN is a more high level business process modelling language and semantically more powerful, even when compared to WS-BPEL (Web Services Business Process Execution Language) [12].

In this paper, we propose to compute BPMN business processes reliability by using the Stochastic Workflow Reduction method [2]. This method applies a set of reduction rules to the process, iteratively, until only one activity remains. The reliability of the remaining activity corresponds to the reliability of the process. To meet this goal, we also propose to extend BPMN with reliability information and we identify the BPMN process blocks for which we can apply reduction rules.

Reliability calculation enables process analysis to ensure that user requirements are met. Process reliability information is used at design time to analyse alternatives, as well as at run time to select participants, execute services, or monitor process executions.

This paper is organized as follows: in the next section we review related work. Section 3 describes the application of a set of reduction rules to BPMN process blocks to determine process reliability, and Section 4 presents a use case scenario. Finally, Section 5 concludes the paper and discusses future work.

2. Related Work

According to Koren and Krishna [8], the *reliability of a system at time t*, denoted by $R(t)$, is the probability that the system has been up continuously in time interval $[0, t]$. This metric is adequate for systems operating continuously for which a single momentary failure can have a high or even critical impact.

In the context of workflow modelling, Cardoso [2] defines *task reliability* as the probability that the components operate on users demand, following a discrete-time model. In this context, the failure rate of a task can be described by the ratio number of *unsuccessful executions*/ *scheduled executions*. The task reliability, denoted by $R(A)$, is the opposite of the failure rate, that is $R(A) = 1 - failureRate(A)$.

Traditionally, reliability has been a major concern for networking, real-time applications, and middleware [15]. Due to its fundamental role in all kind of systems, reliability has also been approached within other contexts such as workflows. Cardoso [2] proposes a predictive QoS model for workflows and Web Services (WS) that, based on atomic task QoS attributes, is able to estimate the QoS for workflows, considering the following dimensions: time, cost, reliability, and fidelity. To compute QoS for the overall workflow, the author developed the Stochastic Workflow Reduction algorithm, which relies on reduction rules to iteratively reduce construction workflow blocks until the complete workflow is reduced to a single atomic task. When this happens, the remaining task encloses the QoS metrics corresponding to the workflow under analysis. Cardoso uses reduction rules for the following construction blocks: sequential, parallel, conditional, loop, fault tolerant, and network systems. He applies his proposal to the METEOR workflow management system. In [13], Coppolino et al. generalize the Cardoso proposal, covering all the generic workflow patterns of Wohed et al. [19], to estimate the reliability of WS compositions.

Mukherjee et al. compute the reliability of WS-BPEL processes taking into account most of the workflow patterns that WS-BPEL can express [11], while Distefano et al. method also incorporates advanced composition features such as fault, compensation, termination and event handling [5].

Considering BPMN, we can find some proposals that extend it with QoS information, such as reliability, performance, cost, and availability, among others. Meyer et al. extend BPMN to model certainty of information provided by the sensor devices (from 0 to 100%) and the availability/potential fault of these devices [10]. Chiu and Wang include availability and fault tolerance rates [3]. Considering performance requirements, Caracas and

Bernauer use the category element of the BPMN message to distinguish transmission modes, such as broadcast and unicast, as well as communication protocols, such as IEEE 802.15.4 and TCP/IP [1]. Sungur et al. use performance annotations to prioritize between reliability and energy consumption of sensor devices [16]. Still in the area of business processes that use sensor devices, in [9] the authors extend BPMN with quality of information and access cost. However, none of them use the reliability of activities to compute the reliability of the overall BPMN process, which is the focus of our proposal.

3. Determining process reliability

We determine the business process reliability value by using the Stochastic Workflow Reduction (SWR) method [2]. This method applies a set of reductions rules to the workflow, iteratively, until only one activity remains. The reliability of the remaining activity corresponds to the reliability of the process. We use the six reduction rules of Cardoso [2]: (1) sequential, (2) parallel, (3) conditional, (4) loop, (5) fault-tolerant, and (6) network. Before we can apply these reduction rules, we extend BPMN with reliability information and we identify how BPMN can represent each type of process block that reduction rules use, as described in the following subsections.

3.1. Extending BPMN with reliability information

To compute process reliability, we need to know the reliability of each activity. We enrich BPMN business processes with reliability information by extending the BPMN *Activity* element with the additional *ReliabilityValue* element, which stores the reliability value. Figure 1 presents the XML schema for this extension.

```
<xsd: schema ...>
   ...
   <xsd: group name="reliabilityInfo">
      <xsd: sequence>
         <xsd: element ref=" ReliabilityValue" minOccurs="0" maxOccurs="1" />
      </xsd: sequence>
   </xsd: group>
   ...
</xsd: schema >
```

Figure 1 - XML schema for the BPMN2.0 extension

In addition, we also include the probability value in sequence flows. This value is used in conditional, loop and fault tolerant reduction rules, as we detail in section 3.3. The default probability value for sequence flows is 1.0 [2].

3.2. Identifying BPMN process blocks for reduction rules

BPMN provides more than one option to represent almost all types of process blocks for which there are reduction rules. The following systematization is based on the works of White [17] and Wohed et al. [18].

1) Sequential process block

In BPMN, a sequential process block is defined with a BPMN *Uncontrolled Flow* that connects two activities. A BPMN *Uncontrolled Flow* is a BPMN *Sequence Flow* that does not have any conditional indicator (mini-diamond) or any intervening gateway. Figure 2 illustrates a sequential process block represented in BPMN.

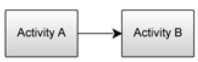

Figure 2 - BPMN pattern for Sequential process block

2) Parallel process block

BPMN has two options to represent concurrency, i.e. multiples activities that are executed in parallel. The first one uses multiple outgoing sequence flows, while the second one uses a parallel gateway, as illustrated in Figure 3.

Figure 3 - BPMN patterns for Parallel process blocks

3) Conditional process block

In BPMN, exclusive gateways represent a branching point where alternatives are based on conditional expressions (cond1, cond2, ..., default) contained within the outgoing sequence flows. Only one of them will be chosen. Exclusive gateways can be shown with or without the "X" marker. In case of a merge gateway, if all the incoming flows are alternatives, then the gateway is not needed, as shown in the left part of Figure 4.

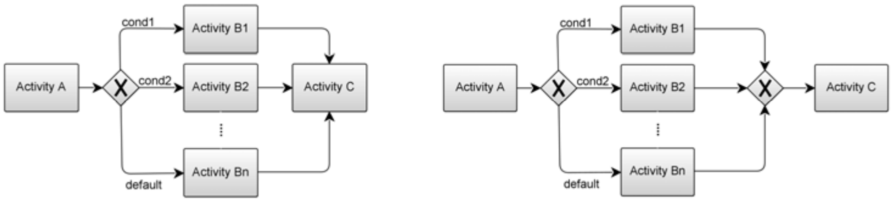

Figure 4- BPMN patterns for Conditional process blocks

4) Loop process block

BPMN represent loops as: (a) Loop Activity - the activity is executed as long as its condition evaluates to true; (b) Parallel Multiple Instances Loop - multiple instances of Activity are executed in parallel; (c) Sequence Multiple Instances Loop - multiple instances of Activity are executed sequentially; and (4) Sequence Flow Looping - loops are created by connecting a Sequence Flow to an "upstream" object. Figure 5 illustrates each of these loop representations from left to right, respectively.

Figure 5 - BPMN patterns for Loop process blocks

5) Fault Tolerant process block

Fault Tolerant process blocks differ from Conditional process blocks in that they allow from one to all of the alternative paths to be chosen. BPMN represents Fault Tolerant process blocks with Inclusive Gateways or Complex Gateways. With Inclusive Gateways, alternatives are based on conditional expressions contained within the outgoing sequence flows, illustrated in the left side of Figure 6.

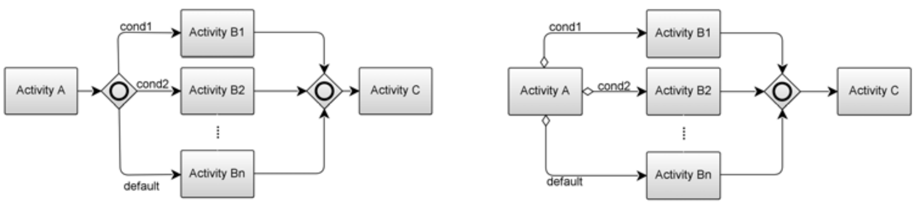

Figure 6 - BPMN patterns for Fault Tolerant process blocks - Inclusive Gateway

Unlike conditional process blocks, each path is independent and, consequently, all combinations of paths may be taken. Alternatively, instead of using an inclusive gateway, it is possible to use conditional Sequence Flows, marked with mini-diamonds, as illustrated in the right side of Figure 6.

BPMN also represents Fault Tolerant process blocks with Complex Gateways. In this case, the modeller defines how many alternative paths are necessary to be executed (*k* out of *n*). Alternative paths can also be represented by conditional flows as illustrated in the right side of Figure 7.

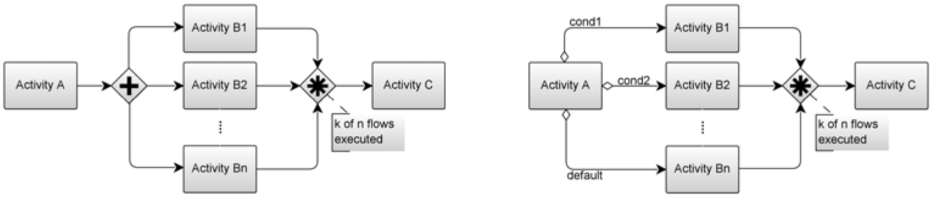

Figure 7 - BPMN patterns for Fault Tolerant process blocks - Complex Gateway

6) Network process block

Network process blocks represent sub-processes. A Sub-process is an Activity that encapsulates a Process, whose internal details have been modelled using BPMN elements.

Figure 8 - BPMN pattern for Network process blocks

3.3. Applying reduction rules to BPMN

In this subsection, we present the result of the reduction rule for each type of process block in the column "Reduced Block" of Table 1, and how we calculate the reliability of the reduced process block in the column "Reliability of Reduced Block" of the same table. For instance, the reliability value of the reduced sequential block, designated as $R(AB)$, is the product of the reliabilities values of Activities A and B. For the parallel block, the reduced activity only executes successfully if all the involved activities succeed. To calculate the reliability value of the reduced conditional block we need to know the probability value of each conditional sequence flow p_i, i.e. the probability p_i that activity Bi is chosen to execute, where $\sum_{1\leq i\leq n} p_i = 1$ and *n* is the number of alternative flows. The reliability of the reduced conditional block is $R(B1n) = \sum_{1\leq i\leq n} p_i R(Bi)$.

Within reduced loop process blocks, where p is the probability of executing the loop, the reliability of the reduction of A' is $R(A') = \frac{(1-p)R(A)}{1 - p\,R(A)}$. An activity A that includes a loop may execute as follows: the loop is not performed, with reliability $(1-p)R(A)$, or the loop is performed once, with reliability $pR(A)(1-p)R(A)$, or the loop performs k times, with reliability $(pR(A))^k(1-p)R(A)$. Thus, $R(A') = \sum_{k\geq 0}(pR(A))^k * (1-p)R(A)$. Summing the terms of the infinite geometric progression, we obtain $R(A') = \frac{1}{1 - p\,R(A)} * (1-p)R(A) = \frac{(1-p)R(A)}{1 - p\,R(A)}$. For loop blocks that execute exactly k times, the reliability of the reduction is $R(A') = R(A)^k$.

Concerning fault tolerant process blocks, the reliability of a reduced block is given by $R(B1n) = \sum_{I_1=0,1} \dots \sum_{I_2=0,1}(\phi(\sum_{i=1}^{n} I_i - k)\prod_{i=1}^{n} 1 - I_i + (2I_i - 1)R(Bi))$. The reliability of a k-out-of-n system is given by the sum of the reliabilities of all the scenarios where at least k activities execute. Variable I_i is used to express whether activity Bi executes or fails, by setting $I_i = 1$ or $I_i = 0$, respectively. We consider the function $\phi(x) = 1, \text{if } x \geq 0; 0,$ otherwise, to determine which combinations of k activities out of n correspond to scenarios where at least k activities execute, and thus should be considered. For activity Bi the reliability value to consider is $1 - I_i + (2I_i - 1)R(Bi)$, which is $R(Bi)$, if Bi executes, and 0, otherwise.

Finally, network process blocks represent sub-processes and the reliability value of the reduced block is set to the reliability value of the sub-process.