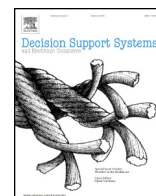




Contents lists available at SciVerse ScienceDirect

Decision Support Systems

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

Detection of naming convention violations in process models for different languages

Henrik Leopold^{a,*}, Rami-Habib Eid-Sabbagh^b, Jan Mendling^{c,e},
Leonardo Guerreiro Azevedo^d, Fernanda Araujo Baião^d^a Humboldt-Universität zu Berlin, Spandauer Straße 1, 10178 Berlin, Germany^b Hasso Plattner Institute, Potsdam, Germany^c Wirtschaftsuniversität Wien, Austria^d Federal University of the State of Rio de Janeiro (UNIRIO), Rio de Janeiro, Brazil^e University of Ljubljana, Slovenia

ARTICLE INFO

Article history:

Received 21 September 2012

Received in revised form 27 June 2013

Accepted 27 June 2013

Available online xxx

Keywords:

Process modeling guidelines

Naming convention checking

Natural language processing

Process model quality

ABSTRACT

Companies increasingly use business process modeling for documenting and redesigning their operations. However, due to the size of such modeling initiatives, they often struggle with the quality assurance of their model collections. While many model properties can already be checked automatically, there is a notable gap of techniques for checking linguistic aspects such as naming conventions of process model elements. In this paper, we address this problem by introducing an automatic technique for detecting violations of naming conventions. This technique is based on text corpora and independent of linguistic resources such as WordNet. Therefore, it can be easily adapted to the broad set of languages for which corpora exist. We demonstrate the applicability of the technique by analyzing nine process model collections from practice, including over 27,000 labels and covering three different languages. The results of the evaluation show that our technique yields stable results and can reliably deal with ambiguous cases. In this way, this paper provides an important contribution to the field of automated quality assurance of conceptual models.

© 2013 Published by Elsevier B.V.

1. Introduction

Nowadays business process modeling plays an important role in many organizations [17]. Business process models are for instance used for documenting business operations or supporting the analysis and design of information systems [43]. The strong uptake of process modeling in industry has lead to huge modeling initiatives which often result in thousands of process models [73]. Such an amount of models raises the question of how to assure the quality of these models. A specific problem here is the fact that many modelers in practice are not sufficiently trained and that a significant share of process model collections from practice contains errors [57].

In order to cope with these challenges organizations introduce modeling guidelines capturing different details on how processes should be modeled. The complexity of such guidelines is often a factor that makes manual enforcement and compliance checking difficult. Several quality aspects can be easily checked in an automated fashion. For instance, formal process properties such as soundness or the absence of deadlocks can be reliably and effectively checked [84,90]. Such automatic analyses are precise and can also be applied

on large process model collections in a short amount of time. However, while many formal and control-flow based aspects are well understood, there is a notable gap for checking linguistic aspects such as naming conventions. This is a serious problem since the importance of text labels for the understanding of a process model has been clearly demonstrated in prior research [26,59]. Hence, an automated solution for checking naming conventions would significantly help to improve the overall quality of process models in practice.

While refactoring of activity labels has been discussed in prior research [53], there are several challenges that hinder the application of automatic label analysis on a broader scale. Label analysis is typically conducted for English process models. Accordingly, any approach for English models can benefit from the rich set of freely available natural language processing tools for the English language, such as the lexical database WordNet for the English language [62]. Although there exist many WordNet solutions for several languages,¹ these tools significantly vary with respect to completeness, quality and availability. For instance, the English WordNet covers 155,287 words, while the German *GermaNet* only covers 93,407 words. In addition, many WordNet solutions do not always cover morphological rules. Hence, a word will be only found in the database if it is given in the base form, e.g. a noun in the singular or a verb in the infinitive. On top of that, most WordNet databases for other languages cannot be freely accessed. As a result, models created

* Corresponding author. Tel.: +49 30 2093 5805.

E-mail addresses: henrik.leopold@wiwi.hu-berlin.de (H. Leopold),
rami.eidsabbagh@hpi.uni-potsdam.de (R.-H. Eid-Sabbagh), jan.mendling@wu.ac.at
 (J. Mendling), azevedo@uniriotec.br (L.G. Azevedo), fernanda.baiao@uniriotec.br
 (F.A. Baião).

¹ http://www.globalwordnet.org/gwa/wordnet_table.html.

with text labels in other important languages such as Portuguese, Spanish, or German cannot be checked with prior solutions.

In this paper we address these problems by introducing a flexible technique for automatically checking naming conventions in process models. This technique does not depend on WordNet and can hence be adapted to other languages. In order to demonstrate the applicability of the technique, we conduct an extensive evaluation with in total nine process model collections from practice. The employed collections contain English, German, and Portuguese labels in order to illustrate the language independence of this technique.

Building on the recommendations from [88,89], the remainder of this paper is structured as follows. Section 2 discusses the background on process modeling guidelines and the potential of natural language for checking linguistic guideline aspects. In addition, we investigate the different label styles of process model elements. Section 3 defines a technique for automatically checking naming conventions in a language independent manner. Section 4 presents the results from the empirical evaluation of the nine process model collections. Finally, Section 5 concludes the paper and gives an outlook on future research.

2. Background

In this section, we discuss the research background. First, we summarize the general idea of establishing process modeling guidelines. Then, we discuss the potential of natural language processing for guideline checking. Finally, we identify different styles for labeling process model elements in different languages.

2.1. Process modeling guidelines

Process modeling guidelines aim at assuring the quality and consistency of process models which are created by different and heterogeneously skilled users. They capture best practices which have proven to be beneficial to support readers in understanding the models [29,76,1]. In their simplest forms, they can be formulated as rules such as “A model must have one start event” or “A model should not make use of complex gateways.” The rationale behind such guidelines is the insight that some modeling practices are easier to understand and hence are superior in terms of clarity.

Guidelines refer to different aspects of process models. Four of the major aspects are formal model properties, model layout, the use of modeling elements and the use of natural language. Thereby, *formal model properties* refer to the correctness of the model structure. It is often suggested to keep a model as structured as possible [58] and to avoid deadlocks [83]. The *model layout* discusses the proper arrangement of the elements. Good layout typically minimizes the number of crossing arcs and utilizes a clear direction of flow either from right to left or from top to bottom [74]. The *usage of modeling elements* dimension defines which elements should or should not be used. Languages like BPMN offer different options to express the same behavior [70], so one option might be preferred. Elements with complicated semantics might be forbidden in order to guarantee a good model understanding also by unexperienced model readers [76]. The *usage of natural language*

can be considered from two perspectives. The first refers to the set of terms which can be used in the model. Some guidelines try to ensure the term consistency by introducing glossaries [67] or forbidden unspecific verbs [74]. The second perspective is concerned with the structure in which these terms are presented in the label. Guidelines usually suggest certain naming conventions as for instance the verb–object style for activities [59].

In order to illustrate these dimensions and to discuss in how far they can be automatically checked, we consider the exemplary BPMN process model from Fig. 1. The process starts with a start event which is then followed by the activity *Cost Planning* and proceeds with an XOR-split gateway. Such a diamond-shaped element defines a decision point such that either the upper or the lower branch of the process will be executed. Accordingly, either the activity *Plan Data Transfer* or the activity *Recalculation of Costs* is conducted. Afterwards, the control is passed to the XOR-join gateway. This diamond-shaped element waits for one of the incoming branches to complete. Once one branch has been executed, the process continues with the activity *Get Approval for Expenses* before the process is terminated. It is quite apparent that this exemplary process model contains some considerable weaknesses which may prevent a reader from understanding it correctly.

Starting with the formal properties perspective, we might want to check whether the process model suffers from structural errors like deadlocks. In fact, such formal issues are well-understood and can be efficiently checked in an automatic way using Petri-net concepts [25]. Further techniques are available for checking the correctness of the data flow [79,87,75] or the satisfiability of resource constraints [9,15,78]. Also automatic techniques for refactoring model structure are available [86,68]. The layout of the model can be discussed in terms of flow direction and crossing arcs. The depicted model appears to be well organized according to these criteria. Poorly arranged models can be laid out using concepts from graph drawing [4], which have been customized for process models [23]. The usage and exclusion of certain model elements is intensively discussed for BPMN [65,76]. We observe that the example model from Fig. 1 only uses a basic set of elements, which is often recommended. Technically, checking the inclusion and exclusion of elements of a certain type is trivial.

If we consider the usage of the natural language in the model we can observe two main problems. The first is concerned with inconsistent terms. While two activities refer to the term *costs* the last activity mentions the term *expenses*. To avoid such inconsistencies an approach for automatically creating and enforcing the usage of glossary terms has been proposed [67]. The second problem is the inconsistent usage of labeling styles. While a verb–object labeling is typically suggested [76,58,1], the example model of Fig. 1 shows different deviations. The activities *Cost Planning* and *Recalculation of Costs* capture the actions *plan* and *recalculate* as nouns at different positions in the label. The activity *Get Approval for Costs* is compliant with the verb–object requirement. This style mix actually causes that the activity *Plan Data Transfer* could be misinterpreted. It could either advice to *plan* a *data transfer* or to *transfer* a record of *plan data*. Also the event and gateway labels can be improved. The gateway label *acceptable?* lacks a reference to a business object, and also the end event label *Approved* does not precisely define the result of the process execution.

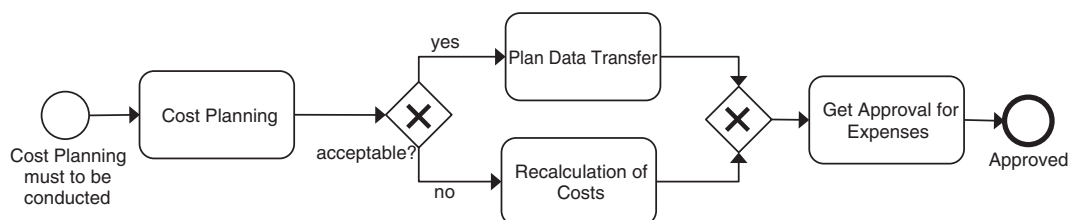


Fig. 1. Example model.

Download English Version:

<https://daneshyari.com/en/article/6948613>

Download Persian Version:

<https://daneshyari.com/article/6948613>

[Daneshyari.com](https://daneshyari.com)