Detecting Role Inconsistencies in Process Models

DETECTING ROLE INCONSISTENCIES IN PROCESS MODELS

Research Paper

Aysolmaz, Banu, Maastricht University, Netherlands
b.aysolmaz@maastrichtuniversity.nl
Iren, Deniz, Center for Actionable Research, Open Universiteit, Heerlen, Netherlands,
deniz.iren@ou.nl
Reijers, Hajo A., Utrecht University, Netherlands
h.a.reijers@uu.nl

Abstract

Business process models capture crucial information about business operations. To overcome the challenge of maintaining process definitions in large process repositories, researchers have suggested methods to discover errors in the functional and the behavioral perspectives of process models. However, there is a gap in the literature on the detection of problems on the organizational perspective of process models, which is critical to manage the resources and the responsibilities within organizations. In this paper, we introduce an approach to automatically detect inconsistencies between activities and roles in process models. Our approach implements natural language processing techniques and enterprise semantics to identify ambiguous, redundant, and missing roles in textual descriptions. We applied our approach on the process model repository of a major telecommunication company. A quantitative evaluation of our approach with 282 real-life activities displayed that this approach can accurately discover role inconsistencies. Practitioners can achieve significant quality improvements in their process model repositories by applying the approach on process models complemented with textual descriptions.

Keywords: business process modeling, inconsistency detection, organizational perspective, natural language processing.

1 Introduction

Process models are used for a variety of purposes in the organizations ranging from communicating process knowledge to analyzing information systems (Davies et al., 2006). To fulfill these purposes, process models capture crucial information on what is to be done, by whom, when, and how in business processes (Curtis, Kellner, and Over, 1992; Weske, 2007). Among these forms of information, the *organizational perspective*, which represents the roles, i.e. actors and systems that perform the activities, plays an important role in achieving organizational purposes, although it has mostly been overlooked in the literature (Aysolmaz and Reijers, 2017). When this perspective is not captured correctly, the benefits of process models for both business side and technical side may be hindered. On the business side, this deficit may lead to problems in understanding responsibilities and managing resources in the organization (Browning, 2010). On the technical side, it may cause workflow systems to improperly allocate resources and rights (Russell et al., 2005; Schefer-Wenzl and Strembeck, 2014).

Process models have been increasingly used in organizations, as shown by the existence of process model repositories containing hundreds of process models (Rosemann, 2006; Weber et al., 2011). Process models are frequently complemented by textual descriptions (Aa et al., 2018). In this way, process complexity is limited and preferences of different stakeholders in the organization are accommodated. However, the existence of a high number of models and the development of process definitions in two different formats, models and text, naturally lead to inconsistencies in process definitions (Aa, Leopold, and Reijers, 2016a). To resolve such problems, many researchers have developed techniques to automatically detect errors and inconsistencies in process models (Mendling et al., 2008; Pittke, Leopold, and Mendling, 2013). More specifically, textual descriptions have been used to reveal inconsistencies in the functional and behavioral aspects of process models, i.e. activities and their ordering relations (Aa, Leopold, and Reijers, 2016a). To date, however, there have been no studies that focus on detecting deficiencies on the organizational aspects of process models based on textual descriptions.

In this paper, we introduce an approach to automatically detect inconsistencies between activities and roles performing those activities in process models and textual activity descriptions. Specifically, our approach identifies three types of inconsistencies about activities and related roles. First, it identifies the activities that lack a clearly specified role in the activity description. Second, the approach detects those activities with missing roles in the process model. Lastly, the approach discovers the activities that are assigned redundant roles. We implement Natural Language Processing (NLP) techniques and organizational semantics to identify the relevant subjects in activity descriptions and linked role elements in the process models. Then, we assess the correspondence of the two sets and discover the inconsistencies between them by using set theory. The approach can be used by organizations to automatically identify the role-related inconsistencies in a collection of process models that are complemented by textual descriptions. It is generalizable to different process modeling notations since notations represent similar organizational process information despite the differences in their symbol sets and rules.

We applied our approach within a major telecommunications company on a set of 36 process models containing 282 activities. The set was composed of five process areas maintained by different units and having diverse characteristics. We developed a prototype tool to implement the approach. We quantitatively evaluated the accuracy of the approach by comparing the identified inconsistencies against a gold standard set, which was manually prepared by a group of domain experts. The findings suggest that our approach may be effectively used by practitioners to enhance the quality of the organizational perspective of their process models by finding out the inconsistency problems.

The remainder of this paper is structured as follows. Section 2 presents a brief overview of the state of the art and illustrates the problem. Section 3 describes the proposed approach in a stepwise manner. In Section 4, we present our evaluation method and results. In Section 5, we discuss the results and highlight their implications. Finally, Section 6 concludes the paper and provides directions for future work.

2 Background

In this section, we discuss the research background. First, in Section 2.1, we present how the organizational perspective of processes is represented in different process modeling notations. Then, in Section 2.2, we describe the use of textual descriptions complementary to process models. Then, we introduce the studies on investigating the inconsistencies and ambiguities in process models and related textual descriptions. Lastly, we illustrate the problem of inconsistencies related to the organizational perspective of process models by using a running example.

2.1 Organizational Perspective in Process Models

Process models are used to capture crucial information on four process perspectives: (1) a *functional perspective* on the activities performed, (2) a *behavioral perspective* on the flow relation between those activities, (3) an *informational perspective* on the artefacts used and produced by these activities, and (4) an *organizational perspective* on the roles that perform the activities (Curtis, Kellner, and Over, 1992). Various process modeling notations have been developed to represent process information including these perspectives. Some of the widely used notations, among others, are Business Process Modeling and Notation (BPMN) (Object Management Group, 2011), Event-driven Process Chains (EPCs) (Scheer, 2000), and UML activity diagrams (Object Management Group, 2017). Despite the use of different symbols for representing process elements, their representational capabilities are mostly comparable (Recker et al., 2009) and their usability for business users are similar (Birkmeier, Klockner, and Overhage, 2010).

The roles in processes can be individual actors, a group of actors, organizational units, or systems. It may be necessary to show multiple roles or a combination of different roles required for performing an activity (Nawrocki et al., 2006). The way roles are represented may differ from one process modeling notation to the other. The BPMN and UML activity diagrams use swimlanes to represent roles, BPMN use horizontal and UML activity diagrams use vertical swimlanes (Dumas et al., 2018). In process models using swimlanes, the whole diagram is structured like a pool having lanes. Each activity is placed in a lane, which represents a specific role or a group of roles that is responsible for performing the activities in that lane. When the EPC notation is used, each role is represented as a symbol. One role or a number of roles related to performing an activity are directly connected to the activity. Therefore, there may be a number of symbols connected to each activity to represent the roles. The EPC notation also supports the use of swimlanes.

In summary, process modeling notations follow different ways to represent the organizational perspective. In practice, there may be even more ways of representing these since organizations tailor the notations for their own use (Recker et al., 2006). Nevertheless, despite the differences in symbols and rules for relating the symbols to each other, it is possible to extract similar role-related information from process models as developed with different notations.

2.2 Process Models and Textual Descriptions

Process models are usually not the sole medium of capturing process knowledge in organizations. Organizations frequently complement their process models with textual descriptions due to two reasons. First, among diverse stakeholders using process models in organizations, some users have difficulties in reading process models or preference for textual information rather than diagrams (Aysolmaz et al., 2018). Second, process models are not suitable for capturing all details of processes (Baier and Mendling, 2013). Process models have found to provide a higher value when they are complemented with textual information (Ottensooser et al., 2012). With the complementary use of process models and textual descriptions, cognitive preferences of diverse users are accommodated (Aa et al., 2018) and a process model's complexity is restrained while understandability is improved (Ferrari et al., 2018).

The organizational perspective is a core chunk of information provided in the textual process descriptions. The textual description of a process typically includes roles, actions performed by those roles, and their interactions (Aa et al., 2018). Roles, i.e. actors or systems, are found to be frequently mentioned in the sentences from real-life textual process descriptions (Ferrari et al., 2018; Soares Silva et al., 2018). Textual descriptions are usually provided separately for each process. In some organizations, descriptions are attached to individual activities. In others, a combined text fragment is provided for the whole process, although they are logically linked to individual activities (Aa et al., 2018). For the latter case, existing NLP approaches can use used to find the correspondance between activities and parts of the text (Aa, Leopold, and Reijers, 2016a).

2.3 Consistency Problems in Process Models and Textual Descriptions

It is a challenging task to maintain the consistency of process information when organizations have large process repositories (Aa, Leopold, and Reijers, 2016a). When process models are developed by different stakeholders and textual descriptions are updated separately throughout the development of the repository, it is highly likely that inconsistencies emerge. Since it is almost unmanageable to identify the errors manually in a large set of process models, automated approaches are needed. There is a body of literature on revealing inconsistency problems among the elements of processes. Textual descriptions, providing complementary information to process models, have been heavily used to reveal such problems (Aa et al., 2015; Aa, Leopold, and Reijers, 2016a; Friedrich, Mendling, and Puhlmann, 2011). For instance, by comparing textual descriptions with process models, missing or inconsistent activity definitions in process models have been discovered (Aa et al., 2018; Aa, Leopold, and Reijers, 2016a). Textual descriptions have also been used to identify ambiguities in the behavioral perspective of processes (Aa, Leopold, and Reijers, 2016b). Activity labels as textual information have been used to detect the deficiencies in the terminology used in process models (Pittke, Leopold, and Mendling, 2013, 2015a), the issues with the scope of the activity such as modeling multiple activities in a single activity element (Pittke, Leopold, and Mendling, 2015b), and incomplete activity names (Pittke et al., 2015). The possible ambiguities in textual process descriptions have also been investigated related to ordering of activities (Soares Silva et al., 2018) and the extent of information covered (Ferrari et al., 2018).

In summary, so far, natural language analysis in process model labels and textual descriptions have been successfully used to automatically reveal inconsistencies in the functional and behavioral perspectives of process models. The studies discussed in this section have shown that it is critical for the success of BPM initiatives to reveal problems in process model repositories. Text descriptions are valuable sources to uncover such problems. However, although the organizational perspective is as important as the functional and behavioral perspectives in process models (Aysolmaz and Reijers, 2017), no approach has been proposed to overcome possible issues in the organizational perspective of process models.

2.4 Problem Illustration

To illustrate the problem, Figure 1 displays a process model, which is adjusted from a real-life process model. This model is expressed in a tailored version of the Business Process Model and Notation (BPMN). The activities are depicted using rounded rectangles with the control-flow relations between them, as can be seen in the top part of the figure. The diamond shapes with a cross inside represent decision points. For the ease of depiction here, we represent individual roles, i.e. role elements, as rectangles and connect them to the activities instead of using swimlanes of the BPMN standard. The model consists of six activities, which are numbered in the top-left corner. At the bottom, textual descriptions related to each activity are presented. The gray shade suggests a relation between a role element and part of the text referring to the

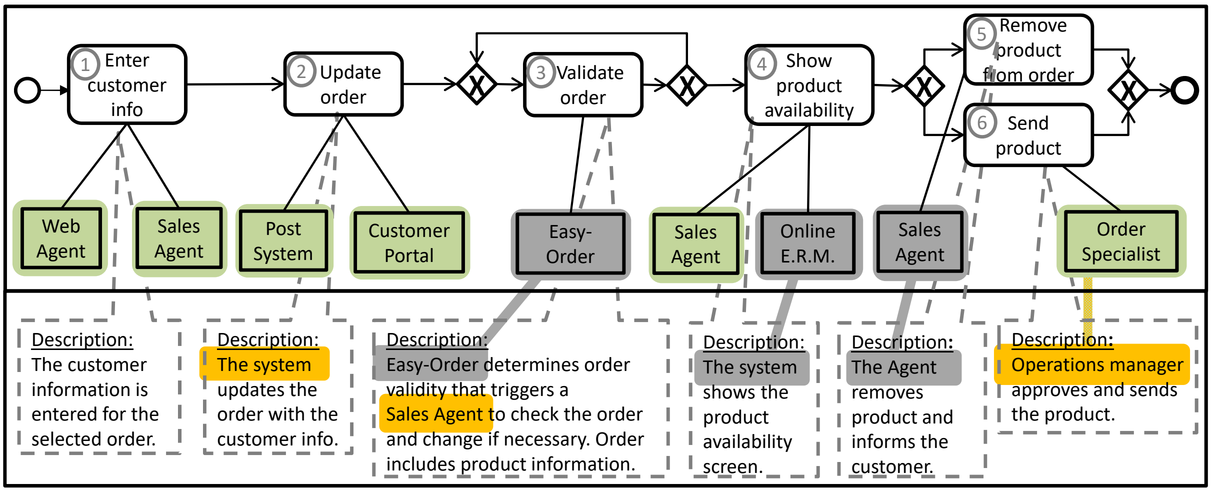

Figure 1: The process model for “Order Update" process and its textual descriptions

role. The green shade highlights role elements that are not referred to in the text, while the orange shade points to roles that are mentioned in the text but not placed as part of the model.

A closer look at the figure reveals that for some activities, the text properly describes the activity and the related role. For example, for Activity 5, the text clearly indicates that “*Sales Agent*" performs some actions in conformance with the role element connected to this activity although the role is referred to as "*The Agent*" in the description. However, in some other cases, there is a clear inconsistency between the role elements connected to an activity and related description. By evaluating the correspondence of the role elements in the model and roles mentioned in the text, we can distinguish three types of such inconsistencies.

Type 1 inconsistency is about ambiguity in roles. Activity 1 and 2 exemplify this situation. Two role elements, namely “*Web Agent*" and “*Sales Agent*", are connected to Activity 1. In its description, no clear responsibility is identified for this activity due to the passive sentence structure. Thus, the responsible role is not clear. For Activity 2, although the activity description mentions an actor, “*the system*", we cannot identify which one “*the system*" refers to, since there are two systems connected to this activity as role elements.

Type 2 inconsistency relates to missing linked role elements. For Activity 3, the first actor mentioned in the activity description, “*Easy-Order*", clearly refers to the connected “*Easy-Order*" element in the process model. However, the second actor mentioned in the activity description, “*Sales Agent*", is absent in the process model, which may indicate the lack of a role element or a problem in the activity scope.

Type 3 inconsistency is about redundant linked role elements in the model. In the description of Activity 4, although one of the connected role elements, “*Online E.R.M.*", is referred to as “*the system*" in the description, the other role element, “*Sales Agent*", is not mentioned. This problem may emerge due to an incomplete activity description or a faulty placement of the role element.

Finally, we can observe the occurrence of inconsistencies of different types for a single activity. For instance, the actor specified in the description of Activity 6, “*Operations Manager*", does not appear in the process model (*type 2*), and at the same time, the role element connected to Activity 6, “*Order Specialist*", is not referred to in its description (*type 3*). It is possible to observe other combinations of inconsistencies as well, i.e. *type 1* and *type 2*, *type 1* and *type 3*, *type 2* and *type 3*, and type 1, 2, and 3 all together.

This brief illustration of issues clearly shows that there are several types of inconsistencies which need to be handled. In the next section, we present our approach to detect role inconsistencies by using process models and activity descriptions as inputs based on NLP techniques and organizational semantics.

3 Approach

To accomplish the task of identifying role inconsistencies in processes, we designed an automated solution, which uses NLP techniques and organizational semantics to identify and compare elements related to the organizational perspective of processes. The approach detects three types of inconsistencies defined in Section 2.4. Section 3.1 presents an overview of the approach. In the subsequent sections (Section 3.2 to 3.5), we describe the detailed steps of the approach.

3.1 Overview

The steps of our approach, as applied for each activity in a process model, are depicted in Figure 2. In the first step, we perform a linguistic analysis of the activity description to identify the set of roles mentioned in the text. This step is composed of three sub-steps: (1) identify subjects in the sentences, (2) specify the set of relevant subjects among them using the organizational semantics captured by the *organizational entity catalog*, and (3) resolve subjects that are used with a determiner (e.g. an article). In the second step, we extract the set of linked role elements connected to the activity in the process model. In the third step, we assess the correspondence of the two sets obtained from the previous steps. The correspondence of two sets means that each element of one set can be mapped to an element of the other set (Stoll, 1979). In our case, the correspondence of the two sets indicates that there is no role inconsistency for the activity. If the two sets are not correspondent, we move to the next step, inconsistency discovery, to identify all role inconsistencies for the activity in question. The details of each step are explained in the following sections.

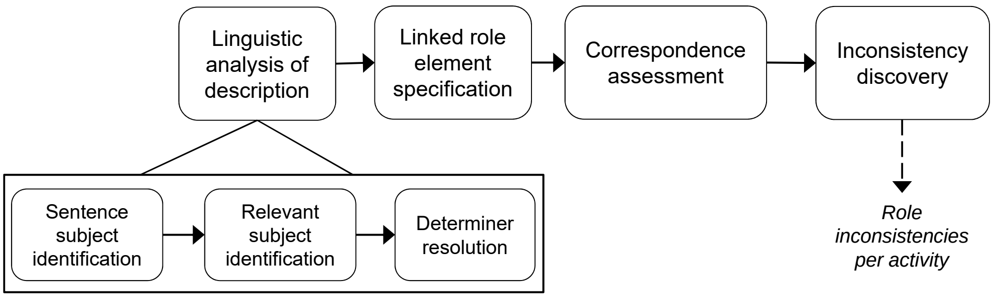

Figure 2: Overview of the Approach

3.2 Linguistic analysis of description

We apply linguistic analysis on an activity description to identify the grammatical structure of the text and determine the relevant sentence subjects as roles. This step consists of three sub-steps as described below.

Sentence subject identification: To identify the subjects in the sentences of the activity description, we make use of part-of-speech tags provided by the Stanford Parser, a widely used NLP tool (Nivre et al., 2016). More specifically, we take the entire set of noun phrases associated with the verb phrases in each sentence. For example, for the description of Activity 3 in our running example (Figure 1), the following subject set is identified:

Subject set of Activity 3 description: {*Easy-Order*, *Sales Agent*, *Order*}

Relevant subject identification: Since an activity description may provide various details about how the activity is performed, it is possible that it contains subjects that are not roles, i.e. actors or systems that actively perform some actions. To remove such irrelevant subjects, we apply named entity recognition using the *organizational entity catalog*. This catalog is a list of all linked role elements in the process repository and their types, i.e. if an element is an actor or a system. For a given process repository, the entity list can be extracted automatically. It may be necessary to assign the type manually depending on the notation used (e.g. it can be extracted automatically for an EPC model because actors and systems are represented with different symbols). An example excerpt from the organizational entity catalog for our running example (Figure 1) can be seen in Table 1. In this step, for each item in the subject set, we compare the item with the entries of the organizational entity catalog. If there is a match, we deduce that it is a relevant subject as a role in the process.

Relevant subject set of Activity 3 description: {*Easy-Order*, *Sales Officer*}.

Determiner resolution: Due to the flexibility of natural language, we recognize that systems and roles can be referred to in the description in a short way by using a determiner. To recognize that a subject with a determiner refers to a relevant role element, we follow different approaches for the systems and actors, as described below.

Actors used with a determiner: This step requires a pre-processing of the entities in the organizational entity catalog that have a compound noun structure (e.g. “*Sales Agent*"). For such entities, the noun of the phrase (e.g. “*Agent*") is identified as the *supertype* of the associated entity, which refers to the full name of the entity. In this way, we extract the supertypes of all entities with a compound noun structure in the organizational entity catalog as exemplified in Table 2.
For each subject used with a determiner in the activity description, we check if that subject is among the supertypes in Table 2. An example is “*The Agent*" in the description of Activity 5. As the word “*Agent*" is among the subtypes, we call it a *supertype subject* and add it to the set of relevant subjects for the activity with a mark “*-det*" at the end of the word.

Relevant subject set of Activity 5 description: {*Agent-det*}.

Systems used with a determiner: The resolution of determiners for the word “system" is straightforward, as it is globally used to refer to all systems, independent from the name of the system. Thus, we add “*System-det*" to the list of relevant subjects as exemplified below for Activity 2:

Relevant subject set of Activity 2 description: {*System-det)*}.

It is also possible to have anaphoric references in the textual descriptions (e.g., he/she). In such a case, it means that the role referred to has already been used before in the text. Since our approach discovers the subject in its first occurrence, it is not necessary to identify it again when an anaphoric reference is used. Therefore, anaphoric references need not be specifically resolved.

ID	Entity	Type
E1	Online E.R.M.	System
E2	Customer Portal	System
E3	Post System	System
E3	Easy-Order	System
E3	Order Specialist	Actor
E4	Operations Manager	Actor
E5	Sales Agent	Actor
E6	Web Agent	Actor

Table 1: An excerpt from an organizational entity catalog

Entity	Supertype
Web Agent Back Office Agent Sales Agent	Agent
Operations Manager Sales Manager	Manager
Post Officer	Officer

Table 2: Entities from the organizational entity catalog and supertypes

3.3 Linked role element specification

In this step, we identify the set of linked role elements for each activity. We find each role element from the organizational entity catalog shown in Table 1 to identify if it is an actor or a system. As an example, the following linked role element sets are identified for Activity 1 and Activity 2.

Linked role element set of Activity 1: {*web agent (actor)*, *sales officer (actor)*}.

Linked role element set of Activity 2: {*post system (system)*, *customer portal (system)*}.

3.4 Correspondence assessment

In this step, we compare the set of *relevant subjects* and the set of *linked role elements* for an activity to assess if the two sets correspond, which means that the relevant subjects in the descriptions and the linked role elements can be mapped to each other. If they do, then there is no role inconsistency for this activity. To assess the correspondence of sets automatically, we approach the problem with the application of *set theory* (Stoll, 1979). In the literature, set theory has been used to represent knowledge and analyze the use of natural language elements (Genesereth et al., 1992). In our approach, we use set theory to formulate the automatic correspondence check.

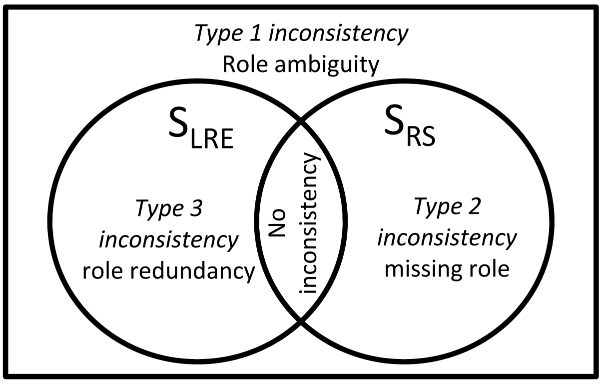

Figure 3: Venn diagram representing the sets and inconsistencies

We automatically apply set comparison operations to see if one of the four cases described below holds. The inconsistency problem at a high level can be depicted with the Venn diagram in Figure 3. Here, we refer to the set of linked role elements as S_{LRE} and the set of relevant subjects as S_{RS}. The existence of relevant subjects that are not modeled as a linked role element, shown at the right-hand-side of the Venn diagram, signals a *type 2* inconsistency: missing role. Linked role elements that are not mentioned as relevant subjects represent *type 3* inconsistencies, i.e. role redundancy. The area outside the sets of $\mathrm{S_{LRE}}$ and $\mathrm{S_{RS}}$ characterizes a *type 1* inconsistency, role ambiguity. Lastly, the intersection of the sets $\mathrm{S_{LRE}}$ and $\mathrm{S_{RS}}$ points to the activities with no inconsistency problems.
Our purpose in this step is to identify if an activity falls into the intersection area. For this, we perform set comparison operations that determine if the elements of the sets can be mapped to each other, as explained below for four possible cases. In equations 1 to 4, we provide formal representations of the set comparison operations for each case. In these equations, the function *det()* is used to check if a subject is a supertype subject, which returns *"true"* or *"false"*. We obtain the type of a subject with the function *type()*, which returns either "*system*" or the supertype of a role (as in Table 2).

Case 1: *No relevant subject and one linked role element*

$$S_{\mathrm{RS}} = \emptyset, \ \ |S_{\mathrm{LRE}}| = 1 \tag{1}$$

For this case, the existence of only one role element linked to the activity signifies that all mentioned descriptions address this single role. Thus, the linked role element maps to the relevant subjects, which do not exist in this case.

Case 2: *“M" linked role elements and “M" relevant subjects equal to those elements*

$$B = \{x : x \in S_{\text{RS}} | det(x) = \text{“}true\text{”}\}, \;\; S_{\text{RS}} = S_{\text{LRE}}, \;\; B = \emptyset \tag{2}$$

In this case, there are no relevant subjects with a determiner. We check if the relevant subjects and the linked role elements are the same. If the two sets are identical, this means that the description explains the actions of only and all of the linked role elements. Thus, the sets correspond.

Case 3: *One linked role element and one relevant supertype subject, which refers to this role element*

$$\begin{gathered} B = \{x : x \in S_{\text{RS}} | det(x) = \text{“}false\text{”}\}, \;\; B = \emptyset, \;\; |S_{\text{LRE}}| = 1, \;\; |S_{\text{RS}}| = 1 \\ \forall x \in S_{\text{LRE}}, \;\; \forall y \in S_{\text{RS}}, \;\; type(x) = type(y) \end{gathered} \tag{3}$$

In this case, although the relevant subject is referred to indirectly, since there is only one linked role, we may find a mapping. For this, we check if the types correspond to each other. If the determiner is used for the word “*system*" and the linked role element is of type “*system*", the relevant subject and the linked role map to each other. Similarly, for all other words with a determiner, if the linked role element is an “*actor*" of the same type with the supertype subject, elements also map to each other. Hence, the sets correspond.

Case 4: *Multiple relevant supertype subjects having different types and multiple linked role elements with corresponding types*

$$\begin{gathered} B = \{x : x \in S_{\text{RS}} | det(x) = \text{“}true\text{”}\}, \;\; C = S_{\text{LRE}} \backslash (S_{\text{RS}} \backslash B) \\ \forall x \in B, \;\; \exists! y \in C, \;\; type(x) = type(y), \;\; \forall z \in C, \;\; \exists! q \in B, \;\; type(z) = type(q) \\ (S_{\text{RS}} \backslash B) = (S_{\text{LRE}} \backslash C) \end{gathered} \tag{4}$$

In this more complicated case, which also covers the previous case, we address the situation where there exist both relevant subjects without a determiner and relevant supertype subjects in S_{RS}. To explain this case, we refer to the example sets of S_{RS} and S_{LRE} below. For the two sets to correspond, the relevant supertype subjects shall have different types (e.g. “*system*" and “*manager*") and they all need to map to one, and only one, linked role element (e.g. there exist only one *system* and one *manager* in the set of linked role elements). Additionally, the sets of all relevant subjects that are not supertype subjects and the rest of the linked role elements that are not the ones mentioned by the supertype subjects need to be equal. In the sets below, these are “*Sales Agent*" and “*Web Agent*".

S_{RS}={system-det, manager-det, Sales Agent, Web Agent}

S_{LRE}={Customer Portal, Sales Manager, Sales Agent, Web Agent}

3.5 Inconsistency discovery

The result of the previous step determines if there is an inconsistency for an activity or not. If the sets S_{RS} and S_{LRE} are deduced to correspond by falling into one of the cases, there is no inconsistency for that activity. The lack of correspondence signals the existence of one or more inconsistencies. We identify the set of inconsistencies by comparing the set of relevant subjects and the set of linked role elements. Below, we explain how we identify one or more inconsistency types based on the result of the set comparison.

Type 1 Inconsistency - Role Ambiguity:

This type of ambiguity is observed in two different ways, as described below.
(1) No relevant subject and multiple linked role elements

$$S_{\text{RS}} = \emptyset, \;\; |S_{\text{LRE}}| > 1 \tag{5}$$

The lack of a relevant subject signals that the description does not specify the actions of the linked role elements, as exemplified in Activity 1 of Figure 1. When this *type 1* inconsistency is found, there is no possibility of other inconsistencies to exist for this activity. Hence, the activity is not processed further.

(2) One relevant supertype subject and at least two linked role elements that have the same type as the supertype subject

$$\begin{aligned} &|S_{\mathrm{RS}}| = 1,\ \ |S_{\mathrm{LRE}}| > 1,\ \ \forall x \in S_{\mathrm{RS}},\ \ det(x) = \text{“}true\text{”} \\ &B = \{y : y \in S_{\mathrm{LRE}} | type(y) = type(x)\},\ \ |B| > 1 \end{aligned} \tag{6}$$

In this case, although there is a supertype subject, we cannot determine which role element it is referring to. This inconsistency is exemplified in Activity 2 of Figure 1. Multiple *type 1* inconsistencies may occur for an activity if multiple linked role elements exist for each supertype subject; for instance when two relevant subjects are found (“*system-det*", “*manager-det*") with four linked role elements (“*Post System*", “*Customer Portal*", “*Sales Manager*", “*Operations Manager*").

Type 2 Inconsistency - Missing Role:

$$B = \{x : x \in S_{\mathrm{RS}} | x \notin S_{\mathrm{LRE}}\},\ \ |B| > 0 \tag{7}$$

If there is any relevant subject which is not placed as a linked role element, for each relevant subject a *type 2 inconsistency* is generated, as exemplified in Activity 3 of the running example, Figure 1. Although this case seems straightforward, the existence of other relevant subjects and linked role elements may cause complications. In such a case, first, we identify the relevant subjects and role elements that are mapped to one another. At this step, we also resolve the subjects with determinants by mapping them to the roles they refer to. Then, we use the unmapped linked role elements to identify *type 2* inconsistencies. Multiple *type 2* inconsistencies may be observed for an activity if there are many relevant subjects that do not match the linked role elements. This is possible, for example, when we add one more sentence to the description of Activity 3 in Figure 1: “*Order specialist may need to approve some of the changes.*". This type can be observed together with *type 1* inconsistency when other relevant subjects of different type are mentioned, but not linked as a role element.

Type 3 Inconsistency - Role Redundancy:

$$B = \{x : x \in S_{\mathrm{LRE}} | x \notin S_{\mathrm{RS}}\},\ \ |B| > 0 \tag{8}$$

If there is any linked role element which is not among the relevant subjects, for each role a *type 3 inconsistency* is generated. Similar to *type 2*, the processing of this case requires careful matching of other subjects and linked role elements including the roles mentioned by the supertype subjects. Multiple *type 3* inconsistencies may take place for an activity if there are linked role elements that are not mentioned as relevant subjects. This type can co-occur with *type 1* when the linked role element mentioned by a supertype subject cannot be resolved. It can also be observed together with *type 2*, as exemplified in Activity 6 of the running example, Figure 1.

4 Evaluation

In this section, we present the result of our quantitative evaluation on how accurately the proposed approach is able to identify inconsistencies about roles in process activities. We developed a software tool to implement the described approach in an automated way [1]. Domain experts have manually identified the inconsistencies in a selected collection of models. We refer to this manual identification as the *gold standard* against which we compare the results generated by our approach. In the following sections, we present the test collection, the evaluation method we used, and its results.

[1] The code can be found at: http://www.expertjudgement.com/RoleInconsistencyAnalyzerSourceCode.zip

4.1 Test Collection

To evaluate our approach, we used an existing collection of process models and related activity descriptions. The collection originates from a major telecommunications company, which has more than 700 process model diagrams and 6000 activities in its repository. The repository is divided into sections based on the major process areas of the company in its value chain. These sections are developed and maintained by different units. Hence, they deviate in their style of modeling and textual description. To be able to evaluate our approach for inputs with varying characteristics and improve the external validity of the results, we randomly selected processes from all five major process areas (PA1 to PA5). The complete test collection consists of 36 models and 282 activities. Table 3 depicts the metrics representing the characteristics and the number of inconsistencies identified in the gold standard for each process area.

Process Area:	**PA1**	**PA2**	**PA3**	**PA4**	**PA5**	**Total**
Models	8	8	4	8	8	36
Activities	56	88	45	38	55	282
Activity per model	7	11	11.25	4.75	6.88	7.83
Role elements (role)	60	175	71	61	81	448
Role per model	7.50	21.88	17.75	7.63	10.13	12.44
Role per activity	1.07	1.99	1.58	1.61	1.47	1.59
Inconsistencies	48	123	49	52	58	330
Inconsistency per activity	0.86	1.40	1.09	1.37	1.05	1.17
Activities with inconsistencies	31	65	28	28	35	187
Type 1 Inconsistency per activity	0	0.14	0.04	0.08	0.07	0.08
Type 2 Inconsistency per activity	0.43	0.38	0.22	0.47	0.40	0.39
Type 3 Inconsistency per activity	0.43	0.89	0.82	0.82	0.58	0.71

Table 3: Overview of the test collection

The gold standard was prepared by three domain experts who were working as business process administrators and consultants in the company. We introduced the three role inconsistency types to the domain experts. We asked them to classify the problems they identify in the test collection according to the inconsistency types and report those problems that do not fit into any type. One of the domain experts identified inconsistencies for the activities in the test collection. The others checked the list of inconsistencies independently and identified the disagreements. The disagreements were discussed and resolved among the domain experts. All problems identified could be classified into one of the inconsistency types. Out of the 282 activities evaluated in the process models, eight included spelling errors and seven included subjects not defined in the organizational entity catalog. Those activities were neither changed nor removed from the evaluation set.

4.2 Evaluation Method

We quantitatively evaluated the accuracy of the inconsistency identification of the approach by calculating the precision and recall scores. Precision shows the ratio of correctly identified inconsistency pairs, while recall describes the ratio of actual inconsistency pairs identified by our approach. We use the equation 9 to calculate precision and recall. S_A denotes the set of actual inconsistencies in the gold standard, and S_I describes the inconsistencies identified by our technique.

$$precision = \frac{|S_A \bigcap S_I|}{|S_I|} \quad recall = \frac{|S_A \bigcap S_I|}{|S_A|} \tag{9}$$

4.3 Results

We computed the precision and recall scores for the complete process set, for the five process areas, and for each inconsistency type. The left part of Figure 4 shows the precision-recall graph of our approach for inconsistency types. The right part of Figure 4 shows the precision-recall graph of the approach for five process areas. The overall precision score for the complete test collection is 0.85. This value varies from 0.74 and 0.91 among inconsistency types and 0.81 to 0.88 among process areas. The recall score for the entire set is 0.91. The score for the inconsistency types range between 0.86 and 1.0 and for the process areas between 0.85 and 1.0. The recall scores indicate that, for some cases, the approach is able to find all inconsistencies identified in the gold standard, whereas the precision measurements highlight the capability of the approach to correctly identify the inconsistencies.

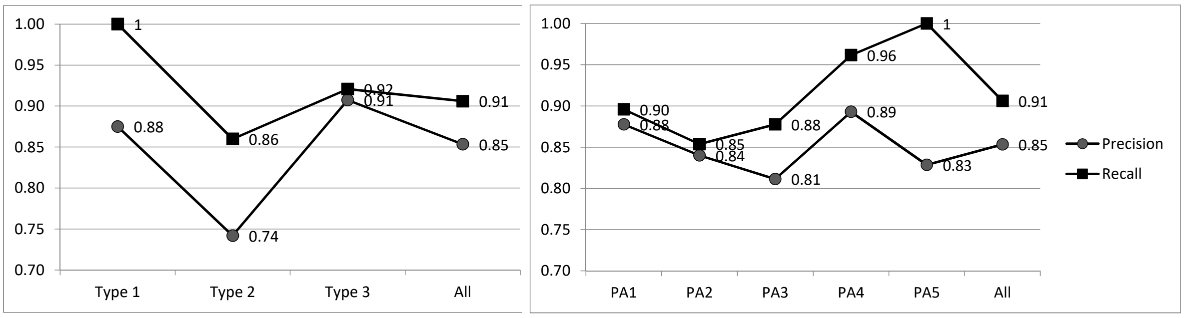

Figure 4: Precision-recall graph for inconsistency types (left) and process areas (right)

5 Discussion

Our results show that the approach successfully identifies inconsistencies between activity descriptions and linked role elements. An analysis of the variation in the precision and recall scores for the three inconsistency types and two process areas reveals the strengths and the limitations of the approach.

First, the low variation among the precision and the recall scores for process areas indicates that the approach may produce consistently accurate results for different process model collections. The recall scores of PA4 and PA5 were relatively higher than the overall recall score. The detailed analysis of these test collections revealed for many activities that the descriptions are either short or listed as bulleted sentences. Since these descriptions were written concisely, our automated approach was able to uncover almost all inconsistencies that were also detected by the human readers. A limitation of our evaluation relates to *type 1* inconsistencies on ambiguities. Due to the low frequency of *type 1* inconsistencies, it was not possible to extensively evaluate the approach for this inconsistency type in this test collection.

Second, among the three inconsistency types, our approach is less successful in finding *type 2* inconsistencies, which are about roles mentioned in the description but not depicted in the process model. This indicates a limitation not uncommon for natural language analysis, since many ambiguous cases that are easy for human readers to resolve may be difficult for automatic NLP operations (Aa, Leopold, and Reijers, 2015). Because the detection of this inconsistency type relies heavily on the exact identification of the sentence subjects, the ambiguities affect it the most. Although the determiner resolution step improves this aspect, not all ambiguities can be resolved. Some of such cases relate to the use of undefined abbreviations or different naming styles of roles such as “*operations manager*" vs. “*manager of the operations*". Other cases relate to wrongly-defined organizational entity catalog items. Though the inconsistency identified by the approach may not be relevant in such a case, it may still be useful to fix the problem in the entity catalog. In other cases, the approach may incorrectly detect inconsistencies because of the wrong organizational entity definitions. Still, detecting these inconsistencies may highlight the root-cause of the problem in the organizational entity catalog and help practitioners to fix it.

In this study, we applied the approach on one process model repository. Yet, the approach is generalizable to be used in different cases. Due to the similarities in how process modeling notations represent the organizational perspective, as argued in Section 2.1, the approach can be used for other notations as well. The only change required is the extraction of process model elements from the process model during the step of linked role element specification (Section 3.3). Another possible difference is that other process model repositories may contain textual descriptions not specifically for each activity but rather concern the whole process. In that case, a preprocessing step needs to be added to the approach to identify parts of the text that map to individual activities. This can be implemented through existing NLP approaches (Aa, Leopold, and Reijers, 2016a). Applying the approach in other repositories would also confirm the completeness of the inconsistency types or reveal new types.

Our study provides several implications for research and practice. The high precision and recall scores highlight the value of the approach in achieving high quality models, which is essential for practitioners (Schuette and Rotthowe, 1998). Since there may exist a considerable number of role-related problems in process models (see the number of inconsistencies in Table 3), it may still require a substantial amount of effort to fix these problems. To enhance the applicability of the approach in practice, it would be useful if recommendations could be automatically generated. The proposed approach provides the means to researchers to generate such recommendations by specifying the type of the inconsistencies and their causes. Additionally, by implementing the approach in a modeling tool, practitioners can use the approach during model creation to prevent errors right from the start, or after developing the models to reveal quality problems and fix them. Lastly, the approach may support practitioners to improve the functional perspective of their process models as well. As they examine the role inconsistencies, it is likely that they discover ill-scoped activities. They may observe that either too many role elements are linked or too many actions are described. The consideration of synonyms for the entities is excluded from our study. It may indeed be more helpful to practitioners not to merge the synonyms, since this can highlight the different terms used for the same concept (e.g., agent and clerk) and inconsistencies in terminology use.

6 Conclusion

In this paper, we presented an approach to automatically detect inconsistencies in the organizational perspective of process models using process elements and related activity descriptions. The approach combines natural language analysis, organizational semantics, and set comparison to identify correspondences among process elements and activity descriptions. The approach identifies three types of inconsistencies: role ambiguity, missing roles, and role redundancy. We quantitatively evaluated our approach on a process set selected from five process areas with diverse characteristics, which resides in a collection of real-life process model repository. The evaluation against a manually constructed gold standard shows that this approach can accurately identify the majority of role inconsistencies in process models. Thus, organizations can use our approach to quickly gain insights into the activities for which problems about role definitions are highly probable. By focusing their attention to those most problematic activities, organizations can significantly improve the quality of the organizational perspective of their process models and at the same time, achieve improvements in other aspects as they resolve inconsistencies.

In future work, we plan to develop additional features aimed at improving the organizational perspective of process models. The approach can be extended to identify activities and related roles that do not follow generally-accepted or organization-specific modeling practices. For instance, activities that describe the actions of more than one role possibly indicate that the scope of the activity is too large such that it needs to be decomposed into multiple activities. Moreover, with an extended natural language analysis, the approach may be used to specify not only the roles that actively perform actions but also those which have a passive role (e.g. receiving information) in the activity. Then, it is possible to distinguish between different role types and develop recommendations to improve processes. Lastly, another feature to be worked on is the generation of recommendations and automated update of process models based on the selected recommendations.

References

Aa, H. der, J. Carmona, H. Leopold, J. Mendling, and L. Padró (2018). "Challenges and Opportunities of Applying Natural Language Processing in Business Process Management." *Proceedings of the 27th International Conference on Computational Linguistics*, 2791–2801.

Aa, H. van der, H. Leopold, F. Mannhardt, and H. A. Reijers (2015). "On the Fragmentation of Process Information: Challenges, Solutions, and Outlook." In: *Enterp. Business-Process Inf. Syst. Model. 16th Int. Conf. BPMDS 2015, 20th Int. Conf. EMMSAD 2015, Held CAiSE 2015, Stock. Sweden, June 8-9, 2015, Proc.* Ed. by K. Gaaloul, R. Schmidt, S. Nurcan, S. Guerreiro, and Q. Ma. Cham: Springer International Publishing, pp. 3–18. ISBN: 978-3-319-19237-6.

Aa, H. van der, H. Leopold, and H. A. Reijers (2015). "Detecting Inconsistencies Between Process Models and Textual Descriptions." In: *Bus. Process Manag. 13th Int. Conf. BPM 2015, Innsbruck, Austria, August 31 – Sept. 3, 2015, Proc.* Ed. by H. R. Motahari-Nezhad, J. Recker, and M. Weidlich. Cham: Springer International Publishing, pp. 90–105. ISBN: 978-3-319-23063-4.

— (2016a). "Comparing textual descriptions to process models – The automatic detection of inconsistencies." *Inf. Syst.* 1–14. ISSN: 03064379.

— (2016b). "Dealing with Behavioral Ambiguity in Textual Process Descriptions." In: *Bus. Process Manag. 14th Int. Conf. BPM 2016, Rio Janeiro, Brazil, Sept. 18-22, 2016. Proc.* Ed. by M. La Rosa, P. Loos, and O. Pastor. Cham: Springer International Publishing, pp. 271–288. ISBN: 978-3-319-45348-4.

Aysolmaz, B., H. Leopold, H. A. Reijers, and O. Demirörs (2018). "A semi-automated approach for generating natural language requirements documents based on business process models." *Information and Software Technology* 93, 14–29. ISSN: 09505849.

Aysolmaz, B. and H. A. Reijers (2017). "Uses Cases for Process Model Comprehension." In: *Proc. 29th Int. Conf. Adv. Inf. Syst. Eng. In Press*. Lecture Notes in Computer Science. Springer-Verlag, Berlin.

Baier, T. and J. Mendling (2013). "Bridging Abstraction Layers in Process Mining by Automated Matching of Events and Activities BT - Business Process Management." In: ed. by F. Daniel, J. Wang, and B. Weber. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 17–32. ISBN: 978-3-642-40176-3.

Birkmeier, D. Q., S. Klockner, and S. Overhage (2010). "An Empirical Comparison of the Usability of BPMN and UML Activity Diagrams for Business Users." *18th European Conference on Information Systems*, 51. ISSN: 03029743.

Browning, T. R. (2010). "On the alignment of the purposes and views of process models in project management." *J. Oper. Manag.* 28 (4), 316–332.

Curtis, B., M. I. Kellner, and J. Over (1992). "Process Modeling." *Commun. ACM* 35 (9), 75–90. ISSN: 0001-0782.

Davies, I., P. Green, M. Rosemann, M. Indulska, and S. Gallo (Sept. 2006). "How do practitioners use conceptual modeling in practice?" *Data Knowl. Eng.* 58 (3), 358–380. ISSN: 0169023X.

Dumas, M., M. La Rosa, J. Mendling, and H. A. Reijers (2018). *Fundamentals of business process management*. Springer. ISBN: 9783662565087.

Ferrari, A., H. F. Witschel, G. O. Spagnolo, and S. Gnesi (2018). "Improving the quality of business process descriptions of public administrations: Resources and research challenges." *Business Process Management Journal* 24 (1), 49–66. ISSN: 14637154.

Friedrich, F., J. Mendling, and F. Puhlmann (2011). "Process Model Generation from Natural Language Text." In: *Adv. Inf. Syst. Eng. 23rd Int. Conf. CAiSE 2011, London, UK, June 20-24, 2011. Proc.* Ed. by H. Mouratidis and C. Rolland. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 482–496. ISBN: 978-3-642-21640-4.

Genesereth, M., R. E. Fikes, R. Brachman, T. Gruber, P. Hayes, R. Letsinger, V. Lifschitz, R. Macgregor, J. Mccarthy, P. Norvig, and R. Patil (1992). *Knowledge Interchange Format Version 3.0 Reference Manual*.

Mendling, J., H. M. W. Verbeek, B. F. V. Dongen, W. M. van der Aalst, and G. Neumann (2008). "Detection and Prediction of Errors in EPCs of the SAP Reference Model." *Data Knowl. Eng.* 64 (1), 312–329.

Nawrocki, J., T. Nedza, M. Ochodek, and L. Olek (2006). "Describing Business Processes with Use Cases." In: *Business Information Systems - BIS 2006*, pp. 13–27.

Nivre, J., M.-C. de Marneffe, F. Ginter, Y. Goldberg, J. Hajic, C. D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira, et al. (2016). "Universal dependencies v1: A multilingual treebank collection." In: *Proceedings of the 10th International Conference on Language Resources and Evaluation*.

Object Management Group (2011). *Business Process Model and Notation (BPMN) formal/2011-01-03*. Tech. rep. Object Management Group. URL: http://www.omg.org/spec/BPMN/2.0.

— (2017). *Unified Modeling Language (UML), Version 2.5.1*. Tech. rep. URL: http://www.omg.org/spec/UML/2.5.1/.

Ottensooser, A., A. Fekete, H. A. Reijers, J. Mendling, and C. Menictas (2012). "Making sense of business process descriptions: An experimental comparison of graphical and textual notations." *Journal of Systems and Software* 85 (3), 596–606. ISSN: 0164-1212.

Pittke, F., H. Leopold, and J. Mendling (2013). "Spotting terminology deficiencies in process model repositories." *Lect. Notes Bus. Inf. Process.* 147 LNBIP, 292–307. ISSN: 18651348.

— (2015a). "Automatic Detection and Resolution of Lexical Ambiguity in Process Models." *Software Engineering, IEEE Transactions on* PP (99), 1. DOI: 10.1109/TSE.2015.2396895.

— (2015b). "When Language Meets Language: Anti Patterns Resulting from Mixing Natural and Modeling Language." In: *Bus. Process Manag. Work. SE - 11*. Ed. by F. Fournier and J. Mendling. Vol. 202. Lecture Notes in Business Information Processing. Springer International Publishing, pp. 118–129. ISBN: 978-3-319-15894-5.

Pittke, F., P. H. P. Richetti, J. Mendling, and F. A. Baião (2015). "Context-Sensitive Textual Recommendations for Incomplete Process Model Elements." In: *Bus. Process Manag. 13th Int. Conf. BPM 2015, Innsbruck, Austria, August 31 – Sept. 3, 2015, Proc.* Ed. by H. R. Motahari-Nezhad, J. Recker, and M. Weidlich. Cham: Springer International Publishing, pp. 189–197. ISBN: 978-3-319-23063-4.

Recker, J., M. Indulska, M. Rosemann, and P. Green (2006). "How Good is BPMN Really? Insights from Theory and Practice." In: *14th European Conference on Information Systems*. Goeteborg, Sweden, pp. 1–12.

Recker, J., M. Rosemann, M. Indulska, and P. Green (2009). "Business process modeling: a comparative analysis." *J. Assoc. Inf. Syst.* 10 (4), 333–363.

Rosemann, M. (2006). "Potential pitfalls of process modeling: part A." *Bus. Process Manag. J.* 12 (2), 249–254.

Russell, N., W. M. P. van der Aalst, A. H. M. ter Hofstede, and D. Edmond (2005). "Workflow Resource Patterns: Identification, Representation and Tool Support." In: *Adv. Inf. Syst. Eng. 17th Int. Conf. CAiSE 2005, Porto, Port. June 13-17, 2005. Proc.* Ed. by O. Pastor and J. e Cunha. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 216–232. ISBN: 978-3-540-32127-9.

Scheer, A. W. (2000). *ARIS - business process modeling*. 3rd Editio. Springer Verlag.

Schefer-Wenzl, S. and M. Strembeck (2014). "Modeling Support for Role-Based Delegation in Process-Aware Information Systems." *Bus. {&} Inf. Syst. Eng.* 6 (4), 215–237. ISSN: 1867-0202.

Schuette, R. and T. Rotthowe (1998). "The Guidelines of Modeling – An Approach to Enhance the Quality in Information Models." In: *Concept. Model. – ER '98 17th Int. Conf. Concept. Model. Singapore, Novemb. 16-19, 1998. Proc.* Ed. by T.-W. Ling, S. Ram, and M. Li Lee. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 240–254. ISBN: 978-3-540-49524-6.

Soares Silva, T., L. Thom, A. Weber, J. Palazzo Moreira de Oliveira, and M. Fantinato (Oct. 2018). "Empirical Analysis of Sentence Templates and Ambiguity Issues for Business Process Descriptions." In: *OTM 2018: On the Move to Meaningful Internet Systems*, pp. 279–297. ISBN: 978-3-030-02609-7.

Stoll, R. R. (1979). *Set theory and logic*. Dover Publications, p. 512. ISBN: 9780486638294.

Weber, B., M. Reichert, J. Mendling, and H. A. Reijers (2011). "Refactoring Large Process Model Repositories." *Comput. Ind.* 62 (5), 467–486.

Weske, M. (2007). *Business Process Management Concepts, Languages, Architectures*. Springer Berlin Heidelberg New York This, p. 372. ISBN: 978-3-540-73521-2.