Dealing with Behavioral Ambiguity in Textual Process Descriptions

Han van der Aa[1], Henrik Leopold[1], Hajo A. Reijers[1,2]

[1] Department of Computer Sciences, VU University Amsterdam, The Netherlands
[2] Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands

Abstract. Textual process descriptions are widely used in organizations since they can be created and understood by virtually everyone. The inherent ambiguity of natural language, however, impedes the automated analysis of textual process descriptions. While human readers can use their context knowledge to correctly understand statements with multiple possible interpretations, automated analysis techniques currently have to make assumptions about the correct meaning. As a result, automated analysis techniques are prone to draw incorrect conclusions about the correct execution of a process. To overcome this issue, we introduce the concept of a *behavioral space* as a means to deal with behavioral ambiguity in textual process descriptions. A behavioral space captures all possible interpretations of a textual process description in a systematic manner. Thus, it avoids the problem of focusing on a single interpretation. We use a compliance checking scenario and a quantitative evaluation with a set of of 47 textual process descriptions to demonstrate the usefulness of a behavioral space for reasoning about a process described by a text. Our evaluation demonstrates that a behavioral space strikes a balance between ignoring ambiguous statements and imposing fixed interpretations on them.

1 Introduction

Automated techniques for the analysis of business processes provide a wide range of valuable opportunities for organizations. Among others, they allow to check for business process compliance [12], to identify redundant activities within an organization [11], and to identify operational overlap between two business processes [5]. What all these techniques have in common is that they rely on *process models* as input. That is, they build on the formally specified relationships between the activities of process models to perform their analyses. Thus, these techniques cannot be applied to less structured forms of process documentation such as textual process descriptions.

The relevance and widespread use of textual process descriptions as source for process analysis has been emphasized in various contexts [1,6,10,18]. However, the inherent ambiguity of textual process descriptions is a challenge to their utilization for analysis purposes. A simple natural language statement such as "*in parallel to the latter steps*" leaves considerable room for interpretation. Whether the word "*latter*" refers to the preceding two, three, or even more activities mentioned in the textual description is, in many cases, impossible to infer with certainty. While human readers can use their context knowledge to make sense of such statements, it is hardly possible for automated analysis approaches to resolve such cases. In prior work, techniques for automatically extracting process models from textual process descriptions circumvented this problem by introducing interpretation heuristics [6,8,19]. In this way, they obtained a single process-oriented interpretation of the text. This interpretation, however, contains assumptions on the correct interpretation of undecidable ambiguity issues. So, there is always the risk that the derived interpretation conflicts with the proper way to execute the process. As a result, the focus on a single interpretation can lead to incorrect outcomes when reasoning about a business process, e.g. incorrect assessments on its compliance to regulations or expectations.

To provide a rigorous solution for these reasoning problems, we introduce a novel concept we refer to as *behavioral space*. A behavioral space formally captures all possible behavioral interpretations of a textual process description. The behavioral space clearly defines which behavior is within and which behavior is outside the reasonable bounds of interpretation. Therefore, it allows us to reason about, for example, compliance without the need to impose assumptions on the correct interpretation of a text.

The remainder of the paper is structured as follows. Section 2 motivates the problem of reasoning under behavioral ambiguity in textual process descriptions. Section 3 introduces the notion of a behavioral space to capture behavioral ambiguity and Section 4 describes how these can be obtained from a text. Section 5 illustrates the usage of behavioral spaces for compliance checking. In Section 6 we demonstrate the importance of behavioral spaces through a quantitative evaluation. Section 7 discusses streams of related work. Finally, we conclude the paper and discuss directions for future research in Section 8.

2 Behavioral Ambiguity in Textual Process Descriptions

In this section, we illustrate the problem of reasoning about business processes based on textual process descriptions. The key challenge in this context is the ambiguity of textual process descriptions, in particular with respect to how the text describes the ordering relations between activities. In the remainder, we refer to such ambiguity as *behavioral ambiguity*. Figure 1 illustrates the problem of behavioral ambiguity by showing a simplified description of a claims handling process. The description uses typical patterns to describe ordering relations, as observed in process descriptions obtained from practice and research [6].

At first glance, the description from Figure 1 appears to be clear. However, on closer inspection, it turn outs that the description does not provide conclusive answers to several questions regarding the proper execution of the described process. For instance:

- Q1. Is it allowed that the claims officer records the claim information before reviewing the request?
- Q2. Which steps must be repeated upon receipt of additional information from the claimant?
- Q3. When can the financial department start taking care of the payment?

After a claim is received, a claim officer reviews the request and records the claim information. The claim officer then validates the claim documents before writing a settlement recommendation. A senior officer then checks this recommendation. The senior officer can request further information from the claimant, or reject or accept the claim. In the former case, the previous steps must be repeated once the requested information arrives. If a claim is rejected, the claim is archived and the process finishes. If a claim is accepted, the claim officer calculates the payable amount. Afterwards, the claims officer records the settlement information and archives the claim. In the meantime, the financial department takes care of the payment.

Fig. 1: Exemplary description of a claims handling process.

Based on the information provided in the textual description, these questions are not clearly decidable. This lack of decidability results from two forms of behavioral ambiguity: type ambiguity and scope ambiguity. *Type ambiguity* occurs when a textual description does not clearly specify the type of order relationship between two activities. For instance, the relation between the "*review request*" and "*record claim information*" activities in the first sentence is unclear. The term "*and*" simply does not allow us to determine whether these activities must be executed sequentially or can be executed in an arbitrary order. *Scope ambiguity* occurs when statements in a textual description underspecify to which activity or activities they precisely refer. This type of ambiguity particularly relates to repetitions and parallelism. For instance, the statement that "*the previous steps must be repeated*" does not clearly specify which activities must be performed again. Similarly, the expression"*in the meantime*" does not define when the financial department can start performing its activities.

As a result of such ambiguities, there are different views on how to properly carry out the described process. When deriving a single structured interpretation from a textual process description, as is done by process model generation techniques (cf. [6,8,19]), there is thus always the risk that a derived interpretation conflicts with the proper way to execute the process. The focus on a single interpretation can, therefore, lead to wrong conclusions when reasoning about a business process. This can, for instance, result in a loss of efficiency by not allowing for parallel execution where possible (Q3). Furthermore, it can even result in noncompliance to regulations, for example, by failing to impose necessary ordering restrictions (Q1) or by not repeating the required steps when dealing with the receipt of new claim information (Q2).

To avoid the problems associated with fixed interpretations, automated reasoning techniques should take into account all reasonable interpretations of a textual process description. For this reason, we use this paper to introduce the concept of a *behavioral space*. A behavioral space allows us to capture the full range of semantics possibly implied by textual descriptions in a structured manner. As such, it provides the basis to safely reason about described processes.

3 Capturing Behavioral Ambiguity using Behavioral Spaces

In this section, we introduce and define the concept of a *behavioral space*. The notion of a behavioral space provides the foundation to reason about properties such as conformance and similarity for behaviorally ambiguous process descriptions. The general idea of the notion is to represent the causes and effects of behavioral ambiguity in a structured manner. Behavioral ambiguity leads to different views on how to properly execute a business process. To capture these views, we first conceptualize a single view or *interpretation* of the process behavior described in a text. For the purposes of this paper, we express this behavior using the *behavioral profile* relations from [22].

Behavioral profile relations capture the ordering restrictions that are in effect between activities. Three different behavioral profile relations can exist for an activity pair (a_i, a_j). The *strict order* relation $a_i \rightsquigarrow a_j$ is used to express that activity a_i cannot be executed after the execution of activity a_j. The *exclusiveness* relation $a_i + a_j$ denotes that either activity a_i or activity a_j can be executed in a single process instance. Finally, the *interleaving order* relation $a_i \,||\, a_j$ states that a_i and a_j can be executed in an arbitrary order. Based on these behavioral profile relations, we define a *behavioral interpretation* of a textual process description as follows:

Definition 1 (Behavioral Interpretation). *Given a textual process description T and the set of behavioral profile relations $\mathcal{R} = \{\rightsquigarrow, +, ||\}$, we define a* behavioral interpretation *as a tuple $BI = (A_T, BP)$, with:*

- *A_T: the set of activities described in the textual process description T;*
- *$BP : A_T \times A_T \nrightarrow \mathcal{R}$: a partial function that assigns a behavioral profile relation from $\mathcal{R}$ to a pair of activities from A_T, if any.*

Table 1: Activities in the running example

ID	**Activity**	**ID**	**Activity**
a_1	Receive claim	a_8	Reject claim
a_2	Review request	a_9	Accept claim
a_3	Record claim information	a_{10}	Receive requested information
a_4	Validate documents	a_{11}	Calculate payable amount
a_5	Write settlement recommendation	a_{12}	Record settlement information
a_6	Check recommendation	a_{13}	Archive claim
a_7	Request further information	a_{14}	Arrange payment

Multiple behavioral interpretations for the same textual process description occur when the text contains statements about behavioral relations that can be interpreted in different ways. We refer to such statements as *behavioral statements*. Each behavioral statement consists of a single or several words and describes pair-wise relations between one or more activity pairs. An ambiguous relational statement can result in multiple, conflicting sets of pair-wise relations. For instance, the statement "*a claim officer reviews the request and records the claim information*", results in two different interpretations because it is unclear whether this statement implies a strict order or an interleaving order between the two described activities. Using the activity identifiers specified in Table 1, this results in two sets of behavioral relations, namely $\{a_2 \rightsquigarrow a_3\}$ and $\{a_2 \parallel a_3\}$. Given the set S_T of the behavioral statements in a text T, the set of possible behavioral interpretations $\mathcal{BI}_T$, follows naturally as the set of possible combinations of interpretations of statements in S_T. This results in a three-dimensional view on the behavioral relations that exist between activities, as visualized in Figure 2.

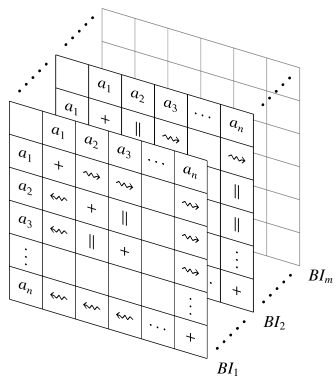

Fig. 2: A behavioral space as a collection of *m* behavioral interpretations

The behavioral space captures this spectrum of possible behavioral interpretations for a textual process description, as given by Definition 2.

Definition 2 (Behavioral Space). *Given a textual process description T and the behavioral profile relations $\mathcal{R} = \{\rightsquigarrow, +, \|\}$, we define a* behavioral space *as a tuple $\mathcal{S}_T = (A_T, S_T, \mathcal{BI}, \delta)$, with:*

- *A_T: the set of* activities *described in the textual process description T;*
- *S_T: the set of* behavioral statements *contained in the textual process description T;*
- *$\mathcal{BI}$: the set of* behavioral interpretations *of a textual process description T;*
- *$\delta : A_T \times A_T \rightarrow \mathcal{P}(S_T \times \mathcal{R})$, as a function that links the behavioral profile relations that can exist between activity pairs to sets of behavioral statements.*

In Definition 2, the function δ provides traceability between behavioral statements and the behavioral profile relations included in the behavioral interpretations for activities. This traceability can be used to provide diagnostic information when reasoning about compliance. We furthermore use $R(a_i, a_j) \subseteq \mathcal{R}$ as a short-hand to refer to the set of behavioral profile relations that can exist between activities a_i and a_j, e.g. $R(a_2, a_3) = \{\rightsquigarrow, \|\}$.

4 Obtaining Behavioral Spaces

The procedure to obtain a behavioral space from a textual process description consists of three main steps, as visualized in Figure 3. First, we identify the process activities described in the text T. This results in an activity set A_T, as shown in Table 1 for the claims handling example. Second, we identify the behavioral relations that exist among these activities. This step involves both the extraction of behavioral relations for unambiguous behavioral statements, as well as the extraction of sets of possible behavioral relations for ambiguous behavioral statements. Third, we combine the different interpretations of individual ambiguous statements into a collection of behavioral interpretations $\mathcal{BI}$ in order to obtain a behavioral space.

Approaches that generate process models from texts, cf. [6], address the challenges related to the identification of activities (step 1) and to the extraction of behavioral relations for *unambiguous* behavioral statements (part of step 2). Therefore, we focus here on the yet unaddressed challenges related to dealing with behavioral ambiguity, that is, obtaining sets of possible behavioral relations for *ambiguous* statements (Section 4.1) and combining these into behavioral interpretations of a described process (Section 4.2).

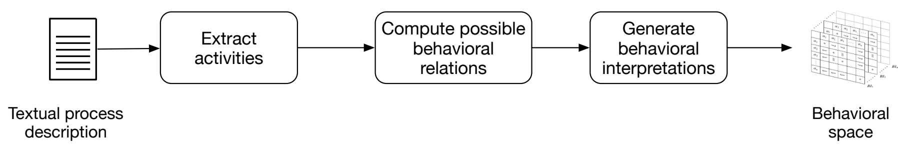

Fig. 3: Steps involved to obtain a behavioral space from a textual description

4.1 Computing Possible Behavioral Relations

Approaches that generate process models from textual descriptions use heuristics-based techniques to identify and analyze behavioral statements in a text. These techniques mainly build on predefined sets of indicators that pinpoint the different types of relations, e.g. "*then*" as well as "*afterwards*" for strict order relations and "*while*" as well as"*meanwhile*" for parallel or interleaving order relations. To identify ambiguous behavioral statements, we isolated a subset of these indicators that result or can result in behaviorally ambiguous statements. For example, the usage of "*meanwhile*" or "*in the meantime*" to indicate interleaving order relations results in statements with scope ambiguity. By contrast, this is not the case for "*while*" because this indicator is naturally accompanied by a scope specifier, e.g. "*while the claim is being archived*". Once a statement with behavioral ambiguity has been identified, we generate possible interpretations for these statements. Here, we treat statements with type and with scope ambiguity differently, since they result in different sets of behavioral relations.

Statements with Type Ambiguity A behavioral statement with type ambiguity describes that there exists a relation among a specific set of activities, but does not clearly state the type of relationship. For example, the first sentence of the running example does not clearly specify whether the order is important when executing the activities a_2 and a_3. To capture these different possibilities in the behavioral space, we generate an interpretation of this statement for each of the possible relation types, i.e. strict order and interleaving order. This results in two sets of relations that are linked to the ambiguous behavioral statement s_1: $\{a_2 \rightsquigarrow a_3\}$ and $\{a_2 \parallel a_3\}$.

Statements with Scope Ambiguity Dealing with behavioral statements with scope ambiguity is more complex. These statements describe the existence of a relation, but do not specify between which activities this relationship holds. For example, the statement "*the previous steps must be repeated, once the requested information arrives*", which we shall refer to as s_2, does not state which activities should be repeated. Though such statements are highly problematic, we do not have to be completely unaware about their meaning, i.e. about the possible sets of activities that the statements can refer to. In particular, we can utilize the notion that statements such as "*the previous steps*" and "*in the meantime*" relate to distinct parts of a process. This means that the set of activities to which these statements refer *cannot* be any arbitrary combination of activities. The activities in the set must rather have something in common, such as activities that are all executed by the same person.

For this reason, we generate interpretations for statements with scope ambiguity based on sets of activities that have a certain commonality. In particular, given a textual process description, we can identify sets of subsequently described activities that are (i) performed by the same resource, (ii) performed on or with the same (business) object, or (iii) are part of the same discourse statement (i.e. a choice in the process). Based on this, we can recognize that "*the previous steps*" in s_2, can refer to either:

1. The activities performed by the *senior claims officer*, i.e. $\{a_6, a_7, a_8, a_9\}$;
2. The activities related to the *settlement recommendation*, i.e. $\{a_5, a_6\}$;
3. All previous activities of the process, i.e. $\{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9\}$.

These three possibilities result in three sets of relations that can follow from the same behavioral statement. In the same way, we can obtain different interpretations for the statement s_3: "*In the meantime, the financial department takes care of the payment*". This statement can refer to the following sets of activities:

1. The activities performed by the *claims officer*, after a senior claims officer has accepted the claim, i.e. $\{a_{11}, a_{12}, a_{13}\}$;
2. The activities related to the *claim* object, i.e. $\{a_{13}\}$;
3. The last mentioned activity before the statement, i.e. $\{a_{13}\}$.

The last interpretation here differs from the third interpretation of statement s_2 because, unlike for s_2, statement s_3 can also refer to a single activity. In that case, "*in the meantime*" is interpreted to simply refer to the preceding activity. Recognizing that the two latter interpretations of statement s_3 encompass the same set of activities, this results in two instead of three possible interpretations of s_3.

4.2 Generating Behavioral Interpretations

Based on the relations extracted from unambiguous behavioral statements and the sets of possible relations for ambiguous behavioral statements, we can generate a set of behavioral interpretations $\mathcal{BI}$ for the entire textual description. As considered in the previous section, the claims handling process contains three ambiguous statements with, respectively, two, three, and two possible interpretations. We obtain behavioral interpretations by combining the interpretations of individual statements in all possible manners. For the claims handling process, this results in a behavioral space with 12 ($2 \times 3 \times 2$) possible interpretations in $\mathcal{BI}$. To complete the full behavioral profile relations for a behavioral interpretation, we make use of the transitivity of the strict order and interleaving order relations [20]. In this way, we can obtain relations beyond those pair-wise relations that we extracted from a textual description. For example, if a text specifies that activity a_i is followed by a_j and a_j is followed by a_k, i.e. $a_i \rightsquigarrow a_j$ and $a_j \rightsquigarrow a_k$, then a_i is also followed by a_k, i.e. $a_i \rightsquigarrow a_k$.

Once the behavioral interpretations have been constructed, the behavioral space is complete. Table 3 visualizes the possible behavioral relations for a fraction of the activities in the running example. The table illustrates that many of the relations are known with certainty. Still, due to the ambiguous behavioral statement s_3, the relations between, on the one hand, activities a_{11} and a_{12}, and, on the other, activity a_{14} can be both strict orders or interleaving orders. Finally, it is interesting to note that although the relation between a_{13} and a_{14} is affected by the ambiguous statement s_3, its relation type is known with certainty. This is because all possible interpretations of s_3 include the relation $a_{13} \parallel a_{14}$.

Table 2: Possible behavioral relations for activities of the claims handling process.

	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}
a_9	$+$	$\rightsquigarrow$	$\rightsquigarrow$	$\rightsquigarrow$	$\rightsquigarrow$	$\rightsquigarrow$
a_{10}		$\parallel$	$\rightsquigarrow$	$\rightsquigarrow$	$\rightsquigarrow$	$\rightsquigarrow$
a_{11}			$+$	$\rightsquigarrow$	$\rightsquigarrow$	$\parallel/\rightsquigarrow$
a_{12}				$+$	$\rightsquigarrow$	$\parallel/\rightsquigarrow$
a_{13}					$+$	$\parallel$
a_{14}						$+$

5 Reasoning Using Behavioral Spaces

By capturing behavioral ambiguity in a structured manner, behavioral spaces allow us to reason about behavioral properties without the need to arbitrarily settle ambiguity. Similar to behavioral profiles and process models, suitable reasoning tasks include similarity analysis, matching, and compliance checking. In this section, we show the usefulness of behavioral spaces for such reasoning tasks. To achieve this, we describe the specific use case of checking the compliance between a behavioral space and an execution trace.

The goal of compliance checking is to determine whether the behavior captured in an execution trace is allowed by the behavioral specification of a business process. The key difference between traditional compliance checking and compliance checking using behavioral spaces lies in the potential outcomes of a check. In traditional compliance checking, a trace is either *compliant* or it is *non-compliant* with a business process. By contrast, due to the behavioral ambiguity captured in behavioral spaces, a trace can be either compliant, non-compliant, but also *potentially compliant* with a behavioral space. The latter outcome occurs for traces that comply with one or more behavioral interpretations in a behavioral space, but not to all of them.

5.1 Behavioral Interpretation Compliance

Compliance checking of a trace t against a behavioral space $\mathcal{S}$ builds on the compliance checking of t against individual behavioral interpretations in $\mathcal{BI}_{\mathcal{S}}$. This is equal to the compliance check of a trace and a behavioral profile, as obtained from a process model (see [23]). This check builds on a comparison of the behavioral profile of a trace BP_t to the behavioral profile relations of behavioral interpretation BI. The behavioral profile BP_t captures the strict order and interleaving order relations for the set of activities A_t in a trace t. Given an activity pair $(a_i, a_j) \in (A_t \times A_t)$, BP_t contains the strict order relation $a_i \rightsquigarrow_t a_j$ iff at least one occurrence of activity a_i precedes an occurrence of activity a_j in t, and no occurrence of a_j precedes an occurrence of a_i in t. BP_t contains the interleaving order relation $a_i \parallel a_j$ iff at least one occurrence of a_i precedes an occurrence of a_j in t, and at least one occurrence of a_j precedes an occurrence of a_i in t.

Given a behavioral profile of a trace BP_t and a behavioral interpretation BI, we can determine if t is compliant to BI by checking if the relations in BP_t do not violate the behavioral relations in BI. Specifically, t is compliant to BI if all relations in BP_t are *subsumed* by the relations in BI. A relation type $R \in \mathcal{R}$ is subsumed by relation type $R' \in \mathcal{R}$ if the relation types are equal, i.e. $R = R'$, or if R' is less restrictive than R. The latter captures the notion that when an activity pair (a_i, a_j) is in a strict order or reverse strict order relation in $\mathcal{B}_t$, this does not violate an interleaving order relation in $\mathcal{B}_I$. In other words, $a_i \rightsquigarrow a_j \in BP_t$ is subsumed by the relation $a_i \parallel a_j \in BI$.

Based on the notion of subsumption, we define compliance between a trace and a behavioral interpretation in Definition 3. Here, for brevity we say that an activity pair (a_i, a_j) is in *reverse strict order*, denoted by $a_i \rightsquigarrow_t^{-1} a_j$, if and only if $a_j \rightsquigarrow_t a_i$.

Definition 3 (Trace to Behavioral Interpretation Compliance). *Let $t = e_1, \ldots, e_m$ be a trace with an activity set A_t and $BI \in \mathcal{BI}_{\mathcal{S}}$ a behavioral interpretation in the behavioral space $\mathcal{S}$, with $A_t \subseteq A_{\mathcal{S}}$.*

- *For an activity pair $(x, y) \in (A_t \times A_t)$, the relation $xRy \in \mathcal{B}_t \cup \{\rightsquigarrow_t^{-1}\}$ is subsumed by relation $xR'y \in \mathcal{B}_I \cup \{\rightsquigarrow_I^{-1}\}$, i.e.* the subsumption predicate *$sub(R, R')$ is satisfied,* iff $R = R'$ or $R' = \parallel$.
- *Trace t complies to behavioral interpretation BI if for each activity pair $(x, y) \in (A_t \times A_t)$ the relation in t is subsumed by the relation in BI, i.e. the* compliance predicate *$compl(t, BI)$ is satisfied,* iff $\forall R \in \mathcal{B}_t \cup \{\rightsquigarrow_t^{-1}\}, \mathcal{B}_I \cup \{\rightsquigarrow_I^{-1}\}$*, it holds $(xRy \wedge xR'y) \implies sub(R, R')$.*

5.2 Behavioral Space Compliance

Based on the compliance check between a trace and individual behavioral interpretations, we can determine the compliance of a trace to the full behavioral space. In particular, we can quantify the *support* of the behavioral space for a trace and extract the conditions under which this trace complies to the textual process description. We define the support of a behavioral space $\mathcal{S}$ for a trace t as the ratio between the number of interpretations to which t is compliant and the total number of interpretations in $\mathcal{BI}_{\mathcal{S}}$:

$$\text{supp}(t, \mathcal{S}) = \frac{|\{BI \in \mathcal{BI}_{\mathcal{S}} \mid compl(t, BI)\}|}{|\mathcal{BI}_{\mathcal{S}}|} \tag{1}$$

The support metric quantifies the fraction of interpretations that allow for a trace to occur. A support value of 1.0 indicates that a trace is without any doubt compliant to the behavioral space, i.e. independent of the chosen interpretation. A support of 0.0 shows that there is no interpretation under which a trace complies to the behavioral space. Therefore, it can be said with certainty that the trace is non-compliant to $\mathcal{S}$. Finally, any trace t with a support value $0.0 < \text{supp}(t, \mathcal{S}) < 1.0$ is potentially compliant to $\mathcal{S}$. This implies that there are certain interpretations of the textual description to which the trace complies. To illustrate the usefulness of the support metric and the additional compliance information that behavioral spaces can provide, consider the following three partial execution traces of the running example:

- Trace $t_1 =< a_1, a_2, a_3, a_4, a_5 >$;
- Trace $t_2 =< a_1, a_3, a_2, a_4, a_5 >$;
- Trace $t_3 =< a_{11}, a_{14}, a_{12}, a_{13} >$.

The traces t_1 and t_2 both describe an execution sequence for the first part of the claim handling process. The difference between the two is that in t_1 activity a_2 occurs before a_3, whereas these are executed in reverse order in t_2, i.e. $a_2 \rightsquigarrow a_3 \in BP_{t_1}$ and $a_2 \rightsquigarrow^{-1} a_3 \in BP_{t_2}$. Furthermore, recall that the behavioral relation between these two activities is given by the ambiguous behavioral statement s_2. Depending on the interpretation of s_2, there either exists a strict order or an interleaving order relation between a_2 and a_3, i.e. $R(a_2, a_3) = \{\rightsquigarrow, \|\}$. The relation $a_2 \rightsquigarrow_{t_1} a_3$ from t_1 is subsumed by both possible interpretations included in the behavioral space, since $\text{sub}(\rightsquigarrow, \rightsquigarrow)$ and $\text{sub}(\rightsquigarrow, \|)$ are both satisfied. Therefore, t_1 is compliant to all interpretations in $\mathcal{BI}$ and, thus, has a support value of 1.0. By contrast, while $a_2 \rightsquigarrow_{t_2}^{-1} a_3$ in trace t_2 is subsumed by relation $a_2 \| a_3$, this relation is not subsumed by $a_2 \rightsquigarrow a_3$. Therefore, t_2 does not comply to half of the behavioral interpretations in $\mathcal{BI}$. This results in $\text{supp}(t_2, \mathcal{S}) = 0.5$.

Aside from providing information on the (fraction of) behavioral interpretations to which a trace is compliant, behavioral spaces allow us to obtain further diagnostic information from this compliance check. In particular, we can utilize the function δ, which relates behavioral statements to relations, to gain insights into the conditions under which a trace is compliant to a process description. For example, we can learn under which interpretations of the statement s_3, "*In the meantime, the financial department takes care of the payment*", trace t_3 is compliant. In t_3, the financial department pays the settlement amount (a_{14}) before the claims officer records the settlement information (a_{12}). This complies with one of two interpretations of statement s_3 and, therefore, results in a support value of 0.5. Furthermore, we know that this trace is compliant, if and only if "*in the meantime*" means "*while the claims officer is performing its tasks*" and not "*while the claims officer is archiving the claim*". Such diagnostic information can be useful when interpreting the support values for a trace or when aiming to resolve the ambiguity contained in a textual description.

6 Evaluation

To demonstrate the importance of behavioral spaces for automated reasoning about textual process descriptions, we conduct a quantitative evaluation that assesses the impact of behavioral ambiguity on compliance checking. The goal of this evaluation is to learn how well behavioral spaces provide a balance between loose and restricted ways of dealing with behavioral ambiguity. In Section 6.1, we introduce the test collection used for the evaluation. Section 6.2 describes the details of the evaluation setup. Finally, we present and discuss the evaluation results in Section 6.3.

6.1 Test Collection

To perform the evaluation, we use the collection of textual process descriptions from the evaluation of the text to process model generation approach by Friedrich et al [6]. The collection contains 47 process descriptions obtained from various industrial and scholarly sources. The included texts differ greatly in size, ranging from 3 to 40 sentences. Furthermore, they differ in the average length of sentences and in terms of how explicitly and unambiguously they describe process behavior. Among others, this results from the variety of authors that created the textual descriptions. Hence, we believe that the collection is well-suited for achieving a high external validity of the results.

6.2 Setup

To conduct the evaluation, we implemented a prototype to generate behavioral spaces from textual process descriptions. To achieve this, we build on the state-of-the-art text to process model generation approach by Friedrich et al. [6]. In particular, the Java prototype builds on a library that is part of the RefMod-Miner[1], which implements a process model generation approach in a stand-alone tool. We use the library to automatically identify activities and extract behavioral profile relations that exist between the activities. Subsequently, we identify and remove those behavioral relations that result from ambiguous behavioral statements. Instead, we replace these relations by generating a behavioral space with the different possible interpretations, following the approach described in Section 4.

To demonstrate the importance of behavioral spaces, we compare the behavior they capture to two alternative ways of dealing with behavioral ambiguity. On the one end of the spectrum, instead of capturing behavioral ambiguity, a possibility is to focus only on the behavioral relations that can be extracted with certainty. For unclear behavioral relations, we take the least restrictive relation, i.e. the interleaving order. We shall refer to the behavioral profile that implements this way of dealing with behavioral ambiguity as a *minimally restricted* behavioral model. On the opposite end, it is possible to impose assumptions on ambiguous statements, resulting in a single interpretation of the described behavior. This is the approach that text-to-process-model generation techniques use to deal with behavioral ambiguity. We refer to this as a *fully interpreted* behavioral model. Together with a behavioral space, we therefore generate three behavioral models for each of 47 textual process descriptions:

[1] http://refmod-miner.dfki.de

1. **Minimally restricted behavioral profile:** This behavioral profile only captures the behavioral relations that can be extracted with certainty from the textual process description, i.e. we removed all behavioral relations obtained by the process model generation algorithm from [6] that result from ambiguous behavioral statements. We refer to the minimally restricted behavioral profile of a text T as BP_T^{min}.
2. **Fully interpreted behavioral profile:** The behavioral profile that is extracted from the process model generated by the process model generation approach from [6]. We refer to the fully interpreted behavioral profile of a text T with BP_T^{full}.
3. **Behavioral space:** The behavioral space generated for the textual description in accordance with the interpretation generation method described in Section 4. We refer to the behavioral space of a text T as $\mathcal{S}_T$.

The goal of the evaluation is to show that a behavioral space provides a balance between the minimally restricted model BP_T^{min}, which takes an agnostic view on ambiguous statements, and a fully restricted behavioral profile BP_T^{full}, obtained by imposing assumptions to arbitrarily settle behavioral ambiguity. We illustrate this by comparing the size of the sets of traces that are (potentially) compliant with the three behavioral models, in accordance to the definitions provided in Section 5.[2] Using $C(BM)$ to refer to the collection of traces that are compliant or potentially compliant to a behavioral model BM, we quantify the differences using for a textual description T using the following two metrics:

$$R_1(T) = \frac{|\, C(\mathcal{S}_T) \,|}{|\, C(BP_T^{full}) \,|} \qquad (2)$$

$$R_2(T) = \frac{|\, C(BP_T^{min})) \,|}{|\, C(\mathcal{S}_T) \,|} \qquad (3)$$

R_1 quantifies the ratio between the number of traces allowed by a behavioral space and a minimally restricted behavioral profile. Its purpose is to illustrate how much behavior that certainly does not conform to the business process description, is allowed by a model that ignores statements with behavioral ambiguity. R_2 quantifies the ratio between the number of traces allowed by a behavioral space and those allowed by a fully interpreted behavioral profile. Its purpose is to illustrate how much behavior that is not unequivocally non-compliant to a process specification, is removed from consideration when imposing assumptions on the interpretation of a textual process description.

6.3 Results

Table 3 summarizes the evaluation results for the textual process descriptions with behavioral ambiguity. The first interesting thing to note is how common textual process descriptions with behavioral ambiguity are. In total, 32 of the 47 textual process descriptions (70%) contained one or more ambiguous phrases. The majority, 28 cases, included just phrases with type ambiguity. Four cases contain statements with scope ambiguity, 3 of which also contain behavioral statements with type ambiguity.

[2] For processes that contain loops, we only include traces with at most one repetition.

Table 3: Evaluation results

Collection	**P**	$\mathbf{S_{type}}$	$\mathbf{S_{scope}}$	**A**	$\lvert\mathcal{BI}\rvert$	$\mathbf{R_1}$	$\mathbf{R_2}$
Only type ambiguity	28	64	0	19.6	11.0	100.0%	37.8%
With scope ambiguity	4	13	4	24.0	76.5	16.4%	0.5%
Total	32	77	4	20.2	19.1	89.5%	33.7%

Legend: P = number of processes, S_{type} = statements with type ambiguity, S_{scope} = statements with scope ambiguity, A = extracted activities per process (avg.), $|\mathcal{BI}|$ = interpretations per behavioral space.

For processes with just type ambiguity in their descriptions, there is a clear difference between the behavior allowed by fully interpreted behavioral profiles $C(BP^{full})$ and the behavior allowed by behavioral spaces $C(\mathcal{S})$. As indicated by metric R_2, the fully interpreted behavioral profiles allow for only 37.8% of the behavior allowed by the behavioral space. The remaining 62.2% represent traces for which it *cannot* be said with certainty that these do not comply to the process described in the text. This difference results from ordering restrictions that the text-to-model generation algorithm imposes on activities, even when these ordering restrictions may not exist. Behavioral spaces do not impose such restrictions and, thus, mark traces that exhibit such execution flexibility as potentially compliant. Though these cases already illustrate the impact of imposing assumptions on the interpretation of textual process descriptions, this impact is much more severe for cases that also contain statements with scope ambiguity.

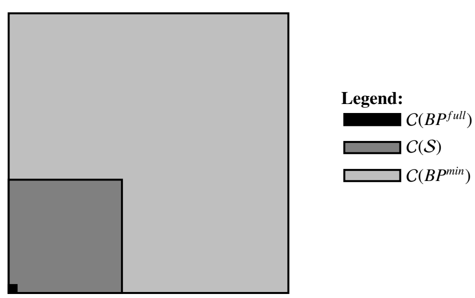

Fig. 4: Visualization of three sets of compliant traces for cases with scope ambiguity.

The behavioral models for the 4 cases with scope ambiguity show considerable differences among the behavior they allow. We visualize the relative sizes of the three sets of compliant traces in Figure 4. There, the light-gray area denotes the set of traces compliant with BP^{min}, i.e. the set of traces that remain when treating ambiguous statements as undecidable. The behavior allowed by the behavioral space, represented by the dark-gray area, is considerably smaller, as also indicated by the R_1 score of 16.4%. This number reveals that 83.6% of the traces in $C(BP^{min})$ represent traces that are not compliant with any reasonable interpretation of the statements with scope ambiguity. For instance, for the running example, this set would include traces where the financial department pays a settlement for an insurance claim, *before* the claim has been accepted. Figure 4 also shows the considerable impact that the usage of single interpretations has on the number of compliant traces. The tiny black area in the figure and the R_2 score of 0.5% indicate that, for the cases with scope ambiguity, the fully interpreted behavior profiles allow for only a very small fraction of the behavior that is (potentially) compliant to a behavioral space. Again, the remaining 99.5% represent traces that do not with certainty conflict with behavior specified in a textual process description.

The evaluation results show the impact both of ignoring ambiguous statements and of imposing single interpretations on them. As visualized by Figure 4, behavioral spaces provide a balance between these loosely restricted and too restricted behavioral models. In summary, behavioral spaces exclude a large number of nonsensical traces that can be excluded by generating proper interpretations for ambiguous statements. Still, they allow for much more traces than the restricted models obtained by imposing assumptions on the ambiguous statements in textual descriptions.

A point to consider for these evaluations results is that some of the statements with type ambiguity are ambiguous to automated approaches, but not for human interpreters. For instance, the meaning of the phrase "*sign and send contract*" can be inferred by human readers, because of the implicit order that exists between *signing* and *sending* of a document. Nevertheless, the decision to treat such statements as ambiguous for automated approaches is justified, because state-of-the-art automated approaches do not succeed in making such inferences.

7 Related Work

The work presented in this paper primarily relates to two major research streams: the analysis of textual process descriptions and the representation of data uncertainty.

The majority of works that consider the analysis of textual process models and other texts related to business processes, focus on the automated derivation of process models from natural language texts. Such techniques have been designed for textual process descriptions [6,7], group stories [8], use case descriptions [19] and textual methodologies [21]. Out of these, the text-to-process-model generation techniques by Friedrich et al. [6], on which we build our prototype and use as benchmark in our evaluation, is recognized as the state-of-the-art [16]. Although these works do not mention the problem of behavioral ambiguity explicitly, all of the presented techniques impose assumptions on the interpretation of ambiguous behavioral statements. This results in a single interpretation, i.e. a process model, for a text. However, as shown in the evaluation, this comes at the great disadvantage that the behavior allowed by this representation is much more strict than the behavior specified in the textual description. Our earlier work on the comparison of textual process descriptions to process models [2], faces similar issues when reasoning about the consistency of the two artifacts.

Similar to behavioral ambiguity inherent to natural language descriptions, uncertain data is also inherent to other application contexts. In the cases, uncertainty can be caused by, among others, data randomness, incompleteness, and limitations of measuring equipment [13]. This has created a need for algorithms and applications for uncertain data managements [4]. As a result, the modeling of uncertain data has been studied extensively, cf. [3,9,14,17]. Our notion of a behavioral space builds on concepts related to those used in uncertain data models. For instance, similar to the behavioral interpretations captured in a behavioral space, the model presented by Das Sarma et al. [17] uses a set of *possible instances* to represent the spectrum of possible interpretations for an uncertain relation. Furthermore, the model described in [3] uses conditions to capture dependencies between uncertain values. This notion has the same result as the sets of behavioral relations we derive from uncertain behavioral statements and convert into different behavioral interpretations. Still, the technical aspects and application contexts of these uncertain data models, mostly querying and data integration [4], differ considerably from the process-oriented view of behavioral spaces.

8 Conclusions

In this paper, we introduced the concept of a behavioral space to deal with the ambiguity in textual process descriptions. A behavioral space captures all possible interpretations of a textual process description and thus avoids the issue of focusing on a single process-oriented interpretation of a text. We demonstrated that a behavioral space is a useful concept for reasoning about a process described by a text. In particular, we used a quantitative evaluation with a set of 47 textual process descriptions and a compliance checking setting to illustrate that a behavioral space strikes a reasonable balance between ignoring ambiguous statements and imposing fixed interpretations on them.

While we defined the behavioral space concept based on textual process descriptions, we would like to point out that its use is not limited to texts. A behavioral space can help to capture the full behavior of different types of process descriptions that contain (ambiguous) natural language text. Consider, for instance, process models containing activities that describe several streams of actions by using ambiguous behavioral statements such as “*and*”. It has been found that such *non-atomic* activities can result in different interpretations of how to properly execute the process [15]. A behavioral space is also useful for application scenarios beyond compliance checking. Among others, it can serve as a basis for computing process similarity and conducting process matching.

In future work, we set out to explore these usage scenarios of behavioral spaces in more detail. What is more, we plan to investigate how we can prune a behavioral space in a systematic fashion.

References

1. Van der Aa, H., Leopold, H., Mannhardt, F., Reijers, H.A.: On the fragmentation of process information: Challenges, solutions, and outlook. In: Enterprise, Business-Process and Information Systems Modeling, pp. 3–18. Springer (2015)
2. Van der Aa, H., Leopold, H., Reijers, H.A.: Detecting inconsistencies between process models and textual descriptions. In: Business Process Management, pp. 90–105. Springer (2015)
3. Abiteboul, S., Kanellakis, P., Grahne, G.: On the representation and querying of sets of possible worlds, vol. 16. ACM (1987)

4. Aggarwal, C.C., Yu, P.S.: A survey of uncertain data algorithms and applications. Knowledge and Data Engineering, IEEE Transactions on 21(5), 609–623 (2009)
5. Dijkman, R.M., Dumas, M., García-Bañuelos, L.: Graph matching algorithms for business process model similarity search. In: Business Process Management, pp. 48–63 (2009)
6. Friedrich, F., Mendling, J., Puhlmann, F.: Process model generation from natural language text. In: Advanced Information Systems Engineering. pp. 482–496. Springer (2011)
7. Ghose, A., Koliadis, G., Chueng, A.: Process discovery from model and text artefacts. In: Services, 2007 IEEE Congress on. pp. 167–174. IEEE (2007)
8. Gonçalves, J.C.d.A., Santoro, F.M., Baiao, F.A.: Business process mining from group stories. In: Computer Supported Cooperative Work in Design, 2009. CSCWD 2009. 13th International Conference on. pp. 161–166. IEEE (2009)
9. Imieliński, T., Lipski Jr, W.: Incomplete information in relational databases. Journal of the ACM (JACM) 31(4), 761–791 (1984)
10. Leopold, H., Mendling, J., Polyvyanyy, A.: Supporting process model validation through natural language generation. IEEE Transactions on Software Engineering 40(8), 818–840 (2014)
11. Leopold, H., Pittke, F., Mendling, J.: Automatic service derivation from business process model repositories via semantic technology. Journal of Systems and Software 108, 134–147 (2015)
12. Liu, Y., Muller, S., Xu, K.: A static compliance-checking framework for business process models. IBM Systems Journal 46(2), 335–361 (2007)
13. Pei, J., Jiang, B., Lin, X., Yuan, Y.: Probabilistic skylines on uncertain data. In: Proceedings of the 33rd international conference on Very large data bases. pp. 15–26 (2007)
14. Peng, L., Diao, Y.: Supporting data uncertainty in array databases. In: ACM SIGMOD International Conference on Management of Data. pp. 545–560. ACM (2015)
15. Pittke, F., Leopold, H., Mendling, J.: When language meets language: Anti patterns resulting from mixing natural and modeling language. In: Business Process Management Workshops. pp. 118–129. Springer (2014)
16. Riefer, M., Ternis, S.F., Thaler, T.: Mining process models from natural language text: A state-of-the-art analysis. In: Multikonferenz Wirtschaftsinformatik (MKWI-16), March 9-11, Illmenau, Germany. Universität Illmenau (2016)
17. Sarma, A.D., Benjelloun, O., Halevy, A., Widom, J.: Working models for uncertain data. In: 22nd International Conference on Data Engineering. pp. 7–7. IEEE (2006)
18. Selway, M., Grossmann, G., Mayer, W., Stumptner, M.: Formalising natural language specifications using a cognitive linguistic/configuration based approach. Information Systems 54, 191–208 (2015)
19. Sinha, A., Paradkar, A.: Use cases to process specifications in Business Process Modeling Notation. In: IEEE International Conference on Web Services. pp. 473–480 (2010)
20. Smirnov, S., Weidlich, M., Mendling, J.: Business process model abstraction based on behavioral profiles. In: Service-Oriented Computing, pp. 1–16. Springer (2010)
21. Viorica Epure, E., Martin-Rodilla, P., Hug, C., Deneckere, R., Salinesi, C.: Automatic process model discovery from textual methodologies. In: Research Challenges in Information Science (RCIS), 2015 IEEE 9th International Conference on. pp. 19–30. IEEE (2015)
22. Weidlich, M., Mendling, J., Weske, M.: Efficient consistency measurement based on behavioral profiles of process models. IEEE Transactions on Software Engineering 37(3), 410–429 (2011)
23. Weidlich, M., Polyvyanyy, A., Desai, N., Mendling, J., Weske, M.: Process compliance analysis based on behavioural profiles. Information Systems 36(7), 1009–1025 (2011)