A Characterisation of Ambiguity in BPM

Marco Franceschetti[1][0000-0001-7030-282X], Ronny Seiger[1][0000-0003-1675-2592], Hugo A. López[2][0000-0001-5162-7936], Andrea Burattin[2][0000-0002-0837-0183], Luciano García-Bañuelos[3][0000-0001-9076-903X], and Barbara Weber[1][0000-0002-6004-4860]

[1] Institute of Computer Science, University of St.Gallen, St.Gallen, Switzerland
{marco.franceschetti | ronny.seiger | barbara.weber}@unisg.ch
[2] Technical University of Denmark, Kgs. Lyngby, Denmark
{hulo | andbur}@dtu.dk
[3] Department of Computer Science, Tecnológico de Monterrey, Puebla, Mexico
luciano.garcia@tec.mx

Abstract. Business Process Management is concerned with *process-related artefacts* such as informal specifications, formal models, and event logs. Often, these process-related artefacts may be affected by ambiguity, which may lead to misunderstandings, modelling errors, non-conformance, and incorrect interpretations. To date, a comprehensive and systematic analysis of ambiguity in process-related artefacts is still missing. Here, following a systematic development process with strict adherence to established guidelines, we propose a taxonomy of ambiguity, identifying a set of concrete ambiguity types related to these process-related artefacts. The proposed taxonomy and ambiguity types help to detect the presence of ambiguity in process-related artefacts, paving the road for improved processes. We validate the taxonomy with external process experts.

Keywords: Ambiguity · Business Process Management · Taxonomy.

1 Introduction

The Business Process Management (BPM) lifecycle iterates through a number of phases, each operating on different representations of a process. These representations include informal process specifications, formal process models, and event logs [15]; hereinafter, we refer to them as *process-related artefacts*. The presence of ambiguity in these process-related artefacts, however, might undermine the success of the BPM initiative these artefacts are part of [1].

According to the Cambridge dictionary[4], ambiguity is *"a situation or statement that is unclear because it can be understood in more than one way"*. In the context of software, authors in [13] relate ambiguity to the existence of multiple possible interpretations (e.g., of a software requirements specification). In the context of BPM, ambiguity is a quality issue (cf. [27]) that can be found in various process-related artefacts, namely informal specifications, formal models,

[4] https://dictionary.cambridge.org/dictionary/english/ambiguity

and event logs, yielding multiple artefact interpretations. Nevertheless, sometimes ambiguity may be the result of a deliberate choice to allow multiple interpretations, for instance, to avoid overly complicated models, or to guarantee flexibility in the application of rules and principles in legal systems, or to facilitate explorative BPM initiatives [20,24,39]. Here, we regard a process-related artefact as ambiguous if it admits multiple interpretations, and regard ambiguity as an artefact characteristic making it ambiguous. Ambiguity has high relevance in BPM; surprisingly enough, it has received only marginal attention to date.

Prior works studied ambiguity in the aforementioned process-related artefacts to some extent: [4] analysed it in user stories used to elicit process requirements, while [1,43] studied it in textual process descriptions. Ambiguity emerging when comparing a process model against its specification was studied in [5]. Authors analysed the manifestation of ambiguity in event logs in the form of uncertainty in [35] and of imperfection patterns in [44]. These previous works acknowledged the presence of ambiguity in process-related artefacts; however, they focused on some particular manifestation of ambiguity, such as linguistic ambiguity in textual process descriptions in [1], and lack a comprehensive and systematic analysis of ambiguity in relation to the different process-related artefacts.

The goal of this work is to reach a first characterisation of ambiguity. From a conceptual standpoint, this characterisation helps to better understand the notion of ambiguity in process-related artefacts and expose its relation with these artefacts. From an operational standpoint, the characterisation indicates process designers and analysts where to look for the presence of which forms of ambiguity in relation to the specific artefact. The main advantage is that it becomes easier to detect the presence of ambiguity in the artefacts, pinpointing potential misunderstandings, modelling errors, non-conformance, incorrect interpretations, as well as the risk of cascading ambiguities across the BPM lifecycle. It also becomes possible to systematically define general remedies rather than remedies specific to particular manifestations of ambiguity, which prior works do (cf. [4]).

Here, we address two research questions: *RQ1:* Where and in what form might ambiguity emerge in process-related artefacts? *RQ2:* What are potential causal relations between ambiguities observed in different process-related artefacts?

To answer *RQ1*, we propose a characterisation of ambiguity in the form of a taxonomy, whose purpose is to identify where specific ambiguity types may emerge in process-related artefacts. The taxonomy was built with a rigorous adherence to the taxonomy development guidelines for Information Systems research following the Design Science Research paradigm proposed in [29] and was evaluated by BPM experts. To answer *RQ2*, we present a relational characterisation of ambiguity across various process-related artefacts in the BPM lifecycle, highlighting potential causal relations. Our results enable further studies more focused on specific ambiguity types and on developing disambiguation strategies.

In Section 2, we establish the scope of our study revising process-related artefacts; in Sect. 3, we present the methodology and the resulting ambiguity characterisation; in Sect. 4, we report on the evaluation; in Sect. 5, we discuss implications; in Sect. 6, we discuss related work; Sect. 7 concludes the paper.

2 Process-Related Artefacts

We contextualise our study of ambiguity in the process-related artefacts describing processes managed in the BPM lifecycle. The BPM lifecycle is composed of the phases of process identification, process discovery, process analysis, process redesign, process implementation, and process monitoring and controlling [15]. We recall the artefacts describing a process usually associated with these phases on which we focus: informal specifications, formal models, and event logs [15].

Typically, a number of artefacts describing (fragments of) a process, data, requirements, key performance indicators, and goals are involved in the identification, discovery, and analysis phases [15]. Some of these artefacts might be preexisting; others might be produced in these phases, e.g., as the outcome of a workshop. Due to heterogeneity in sources, viewpoints, concerns, and formats and despite reconciliation efforts, these artefacts might exhibit ambiguity [25]; however, they are out of the scope of this study, since they are not a process description. Nevertheless, from these artefacts, analysts may distil an *informal* process specification in natural language (cf. [15]), which, as we study here, is an artefact potentially exhibiting ambiguity, too [1,7]. Extending the definition in [13], we call an informal process specification ambiguous if it admits multiple alternative interpretations in the form of process models, each model being consistent with the specification but mutually incompatible with any other model.

To facilitate communication or as a result of automated discovery techniques, an outcome of the aforementioned phases may also be a *formal* process model, represented in a formal language (e.g., Business Process Model and Notation, BPMN [2]). A formal process model is also the outcome of the phases of the process (re-)design and implementation [15]. These phases are concerned with enabling the enactment of the process, hence the resulting executable formal model may include additional information to support deployment to and execution by a BPM system (BPMS) [45]. As prior work indicates, ambiguity might emerge also in formal process models [36]. We say that a formal process model is ambiguous if elements in one or more of its perspectives can be interpreted in more than one way, also regarding the operational semantics [12]. For an example found in a publicly available process dataset, a gateway with a non-exclusive condition followed by activities whose labels indicate mutual exclusion can be interpreted in several ways, as confirmed by the evaluation reported in Section 4.

The lifecycle phase of process monitoring and controlling is related to the enactment of process model instances; it refers to tasks of analysis of data describing these enactments such as conformance checking [11]. The phase of process discovery may also analyse these data with automated approaches for process mining [15]. Typically, these tasks are performed on event logs, i.e., collections of timestamped events that occurred in the process enactments. In Sect. 3, we will show that also event logs might exhibit ambiguity. We say that an event log is ambiguous if it admits multiple interpretations of how the process enactment unfolded. For an example used in the evaluation, an event log may lead to different interpretations if the event timestamps have a too-coarse granularity, admitting multiple possible orderings of events, assuming an unordered log.

3 Characterisation of Ambiguity

3.1 Methodology

In order to reach a characterisation of ambiguity fostering an understanding of its relation with process-related artefacts, we developed a taxonomy of ambiguity following the principles of Design Science Research (DSR). Such an endeavour requires a sound and rigorous methodological approach, hence we selected the taxonomy development guidelines in Information Systems research recently proposed in [29], which revise and extend the well-known and widely adopted ones originally proposed in [34], as the methodological framework for our work. Accordingly, we followed an iterative development process, which started by establishing the following definitions as prescribed by the guidelines:

Observed phenomenon: Ambiguities in BPM have been investigated by previous research works, however, each focuses on some specific ambiguity manifestation without a systematic analysis. Thus, a structured characterisation of ambiguity is missing. This characterisation has the potential of building a common understanding of ambiguity and supporting the further development of strategies for managing ambiguities in BPM.

Taxonomy purposes: Here, we propose a characterisation of ambiguity in the form of a taxonomy whose purposes are: (i) to shed light on various possible types of ambiguity that might affect informal process specifications, formal models, and event logs; (ii) to support the detection and identification of these types of ambiguity in process-related artefacts in BPM.

Target user groups: We expect that process designers and process analysts will benefit from the proposed taxonomy by gaining a clearer understanding of which types of ambiguity emerge in various process-related artefacts, which inter-dependencies exist between these ambiguity types, and which particular elements they affect. In turn, this might help to define type-specific strategies for managing ambiguities during BPM tasks such as modelling, conformance checking, and process discovery.

Meta-characteristic: Defining the meta-characteristic is crucial for taxonomy development since it is the most comprehensive characteristic supporting the identification of characteristics and dimensions, which reflect the taxonomy purpose [29]. With our study, we aim at identifying potential sources of ambiguity in process-related artefacts, i.e., where ambiguity might be observed. Thus, we defined the source of ambiguity as the meta-characteristic of the proposed taxonomy. Our choice is motivated by the stance that for properly dealing with ambiguity it is essential to know where it might emerge.

Building approach: Following [29], two non-mutually exclusive taxonomy building approaches exist: empirical-to-conceptual and conceptual-to-empirical. The former is more suitable when the taxonomy designers have limited domain knowledge, but have a large number of concrete cases to analyse and abstract from. The latter is more suitable when the taxonomy designers are knowledgeable in the taxonomy domain and do not require a large number of concrete cases to analyse; concrete cases can be used to validate the taxonomy. Based on the six authors' expertise in the BPM domain, the proposed taxonomy was constructed following a conceptual-to-empirical approach for most iterations, with intermediate empirical-to-conceptual iterations to validate newly introduced concepts. Publicly available collections of process-related artefacts were used for these empirical-to-conceptual iterations, as well as for the evaluation (cf. Section 4).

Ending conditions: The guidelines in [29] define both objective and subjective ending conditions, which collectively establish the completion of the taxonomy development process. According to the guidelines, objective ending conditions state that the taxonomy encodes a mutually exclusive and collectively exhaustive classification and that stability (a fixpoint) is reached within the development iterations. Subjective ending conditions state that the taxonomy is concise, robust, comprehensive, extendable, and explanatory.

Evaluation goals: After meeting the ending conditions (end of the development phase), an evaluation should be performed with the goal of determining the usefulness of the taxonomy. The taxonomy should clearly describe ambiguity for the target users, and it should facilitate the identification of ambiguity types in concrete use cases. We will report on the evaluation in Sect. 4.

The proposed taxonomy was reached after 8 iterations (cf. [18]), each of which incrementally refined the taxonomy. Each iteration involved and was evaluated by different, disjoint subsets of the authors. The iterative process ended when it was ultimately agreed by all authors that no structural or terminological changes were required anymore and that the subjective ending conditions were met.

3.2 Ambiguity Taxonomy

The taxonomy development process resulted in the taxonomy shown in Figure 1. In line with the discussion in Sect. 2, we identify three main classes of artefacts in which ambiguity might emerge and which correspond to the first level of the taxonomy. The first class of artefacts is that of unstructured representations of a process, such as requirements documents, laws, guidelines, and informal specifications in natural language. Here, ambiguity leads to multiple possible interpretations of the process, hence multiple possible process models: we refer to this ambiguity as *descriptive* ambiguity. The second class of artefacts comprises representations of the process model in (possibly executable) formal languages. Here, ambiguity leads to multiple possible interpretations of the model semantics: in this case, we have *representational* ambiguity. The third class of artefacts comprises event logs, in which ambiguity leads to multiple possible interpretations of the executed process: in this case, we have *observational* ambiguity. We now discuss ambiguity in detail and provide brief yet focused examples for ambiguity types we identified; larger examples are available in [19].

Descriptive ambiguity relates to characteristics of the specification of a process in natural language, which lead to multiple interpretations of the process by a reader. More specifically, descriptive ambiguity may be determined by *linguistic* ambiguity or *epistemic* ambiguity.

Linguistic ambiguity emerges from lexical, syntactic, semantic, pragmatic ambiguity, or vagueness in the language constructs forming the process specifi-

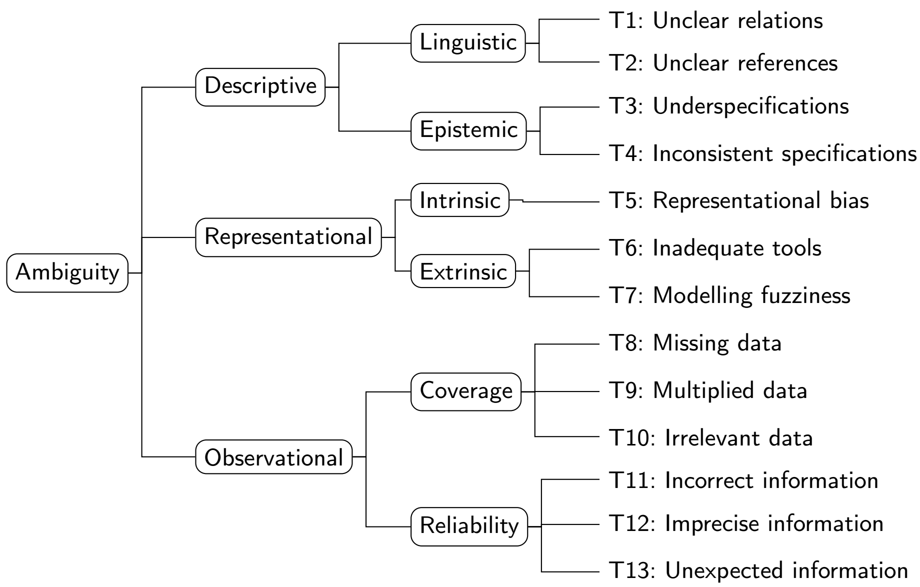

Fig. 1. Ambiguity taxonomy with possible ambiguity types as leaves of the tree

cation [23], as well as polysemy. Generalising the findings in [1], which studies linguistic ambiguity with respect to the specification of the control flow only, we identify *unclear relations* and *unclear references* as possible linguistic ambiguity types occurring in a specification and affecting various process perspectives, e.g., functional or organisational.

– **T1: Unclear Relations:** A process specification may fail to express in a sufficiently clear manner the relations between some process elements. The process model resulting from such a specification may include or exclude constructs or concepts, in contrast to the intended model. For instance, the guidelines for the process of hypokalaemia treatment in [41] state: *"Treat any underlying cause (. . .) and/or review medication."* With this specification, it is unclear whether between the tasks of treatment and medication reviewing there exists a precedence, mutually exclusive or parallel execution relation.
– **T2: Unclear References:** In an informal process specification, it might be unclear what relevant process element is being described by a given statement. The resulting process model may contain erroneous elements or concepts, lack relevant elements, or include elements not matching the specification. For instance, the hotel service process specification in [21] states: *"Eighty per cent of room-service orders include wine or some other alcoholic beverage."* Also considering the whole specification, the relation between this statement and any process element that might relate to the mentioned orders is unclear and might result in a model excluding such element, like in [37].

Epistemic ambiguity reflects an insufficient knowledge of the process or its domain when developing a process specification, which results in knowledge gaps or inconsistencies in various perspectives of the specification. These gaps and inconsistencies may lead to a number of possible interpretations when reading the specification. We identify *underspecifications* and *inconsistent specifications* as epistemic ambiguity types.

- **T3: Underspecifications:** The term underspecification refers to the omission of certain features from a representation [22]. In a process specification, it refers to the deliberate or unintentional exclusion of some characteristics or the partial specification of one or more process perspectives. Underspecifications may exist due to the need to cope with flexible process specifications without cluttering, limited domain or process knowledge, or to negligence. The resulting process model might exhibit some "conceptual gaps". For instance, the "MCT finalise SCT warrant possession" process studied in [21] states: *"After that, some other MC internal staff receives the physical SCT file (out of scope)."* Here, there is a deliberate omission of details around a resource, which makes it unclear who should receive the file and how this should be modelled.
- **T4: Inconsistent Specifications:** A process specification might present conflicting requirements, which cannot be satisfied altogether. In this case, either the modelling language allows including all such requirements into an inconsistent process model, or the process designer has to decide which of these requirements to retain, resp. to discard. Consider the excerpt from the specification of exercise 4 in [47], stating: *"If the combined design fails the test, then they are both sent back (. . .). If the designs pass the test, then they are deemed complete and are then sent to the manufacturing Process (. . .)."* Here, it is not clear whether the condition is based on a single compound data object or on two separate data objects, which requires deciding how to model both the data and the condition.

Representational ambiguity is associated with a formal process model expressed in some modelling language such as BPMN or Petri nets. It refers to the possibility that the process model is formalised in a way that leads to multiple different interpretations of its semantics. Note that here we do not consider erroneous formal models with invalid syntax or that could not be executed by a process engine. We consider syntactically valid formal models that can be executed, but whose execution or interpretation have uncertain semantics. Representational ambiguity may be *intrinsic* or *extrinsic.*

Intrinsic ambiguity refers to inherent characteristics of the modelling language that may enforce or prevent certain modelling constructs, patterns, and styles, which are in contrast to the specific modelling objectives. A possible type of intrinsic ambiguity is *representational bias*:

- **T5: Representational Bias:** Process modelling languages have intrinsic characteristics that may limit the expression of certain process elements, altering or curbing the process semantics. For instance, with classical Petri nets, it is impossible to model the organisational aspect of a process: the resulting model would lead to uncertainty in interpreting the assignment of tasks to roles if these were relevant to the process. Additional examples based on BPMN 2.0 can be found in [2].

Extrinsic ambiguity does not stem from characteristics of a specific modelling language but derives from the modelling task. This can be due to limitations of the modelling tool used, e.g., lack of support for certain modelling constructs – for which case *inadequate tools* is a possible ambiguity type. Alternatively, a lousy use of the modelling language may be attributed to the human process designers – here, *modelling fuzziness* is a possible ambiguity type.

- **T6: Inadequate tools:** Process modelling tools may have limitations in the support to modelling languages, excluding certain constructs allowed by a given language from a process model, or allowing the inclusion of constructs or relations forbidden by the language. For an example, consider Camunda Platform 8: it allows modelling BPMN 2.0 process models, but (currently) does not allow defining signal events. Using other events as a workaround might result in confusion in interpreting such events.
- **T7: Modelling fuzziness:** A process designer might deviate from established modelling best practices, producing process models that are syntactically valid, but that still exhibit uncertain semantics. For an example based on BPMN, consider that a designer may associate several activities with the same name. If these activities are not identical, their interpretation would be unclear and might induce one to consider them to be the same.

Observational ambiguity might affect the representation of a process execution, which is usually in the form of an event log describing activities, roles, and so on. In line with the criteria at the basis of the event log maturity levels indicated in [3], observational ambiguity may relate to both the completeness of data describing a process execution and the trustworthiness of the information such data conveys about the process execution. Therefore, we distinguish between *coverage* ambiguity and *reliability* ambiguity.

Coverage ambiguity refers to the presence, resp. absence, and the amount of data describing an execution, and may result from ambiguity types such as *missing data*, *repeated data*, and *irrelevant data*. Missing data and irrelevant data reflect general data quality issues affecting event logs identified in [10]. Repeated data does not refer to redundancy, i.e., multiple identical occurrences, but repetitions of the same data with variations in the values, which make it unclear which data values should be considered correct.

- **T8: Missing Data:** Process logs may be incomplete due to the absence of certain data in log entities, such as missing case id or timestamp. Missing data can be attributed to a number of reasons, such as nonconforming behaviour, faults in sensors generating process data, resource unavailability, or negligence. Additional examples of missing data are presented in [10].
- **T9: Multiplied Data:** An event log may contain multiple occurrences of the same event, i.e., of the same happening, with variations in the values of

some attributes. Possible causes might be non-conformance of the execution with the process model, or faults in the logging. For instance, in the log of a storage process instance used for the evaluation (see Sect. 4), an event for the single occurrence of an activity "Read colour" reading the colour of a workpiece occurs twice, first reporting that the workpiece is blue, then red.

- **T10: Irrelevant Data:** If no adequate filtering methods are adopted, more data than required is retained about the execution of a process, which may be erroneously considered relevant and lead to erroneous interpretations of the executed process [10]. For instance, the log for a smart factory process may include large amounts of environmental data continuously generated by sensors and stored as events: process discovery based on such an unfiltered log may result in a very large model cluttered with irrelevant activities.

Reliability ambiguity refers to the trust that can be put in the logged information describing a process execution. An event log might lead to multiple interpretations of how an execution unfolded if a trace does not conform to a known process model. It is also possible that the analysis of the traces alone, without any knowledge of the underlying process model, might lead to multiple interpretations. Here, possible ambiguity types are *incorrect information*, *imprecise information*, and *unexpected information*.

- **T11: Incorrect Information:** Information representing the execution of a process instance might be misaligned with the known process model, representing facts that do not hold true as per the model. Wrong interpretations of how the process unfolded are thus possible. For instance, in the log of a storage process used for the evaluation (see Sect. 4), an activity *"Store Workpiece"* is performed by resource *"VGR"*, while according to the corresponding BPMN process model used for the enactment, the activity is assigned to resource *"HBW"*. Thus, it is not clear whether a different resource took over, or there was a logging error.
- **T12: Imprecise Information:** Information may be recorded at a coarse granularity (e.g., due to data anonymisation), losing relevant information about a process execution. For instance, if events in an event log are recorded with minute precision, there might be uncertainty regarding the exact order of execution of consecutive activities, and one could infer several different traces. Additional examples of imprecise information are discussed in [10].
- **T13: Unexpected Information:** Logged execution information might deviate from the values expected as per the analysis of the process log, making it unclear how to interpret the execution against a discovered model. For example, consider the case of a monitored push-down hand sanitiser dispenser: the log usually reports an amount of 5–10ml of sanitiser per dispensing event; however, if an event with 20ml is recorded, it might be unclear whether multiple nurses used the dispenser together, or just one with double the amount.

3.3 Relational Characterisation

Ambiguity may propagate across the process-related artefacts, i.e., ambiguity affecting one artefact might induce cascading ambiguity in the same or other

Table 1. Relational characterisation of ambiguity assuming a transition from a specification to a model and from the model to an event log. → denotes potential direct causal relation, ⇝ indirect, ↻ reflexive.

	Descriptive	Representational	Observational
Descriptive	↻	→	⇝
Representational		↻	→
Observational			↻

artefacts. For a systematic analysis of these relations between ambiguities across artefacts, the artefacts need to be anchored to the respective BPM lifecycle phases and the transitions between the phases need to be established. Here, we give an example for this analysis assuming the case in which a given informal specification is used to design and implement an executable formal model, whose enactment in a BPMS generates an event log. Other cases (for instance, starting from an event log, deriving a formal model with process mining, and from this model generating a textual description) will be analysed in future work. Table 1 summarises the potential causal relations between descriptive, representational and observational ambiguities in this case.

We observe that each ambiguity might cause additional ambiguities of the same kind, which we indicate as *potential reflexive* causal relationship and denote with ↻. Descriptive ambiguity might cause representational ambiguity, and representational ambiguity might induce observational ambiguity: we refer to these as *potential direct* causal relationships (denoted with →). For transitivity, descriptive ambiguity might cause observational ambiguity, which we refer to as *potential indirect* causal relationship and denote with ⇝.

Descriptive ↻: This is the case, for instance, when a linguistic ambiguity causes epistemic ambiguity. Consider, for example, the excerpt from the description of the phylogenetic analysis process in [31]: *"Similarly, alignments were examined and investigated by an MP approach with heuristic search in MEGA"*. The relation between the data object *alignments* and its origin is not introduced in the specification: a *T1: Unclear Relations* ambiguity type. In turn, this causes the process fragment responsible for producing the data object as output to be underspecified: a *T3: Underspecifications* ambiguity type.

Descriptive → Representational: This is the case, for instance, when an inconsistent specification is translated into a formal process model affected by modelling fuzziness. Consider the following fragment of a process specification from [15]: *"(...) once the license is granted, this is sent by EPA directly to the applicant. (...) Once the required permit and/or license have been obtained, the assessment manager notifies the applicant of the final approval"*. Here, the ambiguity type in the informal specification is *T4: Inconsistent Specifications*, since the first sentence states that the applicant receives the license (not the manager); however, the second sentence states that the manager informs the applicant of the reception and subsequent approval. The second sentence implies that it is the manager who receives the permit and license, in contrast with the first sentence.

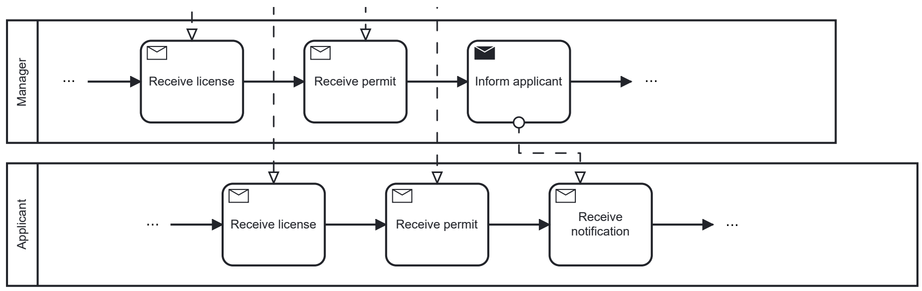

Fig. 2. BPMN fragment for a land development applications process from [15]

One may formalise this specification fragment as the BPMN fragment in Fig. 2, with the same *receive* tasks for both manager and applicant, which generates confusion in the interpretation of these tasks and the subsequent notification.

Representational ↻: This is easily observed, for instance, when ambiguity type *T6: Inadequate Tools* leads to *T7: Modelling Fuzziness*. For example, some BPMN modelling tools allow defining message flows connecting elements in different lanes of the same pool, making it unclear whether the lanes are meant to refer to different organisations, or whether it is the message flow to be incorrect.

Representational → Observational: This is the case when the execution of an ambiguous process model generates a log trace that can be interpreted in multiple possible ways. As an example, consider the case in which the same name is assigned to multiple different activities in a BPMN model (*T7: Modelling Fuzziness* ambiguity). The log describing the execution of an instance of this process model would include multiple events associated with the same name, one for each executed activity (*T9: Multiplied Data* ambiguity). Here, due to modelling fuzziness, it is unclear whether the same activity was executed several times, or the logged events refer to different activities with the same name.

Observational ↻: An example for this case is a coverage ambiguity causing a cascading coverage ambiguity. For instance, consider the case of a smart factory in which all case id attributes are missing from the log (*T8: Missing Data*) due to a malfunctioning of the communication bus during a given time period when parallel process instances were executed. This results in the impossibility of establishing the right activity-instance associations for all tasks carried out during the malfunctioning of the communication bus (*T12: Imprecise Information*).

4 Evaluation

The guidelines in [29] remark that it is not sufficient to evaluate a taxonomy *ex-ante* by assessing the ending conditions, but also an *ex-post* evaluation should be performed after the design process is completed. This evaluation checks *"based on the feedback of (potential) users whether the completed version of a taxonomy fulfils the sufficient condition and evaluation criteria to be a useful taxonomy"* [29].

4.1 Method, Participants, and Dataset

Several evaluation methods for taxonomies exist: for the proposed taxonomy, we deem the *illustrative scenario with real-world objects* the most suitable one. The method aims at demonstrating the usefulness of the taxonomy by applying it to synthetic or real-world situations (here: informal specifications, formal models, and event logs), and is the most frequently adopted evaluation method [29].

We involved four participants with several years of expertise in BPM; none of the authors acted as participants. Each participant had a training session, in which (s)he was educated on the taxonomy. Afterwards, the participant was asked to individually analyse 13 process-related artefacts–for each of which the authors had detected the presence of one ambiguity type–and to identify the specific ambiguity type with the help of the taxonomy, motivating the answer.

For a comprehensive evaluation, we needed a set of realistic examples from various domains, covering all artefacts and ambiguity types from Section 3; we considered only artefacts in English to accommodate the diverse nationalities of the participants. For informal specifications, we analysed the set of 47 pairs process specification–model studied in [21], which comes from academic and industrial sources and has been used in several other studies; we also analysed the process specifications from the exercises in [15], which is a well-known textbook in the BPM community. For formal models, we analysed the models from the BPM Academic Initiative in [46], which also comes from academia and industry and is well-known in the community. For event logs, we analysed the BPI Challenge 2012 log in [14], the Road Traffic Fine Management Process log in [30], and logs from a smart factory simulation environment [42]. In the end, we selected 13 artefacts, in each of which we identified a distinct ambiguity type from Sect. 3.

4.2 Results

The dataset, results, and demographics of participants are publicly available [19]. Out of 52 total identifications of ambiguity types by the participants, 43 matched those by the authors. The 43 matching identifications support the usefulness of the taxonomy. In the case of a mismatch, at most two participants disagreed with the authors' identified ambiguity type. Only in one case one participant firmly argued that there was no ambiguity, while another identified a different type. These mismatches demonstrate the possibility of different interpretations among the participants, indeed underpinning the presence of ambiguity.

What emerged from the discussions with the participants is that different interpretations may stem from the different mental models and frames. For example, a case of *T1: Unclear relations* between activities due to multiple possible interpretations of the term "and" (also identified in [1] as such), was identified by one participant as *T3: Underspecifications*, since in the participant's view the meaning of the term "and" was not further specified; on the other hand, another participant could not detect any ambiguity, with the motivation that the term "and" in that context necessarily denotes a sequential relation. For another example, when analysing a formal model specified in Event-driven Process Chain

(EPC) notation, one participant, for historic reasons, held an interpretation of EPC as an informal language, hence identified a descriptive (*T3: Underspecifications*) instead of a representational ambiguity (*T5: Representational bias*).

Overall, while the evaluation proved the usefulness of the taxonomy for identifying ambiguity in process-related artefacts, it also asserted the need to investigate, in future work, the role played by the mental model of the consumers (i.e., those supposed to use) of these artefacts in detecting and identifying ambiguity.

5 Discussion

The taxonomy proposed in Section 3 gave an answer to *RQ1* on where ambiguity might emerge in process-related artefacts. The discussion on the relational characterisation addressed *RQ2* on possible causal relations. Here, we outline some applications and further research directions, and discuss threats to validity.

5.1 Ambiguity Detection

In the spirit of open science, datasets composed of process-related artefacts are being increasingly shared and reused among the BPM community. However, there is the risk that they are used unaware of the potential presence of ambiguity. Reusing ambiguous processes in experiments without acknowledging and managing ambiguity poses a threat to the validity of these experiments. For example, the PET dataset from [7] is built by annotating the specifications from [21], which we have found to exhibit various ambiguity types. Indeed, in [7] the authors report on the need to discard some processes from the dataset due to the impossibility of reaching a consensus on the interpretation.

We propose our ambiguity taxonomy as a tool for analysing process-related artefacts and detecting ambiguity types in these datasets. Such a systematic analysis is beyond the scope of this paper; however, we expect this paper to trigger further analysis of existing datasets to improve the quality of future experimental evaluations based on these datasets. In general, we envision the proposed taxonomy to raise awareness of the presence of ambiguity in BPM, and to support new research directions such as prompt engineering for BPM.

5.2 Analysis of the Affected Elements

Analysing ambiguity in process-related artefacts in-depth requires examining which specific elements of these artefacts may be affected by ambiguity.

Descriptive ambiguity emerges in informal process specifications whose interpretation results in multiple possible process models. Thus, in order to measure the effect of descriptive ambiguity, it makes sense to determine which process model perspectives are affected by it. The BPM literature identifies several process perspectives, four of which are most agreed upon: control flow, data, organisational, and operational [38]. A systematic analysis should consider at least these perspectives; when relevant, also other perspectives might be considered. In our experience and preliminary observations, we found all descriptive ambiguity types identified in Sect. 3 to potentially affect each of these four perspectives.

Representational ambiguity affects formal process models, and results in different possible model interpretations. These interpretations are relative to the aforementioned process perspectives; hence we argue that these process perspectives should be the object of analysis of formal process models, too.

Observational ambiguity affects event logs, hence analysing its effects requires examining the affected event log entities. Prior work (cf. [44]) identified and studied the following event log entities: Case, Event, Belongs_to (event–case association), Case attribute, Position, Activity name, Timestamp, Resource, Event attributes. These entities constitute a good starting point for analysis.

An in-depth analysis of ambiguity should inspect the above-indicated elements in relation to the artefact at hand. In light of the relational characterisation discussed in Sect. 3, the analysis should also study the potential cascading effects of ambiguity across the elements of the process-related artefacts.

5.3 Ambiguity Reduction Strategies

A reduction of descriptive ambiguity in informal process specifications could be achieved combining different strategies. One possible strategy might be to adopt a controlled language (e.g., the *Attempto* controlled language [40]), as controlled languages impose restrictions on the available linguistic constructs, reducing the risk of linguistic ambiguity. Ontology annotations and glossaries have also been proposed to tackle descriptive ambiguity (e.g., in [4]). Additionally, involving domain experts could help to alleviate epistemic ambiguity.

To reduce representational ambiguity in formal models, it is critical to put great care into the modelling task, starting from the choice of the prospective modelling language [9]. This choice should consider criteria such as expressive power, tool support, and familiarity of the process designer. Additionally, assisted modelling (e.g., [17,33]) and checking (cf. [48]) approaches, as well as ontology-based modelling (cf. [16]) might help to reduce representational ambiguity.

Possible strategies to reduce observational ambiguity require employing *a priori* adequate methods and tools to record process executions comprehensively and faithfully. To this end, the recent idea of integrating Internet of Things (IoT) technologies with BPM to collect rich datasets (cf. [26]) shows great potential for disambiguation and is worth investigating.

The approaches outlined above are examples of possible unstructured strategies to reduce ambiguity. However, if dependencies between ambiguities across artefacts can be identified (cf. Sect. 3), more structured strategies may be achieved by exploiting these dependencies to achieve cascading reductions. For example, reducing descriptive ambiguity in an informal specification may lead to a cascading reduction of representational ambiguity in the formal model derived from the specification. A detailed study of type-specific reduction strategies and of more structured approaches exploiting dependencies will be addressed in future work.

5.4 Threats to Validity

The evaluation confirmed that all ambiguity types identified in the taxonomy are indeed ambiguities. Regarding the taxonomy structural completeness, it can be seen that it is met at the first level, since the taxonomy covers all the artefacts describing processes typically identified in the literature in relation to BPM lifecycle phases and process mining tasks [3,15] (cf. Sect. 2). New ambiguity types for these artefacts, which may be discovered in the future, could find a position in lower levels of the taxonomy in relation to the pertaining artefact, in line with the extensibility principle [29]. Regarding external validity and the generalisability of our findings, strictly adhering to the development guidelines in [29] ensures that the scope of applicability is clearly defined within BPM. The evaluation results indicate applicability and usefulness of the taxonomy in a broad set of domains, suggesting general applicability across domains where BPM is applied. As part of a first step towards a comprehensive understanding of ambiguity, the evaluation involved academics only: in follow-up work, we will involve practitioners for further evaluation, and also investigate the generalisability beyond BPM.

6 Related Work

Prior work recognised ambiguity as a quality issue [13,27]; here, we identified several ambiguity types as its manifestations in process-related artefacts. As not every quality issue is an ambiguity, these ambiguity types can be seen as a *proxy* for a subset of quality issues of process-related artefacts, i.e., analysts can investigate certain quality issues by detecting ambiguity types. For instance, completeness quality issues (cf. [6]) take the form of coverage ambiguity; issues in semantic validity (cf. the SEQUAL framework [27]) take the form of *T7: Modelling fuzziness*; incorrect data issues (cf. [10]) take the form of *T11: Incorrect information*. A systematic analysis of the relations between quality issues and ambiguity types is beyond the scope of this paper and invites further research.

Prior work studied ambiguity in informal documents to elicit requirements or to describe business processes. In [4], authors conduct a systematic literature review focusing on user stories to elicit requirements. They identify four ambiguity problems (vagueness, inconsistency, insufficiency, and duplicates), which can be related to the descriptive ambiguity types identified here. They also summarise proposed solutions to these problems, such as algorithmic solutions, ontologies, and controlled languages: while these are proposed to resolve particular manifestations of ambiguity, our taxonomy identifies ambiguity types as abstractions of particular manifestations, enabling designing type-specific resolution strategies.

The work in [1] studies what we identified here as linguistic ambiguity in textual process descriptions. Based on the concept of behavioural space, the authors design a technique to deal with ambiguity in the context of conformance checking; however, the scope is limited to the control flow. As Section 5 indicates, ambiguity might affect all major process perspectives: we foresee that by extending the concept of behavioural space, one might be able to deal with ambiguity in these perspectives. Ambiguity in textual descriptions is studied also in [43], where it is identified based on sentence templates, resulting in six ambiguity issues, which can all be related to the descriptive ambiguity types we identified.

The work in [8] presents a qualitative analysis of the state of the art in the task of process extraction from texts. Related to this, the work in [32] discusses challenges in the discovery of legal processes arising from the analysis of the natural language. Complementing these works with our study on ambiguity could result in further insights to guide the process of process extraction from text.

In line with our indications from Sect. 5, the work in [36] proposes an automated technique to assist the modelling task and resolve representational ambiguity due to activity labels. Representational ambiguity might result in inconsistencies between a process model and the corresponding specification: the work in [5] studies how to detect such inconsistencies with respect to role associations.

The work in [44] studies imperfection patterns in event logs with the goal of cleaning event logs for process mining. Imperfection patterns are quality issues, in line with the results of [10], which result in the observational ambiguity types presented here. By detecting these ambiguity types, analysts can discover the presence of imperfection patterns and quality issues. Potential ambiguity in object-centric event logs is highlighted in [28]: we expect this recent log format to benefit from our taxonomy for the identification of ambiguity, and to potentially extend the taxonomy with new ambiguity types. A related problem is uncertainty in process logs, i.e., the lack of precise knowledge about certain process aspects [35], e.g., in relation to task durations and event data. Here, we interpret uncertainty as a consequence of ambiguity, aligning with the work in [1,35].

7 Conclusion

Ambiguity in BPM can be found in various process-related artefacts, namely informal specifications, formal models, and event logs. In order to shed first light on ambiguity in these artefacts, we proposed a taxonomy of ambiguity, identifying 13 concrete ambiguity types in it; for each ambiguity type, we provided real examples. Additionally, we studied potential causal relations between ambiguities in relation to the affected artefacts and proposed a relational characterisation of ambiguity. An evaluation with process experts confirmed the usefulness of the proposed taxonomy. We regard these contributions as a tool for helping to detect the presence of ambiguity in process-related artefacts. Detecting ambiguity is the first step towards achieving increased quality of process-related artefacts.

In future work, we will perform further evaluations involving academics and practitioners. We further plan to study the effect of ambiguity on process-related artefact elements. Moreover, we plan to define ambiguity reduction strategies. We expect this work to foster further reflection on how to deal with unresolvable ambiguity, as well as on how to generalise the presented concepts beyond BPM.

Acknowledgements This work has received funding from the Swiss National Science Foundation under Grant No. IZSTZ0_208497 (*ProAmbitIon* project). The authors thank the anonymous reviewers and the evaluation participants.

References

1. van der Aa, H.: Dealing with ambiguity in textual process descriptions. In: Comparing and Aligning Process Representations, pp. 77–101. Springer (2018)
2. Aagesen, G., Krogstie, J.: Bpmn 2.0 for modeling business processes. In: Handbook on Business Process Management 1, pp. 219–250. Springer (2015)
3. van der Aalst, W.M.P., et al.: Process mining manifesto. In: BPM 2011 International Workshops, Revised Selected Papers, Part I. Lecture Notes in Business Information Processing, vol. 99, pp. 169–194. Springer (2011)
4. Amna, A.R., Poels, G.: Ambiguity in user stories: A systematic literature review. Information and Software Technology **145**, 106824 (2022)
5. Aysolmaz, B., Iren, D., Reijers, H.A., et al.: Detecting role inconsistencies in process models. In: ECIS 2019, Proceedings (2019)
6. Batini, C., Scannapieco, M.: Data Quality: Concepts, Methodologies and Techniques. Data-Centric Systems and Applications, Springer (2006)
7. Bellan, P., van der Aa, H., Dragoni, M., Ghidini, C., Ponzetto, S.P.: PET: an annotated dataset for process extraction from natural language text tasks. In: BPM 2022 International Workshops, Revised Selected Papers. Lecture Notes in Business Information Processing, vol. 460, pp. 315–321. Springer (2022)
8. Bellan, P., Dragoni, M., Ghidini, C.: A qualitative analysis of the state of the art in process extraction from text. DP@ AI* IA pp. 19–30 (2020)
9. Bork, D., Karagiannis, D., Pittl, B.: A survey of modeling language specification techniques. Information Systems **87**, 101425 (2020)
10. Bose, R.J.C., Mans, R.S., van der Aalst, W.M.: Wanna improve process mining results? In: 2013 IEEE symposium on computational intelligence and data mining (CIDM). pp. 127–134. IEEE (2013)
11. Carmona, J., van Dongen, B., Solti, A., Weidlich, M.: Conformance checking. Springer (2018)
12. Christiansen, D.R., Carbone, M., Hildebrandt, T.T.: Formal semantics and implementation of BPMN 2.0 inclusive gateways. In: WS-FM 2010. Revised Selected Papers. Lecture Notes in Computer Science, vol. 6551, pp. 146–160. Springer (2010)
13. Davis, A.M., Overmyer, S., Jordan, K., Caruso, J., Dandashi, F., Dinh, A., Kincaid, G., Ledeboer, G., Reynolds, P., Sitaram, P., Ta, A., Theofanos, M.: Identifying and measuring quality in a software requirements specification. In: METRICS 1993, Proceedings. pp. 141–152. IEEE Computer Society (1993)
14. van Dongen, B.: Bpi challenge 2012 (2012). https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f
15. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management. Springer (2018)
16. Fan, S., Hua, Z., Storey, V.C., Zhao, J.L.: A process ontology based approach to easing semantic ambiguity in business process modeling. Data Knowl. Eng. **102**, 57–77 (2016)
17. Ferreira, R.C.B., Thom, L.H., de Oliveira, J.P.M., Avila, D.T., dos Santos, R.I., Fantinato, M.: Assisting process modeling by identifying business process elements in natural language texts. In: ER 2017 Workshops, Proceedings. Lecture Notes in Computer Science, vol. 10651, pp. 154–163. Springer (2017)
18. Franceschetti, M., Seiger, R., Lopez, H., Burattin, A., Garcia-Banuelos, L., Weber, B.: Ambiguity taxonomy development iterations (Aug 2023), https://doi.org/10.5281/zenodo.8268074

19. Franceschetti, M., Seiger, R., Lopez, H., Burattin, A., Garcia-Banuelos, L., Weber, B.: A characterization of ambiguity in bpm (May 2023), https://doi.org/10.5281/zenodo.7944319
20. Freedman, D.S., Singer, B., Swain, F.S.: The regulatory flexibility act: Orienting federal regulation to small business. Dick. L. Rev. **93**, 439 (1988)
21. Friedrich, F., Mendling, J., Puhlmann, F.: Process model generation from natural language text. In: International conference on advanced information systems engineering. pp. 482–496. Springer (2011)
22. Frisson, S.: Semantic underspecification in language processing. Language and Linguistics Compass **3**(1), 111–127 (2009)
23. Gleich, B., Creighton, O., Kof, L.: Ambiguity detection: Towards a tool explaining ambiguity sources. In: International working conference on requirements engineering: foundation for software quality. pp. 218–232. Springer (2010)
24. Hall, J.M., Johnson, M.E.: When should a process be art, not science? Harvard business review **87**(3), 58–65 (2009)
25. Hoppenbrouwers, S., Proper, H.A., van der Weide, T.P.: A fundamental view on the process of conceptual modeling. In: ER 2005, Proceedings. Lecture Notes in Computer Science, vol. 3716, pp. 128–143. Springer (2005)
26. Janiesch, C., et al.: The internet of things meets business process management: a manifesto. IEEE Systems, Man, and Cybernetics Magazine **6**(4), 34–44 (2020)
27. Krogstie, J.: Quality in Business Process Modeling. Springer (2016)
28. Kumar, A., Soffer, P., Tsoury, A.: Normalizing object-centric process logs by applying database principles. Information Systems **115**, 102196 (2023)
29. Kundisch, D., Muntermann, J., Oberländer, A.M., Rau, D., Röglinger, M., Schoormann, T., Szopinski, D.: An update for taxonomy designers. Business & Information Systems Engineering **64**(4), 421–439 (2022)
30. de Leoni, M.M., Mannhardt, F.: Road traffic fine management process (2015). https://doi.org/10.4121/uuid:270fd440-1057-4fb9-89a9-b699b47990f5
31. Liang, L., Xu, B., Chen, Y., Liu, Y., Cao, W., Fang, L., Feng, L., Goodchild, M.F., Gong, P.: Combining spatial-temporal and phylogenetic analysis approaches for improved understanding on global h5n1 transmission. PLoS One **5**(10) (2010)
32. López, H.A.: Challenges in legal process discovery. In: ITBPM@ BPM. pp. 68–73 (2021)
33. López, H.A., Marquard, M., Muttenthaler, L., Strømsted, R.: Assisted declarative process creation from natural language descriptions. In: 23rd International Enterprise Distributed Object Computing Workshop. pp. 96–99. IEEE (2019)
34. Nickerson, R.C., Varshney, U., Muntermann, J.: A method for taxonomy development and its application in information systems. European Journal of Information Systems **22**(3), 336–359 (2013)
35. Pegoraro, M.: Probabilistic and non-deterministic event data in process mining: Embedding uncertainty in process analysis techniques. arXiv preprint arXiv:2205.04827 (2022)
36. Pittke, F., Leopold, H., Mendling, J.: Automatic detection and resolution of lexical ambiguity in process models. IEEE Trans. Software Eng. **41**(6), 526–544 (2015)
37. Rosa, L.S., Silva, T.S., Fantinato, M., Thom, L.H.: A visual approach for identification and annotation of business process elements in process descriptions. Computer Standards & Interfaces **81**, 103601 (2022)
38. Rosa, M.L., Dumas, M., ter Hofstede, A.H.M., Mendling, J.: Configurable multi-perspective business process models. Inf. Syst. **36**(2), 313–340 (2011)

39. Rosemann, M.: Explorative process design patterns. In: Business Process Management: 18th International Conference, BPM 2020, Seville, Spain, September 13–18, 2020, Proceedings 18. pp. 349–367. Springer (2020)
40. Schwitter, R., Fuchs, N.E.: Attempto - from specifications in controlled natural language towards executable specifications. CoRR **cmp-lg/9603004** (1996)
41. Seddon, H., Rea, R.: Derbyshire Shared Care Pathology Guidelines – Hypokalaemia in Adults. https://www.uhdb.nhs.uk/shared-care-pathology-guidelines (2012), [Online; accessed 02-05-2023]
42. Seiger, R., Franceschetti, M., Weber, B.: An interactive method for detection of process activity executions from iot data. Future Internet **15**(2), 77 (2023)
43. Silva, T.S., Thom, L.H., Weber, A., Oliveira, J.P.M.d., Fantinato, M.: Empirical analysis of sentence templates and ambiguity issues for business process descriptions. In: OTM Confederated International Conferences "On the Move to Meaningful Internet Systems". pp. 279–297. Springer (2018)
44. Suriadi, S., Andrews, R., ter Hofstede, A.H., Wynn, M.T.: Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs. Information systems **64**, 132–150 (2017)
45. Weske, M.: Business Process Management - Concepts, Languages, Architectures, Third Edition. Springer (2019)
46. Weske, M., Decker, G., Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Model Collection of the Business Process Management Academic Initiative (Apr 2020). https://doi.org/10.5281/zenodo.3758705
47. White, S.A., Miers, D.: BPMN modeling and reference guide: understanding and using BPMN. Future Strategies Inc. (2008)
48. Winter, K., van der Aa, H., Rinderle-Ma, S., Weidlich, M.: Assessing the compliance of business process models with regulatory documents. In: ER 2020, Proceedings. Lecture Notes in Computer Science, vol. 12400, pp. 189–203. Springer (2020)