Using Glossaries to Enhance the Label Quality in Business Process Models

Nicolas Peters and Matthias Weidlich

Hasso Plattner Institute at the University of Potsdam, Germany
nicolas.peters@student.hpi.uni-potsdam.de
matthias.weidlich@hpi.uni-potsdam.de

Abstract: Conceptual models are mostly used for human to human communication. Besides several other aspects, that is, the chosen modelling notation or the model layout, the labelling has a strong influence on the understandability and, therefore, quality of a process model. Consequently, labels should be reused and aligned across different process models, whereas similar labels such as homonyms should be avoided. In order to support these goals, in this paper, we describe an approach that applies a glossary for process modelling. On the one hand, we show how such a glossary that considers structural as well as control flow aspects is generated from an existing collection of process models. The applicability of our glossary generation and the appropriateness of the chosen structural and behavioural aspects is evaluated with the SAP reference model. On the other hand, we introduce two prototypes, the label checker and the label suggester, that illustrate the application of the glossary in the course of modelling.

1 Introduction

Business processes modelled with the Business Process Modeling Notation (BPMN) [OMG09] or as Event-driven Process Chains (EPC) [KNS92] are often used for human to human communication. Therefore, process models have to be easily understandable for the audience in order to benefit by using these models for discussions. Understandability of a process model always depends on its context, i.e., the purpose of the model and the involved stakeholders. For instance, process models designed for software developers as a basis for a system implementation or configuration will differ significantly from those that are used by managers for high-level decision making. Depending on the context, many specific factors affect the understandability of a process model, among them the chosen notation [RD07], the number of different elements used, as well as the model structure [MRC07].

Besides these aspects, the labelling of process model elements heavily influences the understandability [MRR09]. In order to illustrate this causality, Figure 1 depicts three process models of the same structure and notation with different styles of labelling. Without meaningful labels as in model a) one cannot even guess what the process model is about. Slightly better, in model b), one already knows the actions in the process model. Still, the domain the process model relates to is subject for interpretation. In contrast, using

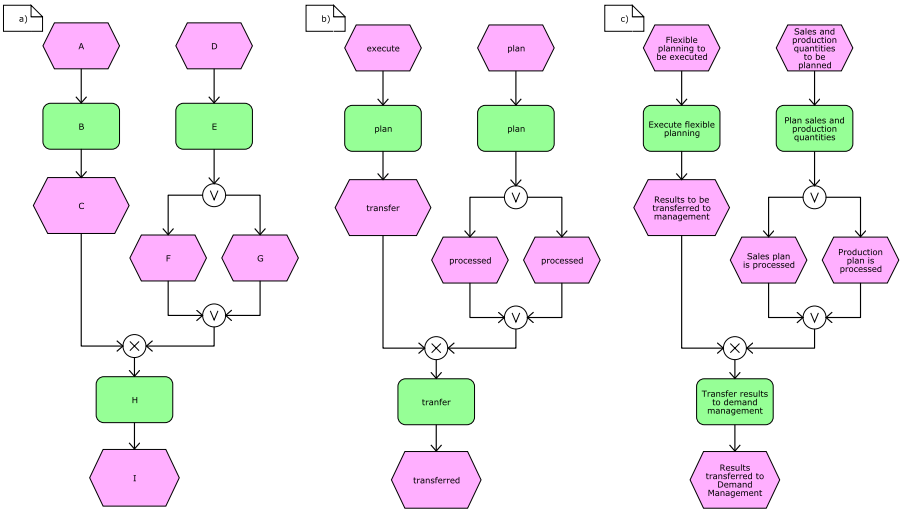

Figure 1: Impact of labelling on the understandability. a) nonsense labelling, b) only verbs, and c) verb-object style labelling

the verb-object style labelling [MRR09] for functions as in model c) allows for a good understanding of the described process. Based thereon, it is easy to infer that the model originates from the sales domain and defines a process of planning sales quantities.

Glossaries have proved useful in team-based environments and, therefore, are an inherent part of many project management methods [Kar88]. Of course, this holds also true for process modelling initiatives [Ros03]. In general, a glossary defines a centralized terminology for a specific domain. Such a glossary usually contains a list of terms and a description for each term. By using a glossary one can ensure that all participants of a collaborative effort have the same understanding of the terms they are using. That, in turn, reduces costs by preventing misunderstandings and shortening discussion times. Furthermore, glossaries are usually controlled by experts and contain terms and descriptions of high quality. This makes glossary entries ideal candidates for the labels of process model elements. Usage of a glossary in the course of modelling results in process models with a high labelling quality. The understandability of these process models increases, because the labels in the different process models are aligned to each other.

In this paper, we describe an approach to use a glossary for labelling process model elements to enhance their labelling quality. In particular, we answer the following two questions: Where does the glossary come from? And how can the glossary be applied to help a modeller creating process models? The first question is tackled by generating a glossary from the labels of an existing process model repository. This approach is motivated by the fact that there exist several reference process models for different domains (cf., [CKL97, Ste01]). These reference models are generic conceptual models that formalise recommended practices [FL03, Fra99, RvdA07]. They are domain-specific and have been

created to streamline existing process models or to improve the understanding of a technical system. Therefore, we assume these models to have a high labelling quality, which, in turn, qualifies them for acting as the basis of a glossary. Furthermore, we do not only consider the labels of the reference model, but also structural and control flow aspects of the given process models. That is, the glossary is enriched with element type information and behavioural characteristics. We show that these aspects are of high value for modelling support by analysing the characteristics of the SAP reference model [CKL97].

With respect to the application of the glossary in the course of modelling, we present two prototypes supporting the modeller. The first application is a label checker that marks all elements with potentially invalid labels and provides feedback on how to resolve the error. With this tool existing models (or whole process repositories) can be checked against the glossary. The second application is a label suggester that proposes labels based on the entered characters, as well as the label's context. The suggester enables a modeller to easily adopt the glossary labels in his models, as well as to find a substitute in the glossary for an invalid label.

Against this background, the remainder of this paper is structured as follows. Section 2 introduces the preliminary concepts used in this paper. Section 3 introduces our approach of generating a glossary for a given collection of process models, while Section 4 reports on findings that stem from the application of our approach to the SAP reference model. Section 5 illustrates the application of the glossary in the course of modelling by presenting two prototypes. Finally, we review related work in Section 6 and conclude in Section 7.

2 Preliminaries

This section provides preliminaries for our work. First, Section 2.1 shortly introduces EPCs as the process modelling language used throughout this paper. Note that, however, our approach for glossary based modelling itself does not rely on specific features of EPCs. Therefore, it can be transferred to other modelling languages directly. Second, Section 2.2 gives details on behavioural profiles as means to capture control flow characteristics of process models.

2.1 Event-driven Process Chains (EPC)

Event-driven process chains (EPCs) [KNS92, NR02] are a popular notation for modelling business processes. They are widely used for human to human communication and have also been applied in the field of reference models. For instance, the SAP reference model consists of hundreds of EPCs. In general, EPC models are a graph comprising functions and events in alternating order. While the former describe elementary actions, the latter specify the process state. Further on, control flow dependencies are expressed using directed flow arcs as well as split and join connectors that are typed as XOR, OR, or AND. A formal definition of EPC syntax can be found in [KNS92]. Note that there are various different

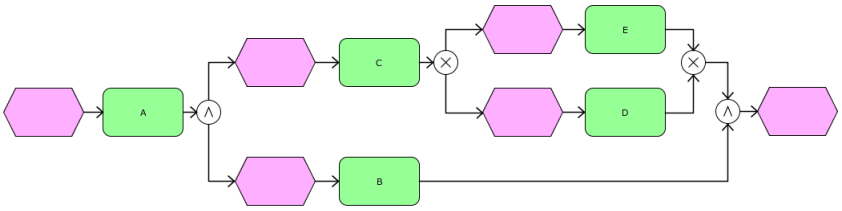

Figure 2: An EPC process model example.

formalisations of execution semantics for EPCs (cf., [KNS92, Men08, Kin04]), as the synchronisation behaviour of the converging OR-connector raises numerous questions (e.g., in cyclic structures). However, the differences of these semantics can be neglected in our context.

2.2 Behavioural Profiles

As mentioned before, our approach assumes a glossary that takes control flow aspects into account. In order to formalise these aspects, we apply the notion of behavioural profiles [WMW09]. These profiles have been introduced as a consistency notion in the field of process model alignment and capture behavioural characteristics of a process model by three different relations, i.e., *strict order*, *exclusiveness*, and *observation concurrency*. All of these relations are defined based on the set of possible traces of a process model.

Strict Order Relation The strict order relation holds between two process elements x and y, if x might happen before y, but not vice versa. In other words, x will be before y in all traces that contain both elements. Moreover, the *reverse strict order relation* holds for any inverted element pair that is in strict order. Note that both relations do not enforce a direct causality. That is, the occurrence of one of the elements in a trace does not enforce the occurrence of the other element.

Exclusiveness Relation Nomen est omen, the exclusiveness relation holds for two process elements, if they never occur together in any process trace.

Observation Concurrency Relation The observation concurrency relation holds for two process elements x and y, if x might happen before y and y might also happen before x. Thus, observation concurrency might be interpreted as the absence of any specific order between two process elements. It is worth to mention that this relation does not imply actual concurrent activation of the process elements. In particular, two process elements that are part of the same control flow cycle are also considered to be observation concurrent.

We illustrate these relations by means of the example EPC model in Figure 2. For instance, functions A and B are in strict order, whereas D and E are exclusive to each other, as there is

no trace of the EPC that contains both functions. Further on, B and C are in the observation concurrency relation, due to their concurrent activation. That is, B might happen before C or vice versa.

Initially, these relations have been defined for free-choice workflow nets [DE95, vdA98] in [WMW09]. There, it was also shown that the four relations (including the reverse strict order relation) partition the Cartesian product of process elements, i.e., every pair of process elements is in one of the four relation. We can easily lift these concepts to the level of EPCs under the assumption of execution semantics that are defined unambiguously. In particular, instantiation semantics for EPCs with multiple start events (cf., [DM09]) and semantics of the converging OR-connector have to be defined properly. Note the latter is an issue solely for complex synchronisation dependencies. For a block-structured joining OR-connector (all incoming arcs originate from a single splitting OR-connector), the behavioural profile would be the same as if the connectors are of type AND. That is, all elements in between would be considered to be observation concurrent to each other.

3 Generation and Setup of a Glossary

Glossaries may contain thousands of entries, which raises the question of how such a glossary is created. Manually adding all terms to a glossary is very time consuming, while it can be done by domain experts only. Thus, if there is existing data in a non-glossary format available for the domain of interest, it saves time and cost to automatically generate the glossary from that data. As mentioned above, we consider reference models consisting of a collection of process models as ideal candidates, as we assume these models to be consistent, precise, and contain labels of high quality.

In this section the structure of the glossary and its features as well as the process of generating a glossary from a collection of process models is described in detail. Section 3.1 discusses the question of which kind of label should appear in the glossary, while Section 3.2 gives details on the technical representation of the glossary. Finally, Section 3.3 and Section 3.4 show how the glossary is enriched with structural and control flow aspects.

3.1 The Choice of Glossary Terms

In general, a glossary is based on a list of terms, which might contain single words or complete phrases. The decision on the appropriate level of granularity for glossary items depends on the primary use case of the glossary. For instance, a glossary might contain names of data objects and a list of actions (verbs) that can be applied on the data objects. Such a glossary would allow to control the labelling of activities in a process model effectively, i.e., an activity label would be a combination of a verb and a data object name. This approach would allow for a glossary that is easier to manage than a glossary that contains all valid combinations of verbs and data object names. However, that would also require the definition of all valid phrase structures. One could imagine to apply association

rules mining [AIS93] in order to derive all valid combinations. Nevertheless, that would require automatic speech tagging [Bri92] in order to identify verbs and objects. Especially for short phrases (like those found in reference models), existing part of speech taggers are not very reliable. Thus, automatic generation is hard to accomplish for this kind of glossary.

In contrast, a glossary might also contain complete phrases that are directly applied as labels for process model elements. Such a glossary is useful, when the set of possible labels is rather small, i.e., the glossary is applied for a distinct domain. In particular, process models that are the basis for process execution might be build from a set of predefine actions.

Due to the obstacle of automatic part of speech tagging, we focus on glossaries that contain full phrases in the remainder of this paper.

3.2 Term Processing and Index Construction

In order to generate a list of labels for our glossary, all process models of the given collection are parsed in order to extract their element labels. However, a pure list of labels is not sufficient. For any reasonable kind of modelling support, the glossary has to offer a querying mechanism in order to search for labels. Similar to information retrieval systems [Kur04] or search engines [CMS09], the glossary has to provide a full text search for labels. Consequently, the basic components of search engines, i.e., text acquisition, text transformation, and index creation, have to be adapted for the use case of a glossary.

First, all labels are tokenized and each token is preprocessed before the search index is created. Such preprocessing aims at increasing the quality of search results by removing leading and trailing white spaces, as well as new line characters. In addition, all label tokens are lower-cased, stop words are filtered, and a stemming algorithm is applied. It is worth to mention that stemming algorithms are language dependant and error-prone. Thus, the stemming result is not always an actual word of that language, e.g., the porter stemmer [Por80] reduces the verb ‘create’ to ‘creat’. However, search queries are preprocessed in the very same way, so that the stemmed terms of a query fit to the stemmed terms in the search index.

After the extracted labels have been preprocessed a search index is created for the glossary. The search index is an inverted index that maps preprocessed terms to the labels in which the terms occur. A very important task of the indexing process is the weighting of terms. Given a search query, these weights are the basis for the calculation of a ranking of result entries. For our glossary, we use the well-known TF-IDF scheme (see [ES07] for further information). The TF-IDF scheme computes for each term in a given label their relevance with regard to a corpus, i.e, the count of term occurrences over all labels. Based on the search index, a query to the glossary is answered using a Vector Space Model (VSM) retrieval model [SWY75].

Based thereon, the glossary can be queried for labels of process elements. Thus, assuming that the process model c) in Figure 1 is part of the model collection from which the glossary is derived, the query ‘flex’ would be answered with the labels ‘Flexible planning to be executed’ and ‘Execute flexible planning’.

Table 1: Derivation of behavioural profile relations for the glossary (SO: strict order, RSO: reverse strict order, EX: exclusiveness, CON: observation concurrency relation)

Relation 1 \ Relation 2	**SO**	**RSO**	**EX**	**CON**
SO	SO	CON	SO	CON
RSO	CON	RSO	RSO	CON
EX	SO	RSO	EX	CON
CON	CON	CON	CON	CON

3.3 Element Types in the Glossary

The glossary that is build as introduced above offers full-text search over all labels. However, it is a common observation that labels for different element types have structural differences in process models. In case of EPCs, functions are often labelled with the verb-object style for describing an action (e.g., 'Execute flexible planning'), whereas events describe the state of the process and, therefore, are often labelled with a passive sentence (e.g., 'Flexible planning to be executed'). This distinction is not reflected in the glossary as described above (the search query 'flex' would return both labels). However, such type information might be used to improve glossary-based modelling support. If a search query also contains information about the element type for which a label is searched in the glossary, this information can be used to narrow the result set. That is, all labels that are not assigned to the element type of interest are removed from the result. Therefore, our glossary stores for each label the types of elements for which the label is used. In order to provide a ranking in case of labels that are used for more than one element type, the number of occurrences of a label in a certain element type is also stored.

Therefore, considering element types ensures that glossary labels are always applied in a *type consistent* manner.

3.4 Behavioural Profiles in the Glossary

Structural information such as element types improve the glossary-based modelling support by reducing the set of retrieved labels for a given query significantly. Similar improvements can be expected when considering the control flow characteristics of the process models from which the glossary is created. Here, the underlying assumption is that labels typically follow some kind of implicit ordering. For instance, 'Receive invoice' will typically occur before 'Archive invoice', whereas 'Handle standard customer' and 'Handle VIP customer' can be expected to never occur both in one process instance. In order to consider these information for modelling support, we also store the relations of the behavioural profile for all pairs of labels in the glossary.

Of course, there might be cases, in which more than one relation is found for a pair of

labels. That, in turn, might be due to fact that a label pair is found in more than one process model. In such a case, the relation to store in the glossary is selected according to Table 1. The idea behind this table is an order of the behavioural relations based on their strictness. We consider the exclusiveness relation as the strongest relation, as it completely disallows two labels to occur in one process trace. In contrast, the observation concurrency relation can be seen as being the weakest relation. It allows two labels to occur in any order in a process trace. Consequently, the strict order and reverse strict order relation are intermediate relations, as they disallow solely a certain order of two labels. Given two labels with different behavioural relations in two process models, the weakest of the two behavioural relations will be stored in the glossary (cf., Table 1). A behavioural relation between two labels is a constraint based on which the result set for a search query is reduced. Therefore, it is reasonable to use solely the weakest of all behavioural relations found for two labels in the respective model collection.

In order to leverage the behavioural relations for labels stored in the glossary, a query against the glossary might specify a so called search context. This context is given by two sets of labels of process model elements that precede or succeed the model element for which the search query is run. Based thereon, the set of results derived based on the full-text search (cf., Section 3.2) and the structural information (cf., Section 3.3) is further reduced. We remove all labels of the result set that fulfil one of the following requirements.

- They are in an exclusiveness relation with one of the labels in the search context.
- They are not in strict order with the succeeding labels in the search context.
- They are not in reverse strict order with the preceding labels in the search context.

As a result, the glossary returns solely these labels for a search query that can be applied for a certain model element without violating the behavioural relations as stored in the glossary for the respective labels. Consequently, the usage of a label from the query result is always *behaviour consistent* with respect to the information stored in the glossary.

4 Case Study: Generating a Glossary from the SAP Reference Model

This section demonstrates the applicability of the glossary generation and setup by presenting an implementation of our approach for the SAP reference model. In addition, we prove the appropriateness of the structural and control flow aspects that are part of our glossary by an experimental setup based on the SAP reference model.

In general, our implementation is based on two components. On the one hand, we use the well-known framework Apache Lucene[1], which provides a full-featured text search engine library written in Java. That is, it offers an implementation of the VSM model, as well as basic term preprocessing as introduced above. While the framework could easily be adapted to take element type information into account, we used the JBPT library[2] for taking the control flow aspects into account. The library provides an implementation of the behavioural profiles for free-choice workflow nets along with an EPC parser and an EPC to

[1] `http://lucene.apache.org/`
[2] `http://code.google.com/p/jbpt/`

Petri net mapping. The mapping follows on standard EPC formalisations (e.g., [vdA99]). As mentioned in Section 2.2, however, a certain preprocessing of EPCs has to take place in order to map them to free-choice workflow nets. In particular, start and end events need to be normalized, such that there is a single start event and a single end event. Further on, block-structured OR-connectors are replaced with AND connectors as that enables a mapping to free-choice Petri nets without affecting the resulting behavioural profile. These normalisation are also provided by the JBPT library.

The SAP reference model [CKL97] that is used to generate the glossary describes the functionality of the SAP R/3 system in its version 4.6. It comprises 604 process diagrams, which are expanded to 737 EPC models as some diagrams contain multiple disconnected EPCs. These EPC models capture different functional aspects of an enterprise, such as sales or accounting. That allows us to assess the amount of reused labels in the reference model and to determine the consistency with respect to structural and control flow aspects. Note that it is well-known that the SAP reference model contains models that are erroneous [MVvD$^+$08]. That is, these models contain deadlocks or livelocks, or even syntactical errors that preclude any reasonable interpretation. Therefore, we exclude these models from the behavioural analysis.

Label reuse. Generation of the glossary based on every second process model of the SAP reference model (that is a set of 368 models) yields a glossary containing 2565 unique labels. If the other half of the reference model is checked against this glossary, 1319 out of 2508 unique labels are also defined in the glossary. That corresponds to a rate of 52.59%. It is obvious that not all labels can be found in the glossary as the test set is an extension to the set of models used for generating the glossary. However, one out of two labels is reused, which indicates how common it seems to reuse labels in reference models.

Element type consistency. For the same glossary and test set, we also analysed the types of elements that have the same label. It is worth to mention that only four labels of the glossary are used for both, functions and events:

- Invoice Verification
- Information System
- Order Settlement
- Shipment Cost Calculation and Settlement

These labels contain no verbs so that an application for both types of process elements is useful in general. Still, 'Information System' does neither describe an activity nor a state and has probably been used accidentally as a label for functions and events, respectively. Besides these four exceptional cases, we see that, despite their enormous quantity, all labels can be identified as being either a function label or an event label. That, in turn, underpins the usefulness to consider such type information in the glossary. As a consequence, it is no surprise that we observed a high consistency value regarding our experimental setup. There is not a single label in the test set that is used for another element type than defined in the glossary, i.e., all labels are type consistent. This result further emphasizes that element types should be considered in a glossary for process modelling.

Behavioural profile consistency. Finally, we evaluated the consistency of behavioural profile relations for labels in the glossary and in the test set. Note that we removed all EPCs

Table 2: Consistency matrix of the behavioural profile relations for our experimental setup.

Rel. in Test Set \ Rel. in Glossary	SO	RSO	EX	CON
SO	yes	no	no	yes
RSO	no	yes	no	yes
EX	yes	yes	yes	yes
CON	no	no	no	yes

that have been identified as erroneous (cf., [MVvD+08]) from the set for the generation of the glossary. As a consequence, behavioural profiles were generated for 268 process models, which led to behavioural relations for 2244 unique pairs of labels. Regarding the test set (again, erroneous EPCs are removed), behavioural profiles are computed for 243 models, yielding behavioural relations for 4732 label pairs (note that these pairs are not unique). Out of these 4732 label pairs, 498 were already defined in the glossary. For these pairs we checked consistency with the behavioural relation stored in the glossary according to Table 2. Following on our discussion of an order of strictness of the behavioural relations (cf., Section 3.4), a relation in the test set is consistent, if the same relation or a weaker relation is defined in the glossary. Again, we observe a high consistency between the relations of the glossary and those of the test set. Only two of the 498 label pairs of the test set showed a behavioural relation that is inconsistent with the glossary. That corresponds to the rate of 99.60%. It is worth to mention that for 494 out of 498 label pairs, the relation in the test set was even equivalent to the relation in the glossary. Thus, our assumption of an implicit ordering between labels seems to hold for the SAP reference model. Therefore, considering control flow aspects between labels based on behavioural profiles is a usefull feature for a process modelling glossary.

5 Application of a Glossary

After we introduced our approach of generating and setting up a glossary, the question of its application during process modelling has to be answered. In this section, we present two ways of applying the glossary to support process modelling aligned to a glossary, which, in the end, increases understandability of the models. Section 5.1 shows how the labelling of a complete process model is analysed based on the glossary, while Section 5.2 shows how label suggestion can be integrated in the process of modelling. For both modelling support features, we also show prototypical implementations based on the Oryx editor[3] [DOW08], a web-based open source framework for process modelling.

[3] `http://www.oryx-project.org`

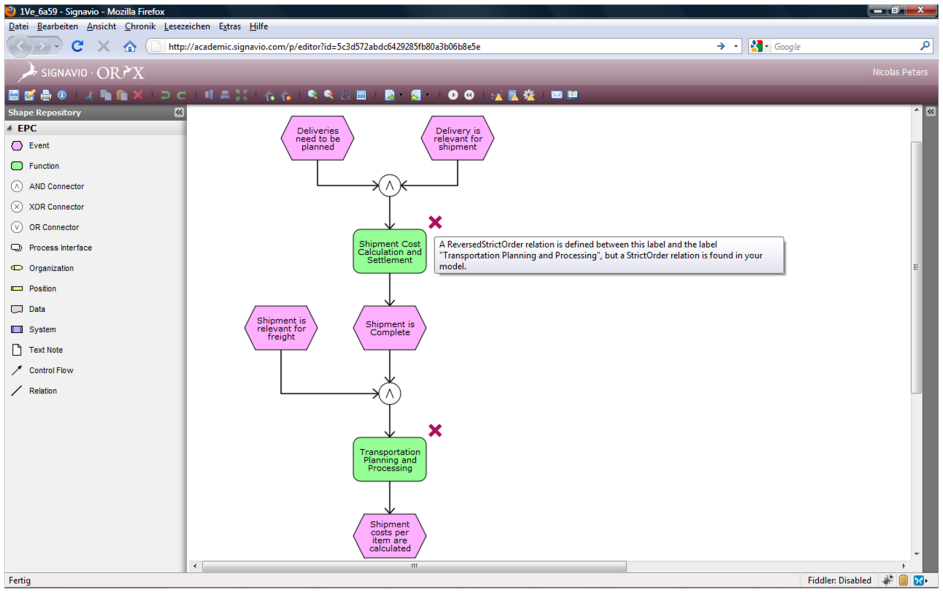

Figure 3: Label analysis reveals an inconsistent behavioural profile relation between two functions.

5.1 Label Analysis

The label analysis feature is a one click application that checks the labelling of a complete model against the glossary. The label analysis follows a two step approach. First, the consistency between the model and the glossary is calculated with respect to the used labels (how many labels are found in the glossary?), the element types (how many of the labels found in the glossary are used for the correct element types?), and behavioural profile relations (how many of the relations between labels in the model are consistent with the respective relations in the glossary?). Such a calculation provides simple consistency metrics that provide the modeller with a first feeling on how well a model is aligned with the glossary.

Figure 4: Label analysis reveals an inconsistent label of an event.

In the second step, the label analysis feature provides detailed feedback on labelling inconsistencies. As illustrated in Figure 3 and 4, process elements with inconsistent labels are highlighted directly in the editor with a red cross. Further on, a mouse over message provides details on the type of labelling inconsistency. That is, either the label cannot be found in the glossary, the label is assigned to a wrong element type, or the label and another label in the process model have an inconsistent behavioural profile relation. Figure 3 depicts an example for inconsistent behavioural profile relations. Here, the glossary defines a strict

order relation between the two function labels marked with red crosses, whereas they are related by reverse strict order in the process model. Figure 4, in turn, depicts the case of an event label that is not defined in the glossary.

5.2 Label Suggestion

In order to use the glossary for labelling elements after their creation, the label suggestion feature can be applied. It uses the information retrieval functionality of the glossary that has been introduced in Section 3. Whenever the modeller starts editing the label of an element, suggested labels from the glossary are immediately presented as depicted in Figure 5. These labels are retrieved by a query that contains the current element's label (or the modeller's input, respectively), the element type, and all preceding and succeeding labels. Each time the modeller enters a new character, the suggested labels are updated. Of course, the modeller might also enter several words, which narrows the result set. An interesting possibility is to enter parts of several words. For instance, using 'proc acqui' or 'acqui proc' as a query, a glossary based on the SAP reference model would return 'Processing of Asset Acquisition' as the first result. Thus, this feature allows to search for possible combinations of verbs and objects.

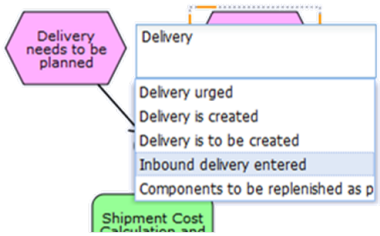

Figure 5: Label suggestion during modelling.

Note that the label suggestion feature might also be applied in order to resolve inconsistent labels that have been detected in the label analysis. In this case, the inconsistent label is queried against the glossary, such that labels similar to the inconsistent one are presented to the modeller. Based thereon, the modeller can replace the label with a similar one that is close to the intended semantics.

6 Related Work

Our approach of using a glossary for process modelling aims at increasing the model quality by providing a centralized terminology. There has been a lot of research on the quality aspects of process models (cf., [HMR08, SRG02, MRvdA09, BGR04]). Although quality of process models is affected by a whole spectrum of different factors, there is no doubt about the importance of the element labelling for the model understandability and, therefore, model quality.

Based on the SAP reference model that we used to generate our glossary, Mendling et al. have investigated common phrase structures [MRR09]. They found out that the verb-object style is the most common phrase structure for EPC functions, a labelling style that is often

referred to as a best practise (e.g., in [MCH03]). They also propose different approaches for a controlled object vocabulary and a controlled verb vocabulary. Such an approach would result in a one word glossary, instead of a complete label glossary as in our approach. As mentioned above these types of glossaries are fundamentally different, as, e.g., the one word glossary raises the question of automatic part of speech tagging. It is worth to mention that not only the functions of the EPCs in the SAP reference model, but also the start events show a set of dedicated phrase structures [DM09]. In particular, the distinction of start events (in the sense of events of the real world) and start conditions (EPC start events that express a condition) is reflected in the label structure. These findings are in line with our observation regarding the clear distinction between event and function labels.

Similar to our approach, Delfmann et al. describe a generic framework for defining a glossary of terms and phrase structures [DHLS08, BDH+09]. Their work is motivated by naming conflicts in process models that are created in distributed teams. Still, the generation of the glossary is regarded as a manual task, which might require serious efforts. Our approach is more lightweight in the sense that only complete labels instead of grammars are considered in order to benefit from automatic glossary generation. In addition, our approach considers structural as well as control flow aspects of process models to ensure a high degree of labelling consistency and to increase the usefulness of term suggestions.

Other work aims at providing modelling support based on a repository of model patterns that are extracted based on the element labelling. In [TRI09] the authors propose a set of generic activity patterns that might be used as basic building blocks of process models. While this approach requires a priori knowledge about the patterns, similar patterns might also be extracted from a model repository using association rules mining techniques [SWMW09]. Here, the patterns are lifted from the level of activities to the more abstract level of actions in order to enable reuse in a broader context. Again, this relies on part of speech tagging.

Support for process modelling might also be based on search techniques [HKL08]. Here, the main idea is to search a process repository for similar models in order to suggest extension of the current model. Of course, such a similarity search considers control flow and structural aspects of a process model, which resembles our idea of taking such information into account when querying a glossary. Similar approaches for modelling support might be based on ontology knowledge, e.g, [KO05]. Obviously, such approaches require the existence of a domain specific ontology. However, automatic generation of such an ontology imposes various challenges that go beyond the aforementioned part of speech tagging issue.

7 Conclusion

In this paper, we presented an approach that aims at increasing the labelling quality of business process models based on a glossary. We argued that the existence of reference models enables us to generate such a glossary automatically for a dedicated domain. Further on, we advocated the enrichment of a glossary with structural and behavioural information in terms of control flow aspects for the labels. The applicability of our approach as well

as the appropriateness of our choice on such structural and behavioural information was demonstrated using the SAP reference model. Further on, we illustrated the application of such a glossary in the course of process modelling by two modelling support features, which we implemented based on an open source modelling framework.

We showed that effective modelling support can be achieved once reference models are available. In particular, our approach of narrowing the set of potential labels for a certain element based on structural and behavioural information proved to be valuable. Even though our experiments provided evidence for the usefulness of the approach, future research has to evaluate the presented prototypes empirically in a user study.

We mentioned before that our approach is independent of the EPC notation and might be applied for other modelling languages as well. Still, languages with a huge set of element types (e.g., BPMN) might require further investigation. Probably, not all types imply differences in the labelling structure, so that clustering of element types has to be explored.

Further on, we based our approach on the assumption of high-quality labels in the collection of models from which the glossary is generated. Therefore, consistency checks for such a model collection (cf., [KMS08]) and quality metrics for the glossary itself have to be defined and evaluated. For instance, the number of homonyms used in a glossary can be regarded as such a metric, as usage of homonyms causes misunderstandings. Such homonymy might be detected using existing tools such as WordNet[4].

References

[AIS93] R. Agrawal, T. Imielinski, and A. N. Swami. Mining Association Rules between Sets of Items in Large Databases. In *COMAD*, pages 207–216, Washington, D.C., 1993.

[BDH+09] J. Becker, P. Delfmann, S. Herwig, L. Lis, and A. Stein. Towards Increased Comparability of Conceptual Models - Enforcing Naming Conventions through Domain Thesauri and Linguistic Grammars. In *ECIS*, June 2009.

[BGR04] Wasana Bandara, Guy G. Gable, and Michael Roseman. Factors and measures of business process modelling: model building through a multiple case study. *European Journal of Information Systems*, 14(4):347 – 360, 2004.

[Bri92] Eric Brill. A Simple Rule-Based Part of Speech Tagger. In *ANLP*, pages 152–155, 1992.

[CKL97] Thomas A. Curran, Gerhard Keller, and Andrew Ladd. *SAP R/3 Business Blueprint: Understanding the Business Process Reference Model*. Prentice-Hall, 1997.

[CMS09] Croft, Metzler, and Strohman. *Search Engines: Information Retrieval in Practice*. Pearson Education, 2009.

[DE95] Jörg Desel and Javier Esparza. *Free choice Petri nets*. Cambridge University Press, New York, NY, USA, 1995.

[4] `http://wordnet.princeton.edu/`

[DHLS08] Patrick Delfmann, Sebastian Herwig, Lukasz Lis, and Armin Stein. Eine Methode zur formalen Spezifikation und Umsetzung von Bezeichnungskonventionen für fachkonzeptionelle Informationsmodelle. In *MobIS*, volume 141 of *LNI*, pages 23–38. GI, 2008.

[DM09] Gero Decker and Jan Mendling. Process Instantiation. *Data & Knowledge Engineering (DKE)*, 68:777–792, 2009.

[DOW08] Gero Decker, Hagen Overdick, and Mathias Weske. Oryx - An Open Modeling Platform for the BPM Community. In *BPM*, volume 5240 of *LNCS*, pages 382–385. Springer, 2008.

[ES07] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer-Verlag, 2007.

[FL03] P. Fettke and P. Loos. Classification of reference models - a methodology and its application. *Information Systems and e-Business Management*, 1(1):35–53, 2003.

[Fra99] U. Frank. Conceptual Modelling as the Core of the Information Systems Discipline - Perspectives and Epistemological Challenges. In *Proceedings of the America Conference on Information Systems (AMCIS)*, pages 695–698, 1999.

[HKL08] Thomas Hornung, Agnes Koschmider, and Georg Lausen. Recommendation Based Process Modeling Support: Method and User Experience. In *ER*, volume 5231 of *LNCS*, pages 265–278. Springer, 2008.

[HMR08] Mitra Heravizadeh, Jan Mendling, and Michael Rosemann. Dimensions of Business Processes Quality (QoBP). In *Business Process Management Workshops*, volume 17 of *LNBIP*, pages 80–91. Springer, 2008.

[Kar88] J. Karimi. Strategic Planning for Information Systems: Requirements and Information Engineering Methods. *Journal of Management Information Systems*, 4:5–24, 1988.

[Kin04] Ekkart Kindler. On the semantics of EPCs: A framework for resolving the vicious circle. In J. Desel, B. Pernici, and M. Weske, editors, *Business Process Management (BPM)*, volume 3080 of *Springer, LNCS*, pages 82–97, Potsdam, Germany, June 2004.

[KMS08] E. Knauss, S. Meyer, and K. Schneider. Recommending Terms for Glossaries: A Computer-Based Approach. In *First International Workshop on Managing Requirements Knowledge*, pages 25–31, 2008.

[KNS92] G. Keller, M. Nüttgens, and A.-W. Scheer. Semantische Prozeßmodellierung auf der Grundlage 'Ereignisgesteuerter Prozeßketten (EPK)'. Veröffentlichungen des instituts für wirtschaftsinformatik (iwi), Universität des Saarlandes, January 1992.

[KO05] Agnes Koschmider and Andreas Oberweis. Ontology Based Business Process Description. In *EMOI-INTEROP*, volume 160 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2005.

[Kur04] D. Kurupka. *Modelle zur Repraesentation natürlichsprachlicher Dokumente. Ontologie-basiertes Information-Filtering und -Retrieval mit relationalen Datenbanken*. Logos Verlag, 2004.

[MCH03] Th. W. Malone, K. Crowston, and G. A. Herman. *Organizing Business Knowledge: The MIT Process Handbook*. The MIT Press, Cambridge, MA, USA, 1st edition, September 2003.

[Men08] Jan Mendling. *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness*, volume 6 of *Lecture Notes in Business Information Processing*. Springer, 2008.

[MRC07] Jan Mendling, Hajo A. Reijers, and Jorge Cardoso. What Makes Process Models Understandable? In *BPM*, volume 4714 of *LNCS*, pages 48–63. Springer, 2007.

[MRR09] J. Mendling, H. Reijers, and J. Recker. Activity labeling in process modeling: empirical insights and recommendations. *Information Systems (IS)*, 2009. to appear.

[MRvdA09] J. Mendling, H.A. Reijers, and W.M.P. van der Aalst. Seven Process Modeling Guidelines (7PMG). *Information and Software Technology (IST)*, 2009. to appear.

[MVvD+08] Jan Mendling, H. M. W. Verbeek, Boudewijn F. van Dongen, Wil M. P. van der Aalst, and Gustaf Neumann. Detection and prediction of errors in EPCs of the SAP reference model. *Data Knowl. Eng.*, 64(1):312–329, 2008.

[NR02] Markus Nüttgens and Frank J. Rump. Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In *Promise*, volume 21 of *LNI*, pages 64–77. GI, 2002.

[OMG09] OMG. *Business Process Modeling Notation (BPMN) 1.2*, January 2009.

[Por80] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[RD07] J. Recker and A. Dreiling. Does it matter which process modelling language we teach or use? an experimental study on understanding process modelling languages without formal education. In *18th Australasian Conference on Information Systems*, pages 356–366, 2007.

[Ros03] M. Rosemann. *Process Management: A Guide for the Design of Business Processes*, chapter Preparation of process modeling, pages 41–78. Springer, 2003.

[RvdA07] Michael Rosemann and Wil M. P. van der Aalst. A configurable reference modelling language. *Inf. Syst.*, 32(1):1–23, 2007.

[SRG02] Wasana Sedera, Michael Rosemann, and Guy G. Gable. Measuring Process Modelling Success. In *ECIS*, 2002.

[Ste01] S. Stephens. The Supply Chain Council and the Supply Chain Operations Reference Model. *Supply Chain Management*, 1:9–13, 2001.

[SWMW09] S. Smirnov, M. Weidlich, J. Mendling, and M. Weske. Action Patterns in Business Process Models. In *7th International Joint Conference on Service Oriented Computing (ICSOC)*, Stockholm, Sweden, 2009.

[SWY75] Gerard Salton, A. Wong, and C. S. Yang. A Vector Space Model for Automatic Indexing. *Commun. ACM*, 18(11):613–620, 1975.

[TRI09] Lucineia Thom, Manfred Reichert, and Cirano Iochpe. Activity Patterns in Process-aware Information Systems: Basic Concepts and Empirical Evidence. *International Journal of Business Process Integration and Management (IJBPIM)*, 2009. to appear.

[vdA98] Wil M. P. van der Aalst. The Application of Petri Nets to Workflow Management. *Journal of Circuits, Systems, and Computers*, 8(1):21–66, 1998.

[vdA99] Wil M. P. van der Aalst. Formalization and verification of event-driven process chains. *Information & Software Technology*, 41(10):639–650, 1999.

[WMW09] Matthias Weidlich, Jan Mendling, and Mathias Weske. Computation of Behavioural Profiles of Processes Models. Technical report, BPT Technical Report 07, 2009.