Nicolas Peters, Matthias Weidlich

Automatic Generation of Glossaries for Process Modelling Support

Process models are often used for human to human communication. Besides other aspects, e.g., the chosen modelling notation or the model layout, the labelling has a strong influence on the understandability and, therefore, quality of a model. Consequently, labels should be reused and aligned across different process models. In order to support these goals, a glossary might be applied in the course of modelling. In this article, we argue that such a glossary can be generated automatically from the labels of an existing process model collection, e.g., a reference model. We introduce an approach for such a glossary generation that takes additional information on structural as well as control flow aspects into account. The applicability of our approach is illustrated by means of two case studies. Based thereon, we also report on findings regarding the appropriateness of the chosen structural and behavioural aspects.

1 Introduction

Conceptual models in general, and business process models in particular are often used for human to human communication. Thus, *understandability* is a major quality criterion for such models. Still, understandability of a model always depends on its context, i.e., its purpose and the involved stakeholders. Besides several other aspects, e.g., the chosen modelling notation (Recker and Dreiling 2007) or the model structure (Mendling et al. 2007), the labelling has a strong influence on the understandability and, therefore, quality of a process model (Mendling et al. 2010a).

The understandability of the labelling of a single process model might be investigated in isolation (Friedrich 2009; Mendling et al. 2010a). However, the labelling can also be assessed with respect to a certain corpus of process models, i.e., an existing process model collection. In this case, we aim at a consistent usage of labels throughout a process model collection. In order to achieve this goal, process modelling initiatives might be guided by glossaries that provide a centralised terminology for a specific domain (Rosemann 2003). 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 modelling 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.

In particular, we see two use cases for the application of glossaries in process modelling initiatives. First, the labelling of a dedicated process model can be checked against the glossary. Thus, we can identify the labels that are not contained in the glossary, or for which there are inconsistencies in the usage as imposed by the glossary. An example for the latter would be the usage of a label for an element type, which is not allowed for by the glossary. Based thereon, we are also able to quantify any potential deviation, such that a process analyst is provided with a first feeling on how well a model is aligned with the glossary. Second, a glossary can be integrated directly in the course of modelling. Labels from the glossary might be suggested, whenever a process analyst starts editing a label of a process model. That, in

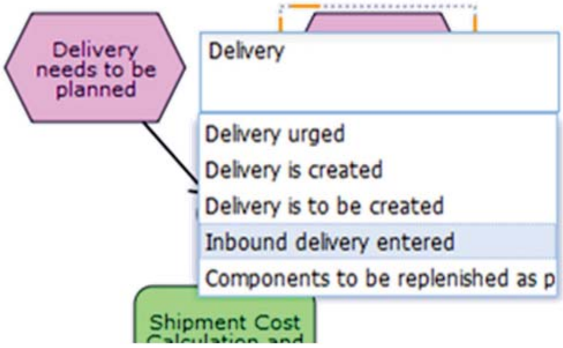

Figure 1: Glossary based label suggestion

turn, enables process analysts to easily adopt the glossary labels in their models. This use case is illustrated in Figure 1, which depicts the integration of label suggesting features into the Oryx editor[1] (Decker et al. 2008).

While there is no doubt about the benefits of applying a glossary in process modelling, the question of how to come up with a glossary has to be addressed. In this article, we argue that a glossary might be generated automatically from the labels of an existing process model collection. This approach is motivated by the fact that there exist several reference process models for different domains, cf. Curran et al. (1997); Stephens (2001). Reference models are generic conceptual models that formalise recommended practices (Fettke and Loos 2003; Frank 1999; Rosemann and Aalst 2007). They are domain-specific and have been created to streamline existing process models or to improve the understanding of a technical system. Thus, we assume these models to have a high labelling quality, which, in turn, qualifies them for acting as the basis of a glossary.

This article is an extended and revised version of our earlier work (Peters and Weidlich 2009), in which we introduced a first approach to generate a glossary, evaluated it based on the SAP reference model (Curran et al. 1997), and discussed its application in detail. In particular, our approach considers structural and control flow aspects of the given process models besides the pure element labelling. In this article, we extend the process of generating a glossary by taking dependencies in terms of co-occurrence of labels into account. Thus, the glossary is enriched with further information, while still following a fully automatic approach. In addition, we present a new case study for our approach using a model collection obtained from an insurance company, which is currently used as a basis for a modelling initiative.

Against this background, the remainder of this article is structured as follows. Section 2 introduces the preliminaries for our work. Section 3 introduces our approach of generating a glossary for a given collection of process models. Section 4 reports on findings that stem from the application of our approach in two case studies. We discuss our results and elaborate on limitations of our approach in Sect. 5. Finally, we review related work in Sect. 6 and conclude in Sect. 7.

2 Preliminaries

This section provides preliminaries for our work. First, Sect. 2.1 shortly introduces EPCs as the process modelling language used throughout this article. However, our approach itself does not rely on specific features of EPCs and can, therefore, be transferred to any other modelling language. Second, Sect. 2.2 discusses behavioural profiles as means to capture control flow characteristics of process models.

2.1 Event-Driven Process Chains (EPC)

Event-Driven process chains (Keller et al. 1992; Nüttgens and Rump 2002) 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. 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

[1] http://www.oryx-project.org

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 Keller et al. (1992). Note that there are various different formalisations of execution semantics for EPCs (cf. Keller et al. 1992; Kindler 2004; Mendling 2008), 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 generation of a glossary takes control flow aspects into account. In order to formalise these aspects, we apply the notion of behavioural profiles (Weidlich et al. 2010). 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 *interleaving order*. All of these relations are defined based on the set of possible traces of a process model.

Strict Order. 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. The exclusiveness relation holds for two process elements, if they never occur together in any process trace.

Interleaving Order. The interleaving order relation (also referred to as observation concurrency) holds for two process elements x and y, if x might happen before y and y might also happen before x. Thus, interleaving order 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 in interleaving order.

We illustrate these relations by means of the example EPC model in Fig. 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 interleaving order, 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 (see Aalst 1998) in Weidlich et al. (2010). 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 relations. 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. Decker and Mendling (2009)) 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 in interleaving order 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 time consuming and 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.

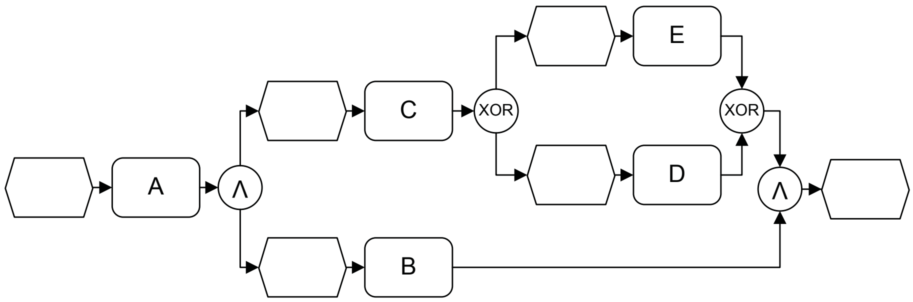

Figure 2: An EPC process model example

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 as well as the process of its generation from a collection of process models is described in detail. Section 3.1 discusses the question which kind of label should appear in the glossary and how labels are pre-processed. Subsequently, Sect. 3.2 and Sect. 3.3 show how the glossary is enriched with structural aspects, such as type information and co-occurrence dependencies. We also consider control flow aspects in the glossary in Sect. 3.4. Finally, Sect. 3.5 summarises the actual process of generating a glossary.

3.1 Terms of a Glossary

In general, a term of a glossary might be a single word or a complete phrase. 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. Obviously, such a glossary is easier to manage than a glossary that contains all valid combinations of verbs and data object names. However, this also requires the definition of all valid phrase structures. In order to extract such phrase structures automatically, automatic speech tagging (Brill 1992) has to be applied in order to identify verbs and objects. Especially for short phrases as they are used as labels for process model elements, existing part of speech taggers are not very reliable, cf. Leopold et al. (2009). While Leopold et al. (2009) achieve good tagging results based on Word Net[2] for certain phrase structures, a more generic solution for different phrase structures still has to be presented. Therefore, automatic generation is hard to accomplish for a glossary that separates actions and objects.

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, creation of process models that are (at least partly) built from a set of predefined actions can be guided appropriately. Due to the obstacle of automatic part of speech tagging, we focus on glossaries that contain full phrases in the remainder of this article.

3.2 Element Types in the Glossary

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

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

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 should be reflected in the glossary to improve glossary-based modelling support. In order to suggest a label for a process model element, a search query checked against the glossary contains information on the element type for which a label is searched. Based thereon, the result set of labels from the glossary that is derived via full-text search is further narrowed. That is, all labels that are not assigned to the element type of interest are removed from the result. Therefore, we store the types of elements for which a 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.

3.3 Label Co-Occurrences in the Glossary

Another structural aspect that is considered in the glossary is the co-occurrence of labels throughout the process models in a model collection. The idea behind is that process models that show a certain overlap in the described functionality are likely to have a set of labels in common. Thus, analysis of the co-occurrences of labels reveals clusters of labels that are semantically related. For instance, it might be observed that there are multiple process models containing the labels 'Receive invoice', 'Trigger payment', and 'Archive invoice', i.e., these labels are co-occurring. Once this information is stored in the glossary, it can be leveraged for modelling support. Clearly, the type of support depends on the strictness of the relation between such labels. On the one hand, labels missing in a process model might be detected during a consistency check, if there is a very strong causal coupling. On the other hand, a rather loose causal relation can still be used to filter label suggestions in the course of modelling.

The idea of discovering such co-occurrence patterns between entities of a large collection stems from the domain of data mining. That is, association rules mining (Agrawal et al. 1993) aims at the discovery of relations between variables of a database. Note that these techniques have been adapted for the domain of process models in Smirnov et al. (2009), which introduces the notion of co-occurrence and behavioural action patterns. Those patterns are not defined on the level of labels, but on the more abstract level of actions, which enables reuse in a broader context. However, this approach assumes a technique to derive the action from the label of a process model element. Thus, it relies on part of speech tagging, which seems to be inappropriate in our context according to the state of the art, cf. Sect. 3.1. Therefore, we focus on co-occurrences between labels of process model elements in the glossary.

Following on the approach introduced in Smirnov et al. (2009), we first extract all groups of labels that are often co-occurring. Here, the *support* for a dedicated group of labels is the number of occurrences of these label altogether in models in the collection. In order to measure the strength of the co-occurrence of a group of labels, we compute the *confidence* for a cluster of labels. That is, the fraction of models supporting the group (models that contain all labels) and those that contain at least one of the labels is calculated.

As mentioned above, the information on co-occurrence is used either in a rather strict consistency check, or for filtering or ranking label suggestions. Clearly, the choice of how to consider details on co-occurrences defined in the glossary is guided by the support and confidence values. Here, it seems reasonable to require a minimal support level in order to take label clusters into account, whereas solely clusters with a very high confidence are applicable for checking labelling consistency. In order to leverage the co-occurrence dependencies for label suggestions, 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 is derived based on the full-text search on labels, while the structural information (cf. Sect. 3.2) further reduces the set. We remove any label of the result set for which there is no label cluster that is built from this label and labels of the search context, while the confidence for the cluster is above a certain threshold. For obvious reasons, this filtering is only applicable if the search context contains a reasonable number of labels.

3.4 Behavioural Profiles in the Glossary

Structural information such as element types and co-occurrence dependencies 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. Although it is possible to consider not only pairs, but also n-tuples of co-occurring labels (cf. Sect. 3.3 and the behavioural action patterns in Smirnov et al. 2009), we focus on the control flow aspects for pairs of labels. That is motivated by our use case of modelling support that does not aim at the identification of a few very prominent patterns (with high support and confidence), but focusses on the whole corpus of pairs of labels. Even label pairs with low support and confidence values can be considered in ranking label suggestions, although they do not qualify for being an action pattern.

Table 1: Derivation of behavioural profile relations for the glossary (so: strict order, rso: reverse strict order, ex: exclusiveness, io: interleaving order)

		Relation 2			
		so	**rso**	**ex**	**io**
Relation 1	**so**	so	io	so	io
	rso	io	rso	rso	io
	ex	so	rso	ex	io
	io	io	io	io	io

Of course, there might be cases, in which more than one relation is found for a pair of labels. In such a case, the relation to store in the glossary is selected according to Tab. 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 interleaving order 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. Tab. 1). A behavioural relation between two labels is a constraint based on which violations are detected or 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.

When checking the consistency of the labelling of a process model, the behavioural relations are compared along the aforementioned hierarchy of behavioural relations. The behavioural relation must be equal or stricter than the one defined in the glossary for both labels. When using the behavioural relations to narrow the set of label suggestions for a given search query, again, the existence of a search context is assumed. Thus, the search query contains two sets of labels of process model elements that precede or succeed the model element for which a label is searched. Given the set of results derived based on full-text search on labels and filtered according to Sect. 3.2 and Sect. 3.3, labels that fulfill one of the following requirements are removed.

- 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.

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.

3.5 The Process of Glossary Generation

In the previous sections, we discussed several aspects of process model elements that are considered in our glossary. Based thereon, this section summarises the process of glossary generation. We do not aim at a complete formalisation of the glossary generation as most of the major steps have been formalised in related work. Still, we try to disambiguate the essence of the glossary generation with some formal syntax.

For this purpose, we assume a process model to be defined as a tuple $P = (N, F, t, l)$ with N being a finite non-empty set of nodes and $F \subseteq N \times N$ a flow relation, such that (N, F) is a connected graph. Further on, we postulate Ω as being the set of all node types and Γ as the set of all node labels. Then, $t : N \mapsto \Omega$ is a function assigning a type to each node, while $t : N \mapsto \Gamma$ assigns labels to nodes. We lift the labelling function to the level of a process model, i.e., $l(P) = \bigcup_{n \in N}\{l(n)\}$. Although this model abstracts from various aspects of common process modelling languages, such as different connector or gateway types and their semantics, it suffices to illustrate the process of glossary generation. Still, we assume an unambiguous definition of execution semantics to be able to compute behavioural profiles (cf. Sect. 2.2).

With $C = \{P_1, \dots, P_n\}$ as a collection of process models, the generation of a glossary $\mathcal{G}$ comprises the following steps.

1. First and foremost, the glossary $\mathcal{G}$ contains all labels that are used for elements in all models. This yields a glossary that is defined as $\mathcal{G} = (L)$ with $L = \bigcup_{i \in \{1,\dots,n\}} l(P_i)$ being a set of labels.
2. The glossary is enriched with type information. That is, the glossary $\mathcal{G} = (L, g)$ comprises a function $g : L \mapsto \Omega^*$ that maps a label to a sequence of element types for which the label might be applied. Here, the sequence also encodes the ranking of element types in case a label can be applied to multiple element types. Obviously, this function is defined based on the types for which a certain label is used in the original process models. Thus, $g(s) = \omega_0, \dots, \omega_m$ for a label s, if and only if, for every ω_i there is a model $P_j = (N_j, F_j, t_j, l_j)$ that is part of C and for some $n \in N_j$ it holds $l_j(n) = s$ and $t_j(n) = \omega_i$.
3. Further on, the glossary is enriched with information on co-occurrences of labels. Then, the glossary $\mathcal{G} = (L, g, LC, supp, conf)$ consists of a set of label clusters $LC \subseteq \wp(L)$ that represent sets of co-occurring labels along with information on their support and confidence value (cf. Sect. 3.3). The latter is defined as two functions, $supp : LC \mapsto \mathbb{N}$ and $conf : LC \mapsto [0, 1]$. Computation of the support and confidence value follows the formalisation that has been presented in Smirnov et al. (2009) and the algorithms presented in Agrawal et al. (1993) and Agrawal and Srikant (1994).

4. In the last step, the glossary is further enriched with information on behavioural relations between labels. With $\mathcal{R}$ being one of the relations of the behavioural profile (cf. Sect. 2.2), we map $\mathcal{R}$ to one of the symbols $R = \{so, rso, ex, io\}$ that represent the respective relation. In other words, the mathematical relation is resolved to a symbol that indicates the type of the relation. Based thereon, each pair of labels (s_1, s_2) can be associated to one of these relation symbols in $R = \{so, rso, ex, io\}$. If there is solely one pair of nodes (n_1, n_2) in a process model $P = (N, F, t, l)$ with $n_1, n_2 \in N$, $l(n_1) = s_1$, and $l(n_2) = s_2$, the symbol for the relation between the labels is deduced from the relation of the behavioural profile between the nodes directly. In case different nodes carry the same label, but show different behavioural relations, a common relation symbol is derived according to Tab. 1. The same approach is taken for pairs of labels that show different behavioural relations in different process models. Hence, the glossary associates each pair of labels to a single symbol representing a relation of the behavioural profile. The glossary is defined as $\mathcal{G} = (L, g, LC, supp, conf, r)$ with $r : (L \times L) \mapsto R$ associating pairs of labels with (symbols referring to) a behavioural relation.

These steps lead to the generation of a glossary that is defined as $\mathcal{G} = (L, g, LC, supp, conf, r)$.

4 Case Studies: Generating a Glossary

This section elaborates on two case studies in order to demonstrate the applicability of our approach for the automatic generation of a glossary. Further on, we report on findings concerning the appropriateness of the structural and control flow aspects that are part of our glossary by an experimental setup. That is, we generate a glossary based on half of the models in each collection and analyse the relation of the other half of the collection against this glossary.

First, Sect. 4.1 discusses the case of the SAP reference model. Second, Sect. 4.2 turns the focus towards a model collection that we obtained from a German health insurer.

4.1 A Glossary based on the SAP Reference Model

The SAP reference model (Curran et al. 1997) 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 (Mendling et al. 2008). 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, i.e., 'Invoice Verification', 'Information System', 'Order Settlement', and 'Shipment Cost Calculation and Settlement'. These labels contain no verbs so that

Table 2: Extract of the analysis of label co-occurrences: number of label triples in the SAP reference model

		Support			
		2	**5**	**10**	**20**
Confidence	**0.2**	114448	7775	464	11
	0.4	7430	3825	294	10
	0.6	98	91	86	7
	0.8	13	6	5	5
	1.0	2	2	1	1

an application for both types of process elements is useful in general. Still, 'Information System' neither describes 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 emphasises that element types should be considered in a glossary for process modelling.

Label co-occurrence analysis. For the labels that are contained in the generated glossary, we also analysed their co-occurrence dependencies. In particular, we identified pairs, triples, and quadruples of labels that are co-occurring. For the discussion of our findings, we focus on the case of label triples, i.e., clusters that are build of three different labels. Table 2 shows an extract of the results by providing the number of triples in relation to a given support and confidence value, respectively. We see that there are 114448 distinct label triples that are co-occurring in at least two process models, if the confidence value is required to be at least 0.2 for the cluster. Our extract illustrates that there are only a few label triples with high confidence values above 0.6. In fact, a confidence value of one can be observed only for two triples. A confidence value of 0.2 has to be interpreted as follows. In 20% of the cases, the occurrence of one of the labels of the cluster implies the occurrence of the remaining labels in the same model. For a confidence of one, in turn, we know that a process model contains either all or none of the labels of a cluster.

As mentioned in Sect. 3.3, we might use the information on co-occurrences for checking the consistency of the labelling of a process model. Apparently, solely label clusters with a high confidence value near to one should be considered in this step. For those rules, a violation hints at a modelling error directly. According to Tab. 2, however, there are only a few of such clusters in the SAP reference model.

Regarding the application of the glossary for label suggestion, it is important to notice that even clusters with a rather low confidence value are worth to be considered. Of course, a label cluster with a confidence value below 0.5 cannot be regarded as a reusable pattern. That is due to the fact that such a value hints at a high probably for some of the labels to occur in a model that does not contain the other labels of the cluster. However, in our context that aims at suggesting labels, it is reasonable to consider also labels that are co-occurring only in some models with labels of the search context (cf. Sect. 3.3). Obviously, labels of the search result that are co-occurring with some of the labels of the search context are more likely to be chosen than labels that do not show any co-occurrence dependency with labels of the search context, despite a potentially low confidence value. Still, the results shown for label triples in Tab. 2 suggest to define a threshold w.r.t. support of a certain co-occurrence relation, as a support of two yields a very large number of co-occurrence dependencies. That, in turn, might have a negative impact on the ability to filter label suggestions. Owing to the enormous amount of co-occurrence dependencies, virtually all labels of the search result can be assumed to be part of a co-occurrence cluster with some label of the search context. In addition, there are still numerous label triples that have a support of 5 or 10, respectively, such that cluster with low support values can be neglected. With a support of 20, however, nearly no label triples are identified.

With these high support values, it is no surprise that a comparison of the test set against the glossary in terms of label clusters reveals a big overlap. For instance, for the case of label triples, 48.06% of all label clusters found in the test set are already defined in the glossary. This result, along with the huge amount of label clusters with high support values suggests to consider the information on co-occurrences of labels in the glossary. That, in turn, allows for narrowing the result set when deriving label suggestions from the glossary. In addition, a few label clusters with high confidence values can be used for an assessment of labelling consistency.

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 that have been identified as erroneous (cf. Mendling et al. 2008)) or ambiguous (cf. Decker and Mendling 2009) 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 (that are not unique). Out of these 4732 label pairs, 498 were already defined in the glossary, such that their consistency with the glossary could be determined. Following on our discussion on an order of strictness of the behavioural relations (cf. Sect. 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 useful feature for a process modelling glossary.

4.2 A Glossary for a Health Insurance Company

The model collection for this case study has been provided by a German health insurer. The collection comprises 1029 process diagrams that are expanded to 1350 EPC models. They describe the business functions from an organisational point of view and have been applied for staff planning. Clearly, this model collection cannot be seen as a reference model in the general sense. However, the models have been created by a rather small group of experts, such that the used terminology can be assumed to be consistent. Currently, the insurance company is facing a follow-up process modelling initiative. Therefore, the question of how to leverage the existing model collection to guide these efforts is of particular importance. To this end, the generation of a glossary following on the approach introduced in this article might be applied to support the creation of process models.

In the remainder of this section, we report on the results of repeating the analysis of a generated glossary as introduced in the previous section. That, in turn, allows us to investigate to which extent the observation obtained for the SAP reference model can be transferred to a company specific model collection.

Label reuse. First and foremost, we generated a glossary based on every second process model of the collection. Considering 675 EPC models, we generated a glossary comprising 6688 unique labels. Again, the other half of the model collection was applied as a test set and checked against the

Table 3: Extract of the analysis of label co-occurrences: number of label triples in the model collection

		Support			
		2	**10**	**50**	**100**
Confidence	**0.2**	1852309	72618	2845	482
	0.4	784639	60550	2396	443
	0.6	172454	51600	2070	393
	0.8	99961	37397	1880	344
	1.0	50039	26238	1771	307

glossary. For 1310 out of 6089 unique labels of the test set, there is an entry in the glossary, which corresponds to a rate of 21.51%. We see that the reuse of labels is less common in this model collection compared to the SAP reference model. Obviously, the functional overlap between the process models is smaller and there is less redundancy in the model collection. Nevertheless, the reuse of every fifth label still indicates a huge potential for modelling support, if we assume this ratio to persist for process models created in the current modelling initiative.

Element type consistency. As for the previous case study, we also analysed the relation between labels and element types. Our results confirm the observation made for the SAP reference model. That is, 99.92% of the labels in the test set are used solely for the element types as stored in the glossary. This further underpins the need to consider element types in a glossary.

Label co-occurrence analysis. In order to analyse the dependencies in terms of co-occurrence of labels in the generated glossary we, again, identified pairs and triples of co-occurring labels. Focussing on the triples, Tab. 3 provides an extract of our results similar to one presented for the SAP reference model in Tab. 2. Note that the scale for the support is different though. In general, we see that there is a very high number of label triples. For a support value of two and a confidence value of 0.2, there are nearly two million label clusters. Moreover, we see that there is a huge number of label clusters that shows a confidence value of one. For these clusters, we know that a process model contains either none or all of the labels of the cluster. As illustrated in Tab. 2, there is a high number of these clusters even for high support values.

As discussed in Sect. 3.3, these clusters with a confidence value of one can be leveraged for checking the consistency of the labelling of a process model. Thus, our results reveal a huge potential for checking the consistency of the labelling of a process model against this glossary. According to Tab. 3, for instance, there are 26238 label triples that always occur together and have been observed in at least ten process models. For the use case of suggesting labels for process model elements, we argued above that even clusters with a rather low confidence value are worth to be considered. Against the background of the results summarised in Tab. 3, however, it seems to be inevitable to consider only label clusters with a support value higher than a certain threshold. We see that even a support value of ten still results is numerous label triples.

The high number of label clusters with high support and confidence values is at least partly due to process models that contain a certain process fragment multiple times. A manual analysis of process models that were taken as input for the generation of the glossary revealed that several process models contain duplications of whole process fragments. As our notion of support is based on the number of occurrences of the labels in the model collection (cf. Sect. 3.3), the duplicated process fragments within one process model increase the respective support values notably. That, in turn, seems to be reasonable as the duplication of a whole process fragment can be seen as an indicator for a strong coupling of the respective labels in terms of co-occurrence.

Although the phenomenon of duplicated process fragments increased the observed support values for co-occurrence clusters, the high support values cannot be traced back to it in their entirety. That is, we observed a big overlap between the label clusters in the glossary and the test set. For instance, 64.90% of all label triples identified in the test set are already defined in the glossary.

We conclude that the high number of co-occurrence dependencies offers a huge potential for considering this information when using the glossary for label suggestions. In addition, we were able to identify a large number of a label clusters with a confidence value of one, which allow for a consistency analysis of the labelling of a given process model against the glossary.

Behavioural profile consistency. For the analysis of the behavioural profile relations in the glossary and in the test set, we could not consider all process models. In particular, EPC models with ambiguous instantiation semantics, cf. Decker and Mendling (2009), were neglected. Consequently, the behavioural relations are stored in the glossary based on 500 out of 675 process models, which led to behavioural relations for 58628 unique pairs of labels. Regarding the test set, the relations of the behavioural profile have been computed for 502 process models. That, in turn, led to behavioural relations for 90924 pairs of labels (note that they are not unique). Out of these 90924 label pairs, we could check the consistency of the behavioural relations for 14092 pairs of labels as those have been defined already in the glossary.

For 13458 label pairs the relation observed in the test set has been consistent with the relation stored in the glossary (cf. Sect. 3.4), which corresponds to a rate of 95.50%. We see that this number is in line with our observation for the SAP reference model. Thus, the assumption of an implicit ordering between labels does hold also for company specific process model collection. Consequently, the kind of information should be considered in a process model glossary.

5 Discussion

This section discusses the results of applying our approach and elaborates on limitations. We see that both glossaries generated as part of our case studies, in general, provide similar results when assessed in an experimental setting. Notable differences are observed for the level of reuse of labels and the co-occurrence dependencies between them.

Obviously, there is a trade-off between the effort needed to create a glossary and the effectiveness of applying the glossary for modelling support. Our approach minimises the effort for the generation and can be done in a fully automatic manner. Apparently, approaches that involve manual processing during the glossary creation will be superior with respect to the effectiveness of the modelling support. Albeit to a different extent, our case studies show a high reuse of labels as every second to every fifth label has been reused. This indicates a huge potential for modelling support compared to the absence of any controlled vocabulary. However, it also illustrates that there is a huge number of labels for which we are not able to derive suggestions based on the generated glossary. Clearly, a more fine-granular approach that builds upon single terms and some definition of valid relations between them (e.g., phrase structures) can be assumed to be more effective. Various of the labels that are not contained in our glossary directly might be derived by combining single terms of different labels that are already part of the glossary.

Besides this aspect, a second limitation of our approach has to be mentioned. Our approach focusses on a *consistent usage of labels* rather than on *consistency between labels*. We ensure that labels are applied correctly with respect to the type of the respective model element as well as co-occurrence and control flow dependencies between them. Still, our approach does not offer any control on the actual creation of labels and the consistency between different labels. Consequently, flaws such as misspellings, incomplete labels, labels that contain control flow information, or the usage of synonyms and homonyms are not addressed by our approach. Hence, it is of particular importance that the labels from which the glossary is generated are of high-quality.

Based thereon, implications can be drawn for practice as well as for research. On the one hand, we illustrated that even a lightweight approach for glossary generation offers a huge potential for modelling support. Once reference models or existing model collections are available, therefore, they should be leveraged for modelling support and integrated into modelling tools directly. On the other hand, our results highlight the importance of taking the structural and control flow aspects into account in order to increase the label quality. Research proposals often focus on labelling quality by taking solely single labels into account. Without doubt, this is a very important and challenging problem. Still, relations between labels, such as co-occurrences and control flow dependencies, should not be neglected.

6 Related Work

Our approach of using a glossary for process modelling aims at increasing the model quality by providing a centralised terminology. There has been a lot of research on the quality aspects of process models (cf. Bandara et al. (2004); Heravizadeh et al. (2008); Mendling et al. (2010b); Sedera et al. (2002)). Although quality of process models is affected by a whole spectrum of different factors, the importance of the element labelling for the model understandability and, therefore, model quality is not questioned.

Based on the SAP reference model that we used in one of our case studies, Mendling et al. (2010a) have investigated common phrase structures. 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 Malone et al. (2003). 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 (Decker and Mendling 2009). 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.

Similar to our approach, Delfmann et al. (2008), and Becker et al. (2009) describe a generic framework for defining a glossary of terms and phrase structures. Their work is motivated by naming conflicts in process models that are created in distributed teams. This approach has the advantage that fine-granular phrase-structures can be enforced in the labels of model elements, while linguistic phenomena such as synonymy are addressed using thesauri. Consequently, this approach enables control of the phrase structures that are used in element labels. Still, Becker et al. (2009) consider the generation of the glossary to be a manual task, which might require serious efforts. As discussed in the previous section, 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, structural and control flow aspects of process models are considered to ensure a high degree of labelling consistency and to increase the usefulness of term suggestions. Clearly, the approach presented in Delfmann et al. (2008), and Becker et al. (2009) focusses more on the consistency of labels, while a consistent usage of labels is not addressed extensively.

Other work aims at providing modelling support based on a repository of model patterns that are extracted based on the element labelling. In Sect. 3.3, we already reported on action patterns that are derived using association rules mining techniques (Smirnov et al. 2009). While these patterns inspired our approach, the concrete operationalisation is different due to our focus on modelling support for a narrow domain, instead of patterns of abstract actions. The focus on abstract and rather generic actions is an advantage of this approach, as it allows for reusing knowledge in a broad context. Once the question of automatic part of speech tagging has been addressed appropriately, these patterns might also be incorporated in our approach. Further on, Thom et al. (2009) propose a set of generic activity patterns that might be used as basic building blocks of process models. Based thereon, the detection of co-occurrence dependencies for these patterns is discussed in Lau et al. (2009). Similar ideas have also been presented in Becker et al. (2007), which introduces a process modelling language tailored for the public sector that is based on process building blocks. Here, process modelling is guided by restricting the set of available activities and control flow structures. As a consequence, process models are consistent with respect to the labels and the usage of labels by construction. However, this approach has to be scoped for a dedicated domain (e.g., the German public sector) which prevents an application in many other scenarios.

Support for process modelling might also be based on search techniques (Hornung et al. 2008). 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., Koschmider and Oberweis (2005). Again, such an ontology allows for coping with various linguistic phenomena such as synonymy, once a domain specific ontology is available. However, automatic generation of such an ontology imposes various challenges that go beyond the aforementioned issue of part of speech tagging.

7 Conclusion

In this article, we presented an approach to automatically generate a glossary from a process model collection. We argued that such a glossary can be applied for process modelling, either to check the labelling of a given process model against the glossary, or to provide support in terms of label suggestion features. In addition, the existence of reference models motivates our approach, as those models can be assumed to have labels of high quality for a dedicated domain. Further on, we advocated the enrichment of a glossary with structural and control flow aspects. That is, types of process model elements, co-occurrence dependencies, and behavioural relations between labels can be leveraged to provide more mature modelling support. We also presented two case studies to show the applicability of our approach and demonstrate the appropriateness of our choice on information stored in the glossary. In particular, our second case study showed that the results obtained for a reference model can be transferred to the case of a company specific model collection.

A glossary as proposed in this article can be generated automatically. It can be seen as a lightweight approach to achieve effective modelling support. In particular, our approach of narrowing the set of potential labels for a certain element based on various structural constraints and control flow information goes beyond pure textual querying. Even though our experiments provided evidence for the usefulness of the approach, future research has to evaluate the usage of such a glossary empirically in a user study.

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. Knauss et al. 2008) 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. To this end, recent work in the field of isolated label analysis (e.g., Friedrich 2009) might be lifted to the level of a glossary.

Finally, we foresee various ways to combine existing work on label quality with our approach. The information of relations between labels that is stored in our glossary might be exploited to increase the quality of results obtained by automatic part of speech tagging. For instance, a set of highly co-occurring labels that appear to be sorted in terms of their control flow dependencies might share a dedicated set of terms. In this case, it is likely that these terms describe a business object that is processed by different activities. Further on, the relation between labels and element types can also be leveraged. Probably, labels assigned to input and output data objects in (extended) EPCs can also be assumed to relate to business objects. Such techniques along with automatic part of speech tagging might prove valuable for splitting the complete phrases of the glossary into sub-phrases or even single terms. Such techniques cannot be assumed to result in a glossary that shows the same quality as a glossary that is manually created by domain experts (and includes semantic annotations or predefined phrase structures). However, they might result in a more fine-grained structure of the glossary and, therefore, more mature modelling support.

References

van der Aalst W. M. P. (1998) The Application of Petri Nets to Workflow Management. In: Journal of Circuits, Systems, and Computers 8(1), pp. 21–66

Agrawal R., Srikant R. (1994) Fast Algorithms for Mining Association Rules in Large Databases. In: Procceedings of the 20th International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc., San Francisco, pp. 487–499

Agrawal R., Imielinski T., Swami A. N. (1993) Mining Association Rules between Sets of Items in Large Databases. In: International Conference on Management of Data. Washington, pp. 207–216

Bandara W., Gable G. G., Roseman M. (2004) Factors and measures of business process modelling: model building through a multiple case study. In: European Journal of Information Systems 14(4), pp. 347–360

Becker J., Pfeiffer D., Räckers M. (2007) Domain Specific Process Modelling in Public Administrations – The PICTURE-Approach. In: Wimmer M., Scholl H. J., Grönlund Å. (eds.) Proceedings of the 6th International Conference on Electronic Government. Lecture Notes in Computer Science Vol. 4656. Springer, pp. 68–79

Becker J., Delfmann P., Herwig S., Lis L., Stein A. (2009) Towards Increased Comparability of Conceptual Models – Enforcing Naming Conventions through Domain Thesauri and Linguistic Grammars. In: De Marco M., Loebbecke C., Willcocks L. (eds.) 17th European Conference on Information Systems. Verona, pp. 2636–2647

Brill E. (1992) A Simple Rule-Based Part of Speech Tagger. In: 3rd Applied Natural Language Processing Conference, pp. 152–155

Curran T. A., Keller G., Ladd A. (1997) SAP R/3 Business Blueprint: Understanding the Business Process Reference Model. Prentice-Hall, New Jersey

Decker G., Mendling J. (2009) Process Instantiation. In: Data & Knowledge Engineering 68(9), pp. 777–792

Decker G., Overdick H., Weske M. (2008) Oryx – An Open Modeling Platform for the BPM Community. In: Procceedings of the 6th International Conference on Business Process Management. Lecture Notes in Computer Science Vol. 5240. Springer, Berlin, pp. 382–385

Delfmann P., Herwig S., Lis L., Stein A. (2008) Eine Methode zur formalen Spezifikation und Umsetzung von Bezeichnungskonventionen für fachkonzeptionelle Informationsmodelle. In: Modellierung betrieblicher Informationssysteme. Lecture Notes in Informatics Vol. 141. GI, pp. 23–38

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

Frank U. (1999) Conceptual Modelling as the Core of the Information Systems Discipline – Perspectives and Epistemological Challenges. In: Proceedings of the America Conference on Information Systems, pp. 695–698

Friedrich F. (2009) Measuring Semantic Label Quality Using WordNet. In: Proceedings of the 8th GI-Workshop Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten. CEUR Workshop Proceedings Vol. 554. Berlin

Heravizadeh M., Mendling J., Rosemann M. (2008) Dimensions of Business Processes Quality (QoBP). In: Procceedings of the 6th International Conference on Business Process Management. Lecture Notes in Business Information Processing Vol. 17. Springer, Berlin, pp. 80–91

Hornung T., Koschmider A., Lausen G. (2008) Recommendation Based Process Modeling Support: Method and User Experience. In: ER. Lecture Notes in Computer Science Vol. 5231. Springer, Berlin, pp. 265–278

Keller G., Nüttgens M., Scheer A.-W. (1992) Semantische Prozeßmodellierung auf der Grundlage 'Ereignisgesteuerter Prozeßketten (EPK)'. Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi) 89. Universität des Saarlandes

Kindler E. (2004) On the semantics of EPCs: A framework for resolving the vicious circle. In: Desel J., Pernici B., Weske M. (eds.) 2nd International Conference on Business Process Management. Lecture Notes in Computer Science Vol. 3080. Springer, Berlin, pp. 82–97

Knauss E., Meyer S., Schneider K. (2008) Recommending Terms for Glossaries: A Computer-Based Approach. In: Proceedings of the 2008 First International Workshop on Managing Requirements Knowledge. IEEE Computer Society, Washington, pp. 25–31

Koschmider A., Oberweis A. (2005) Ontology Based Business Process Description. In: Proceedings of the Open Interop Workshop on Enterprise Modelling and Ontologies for Interoperability. CEUR Workshop Proceedings Vol. 160. CEUR-WS.org

Lau J. M., Iochpe C., Thom L. H., Reichert M. (2009) Discovery and Analysis of Activity Pattern Co-occurrences in Business Process Models. In: Cordeiro J., Filipe J. (eds.) Proceedings of the 7th International Conference on Enterprise Information Systems, pp. 83–88

Leopold H., Smirnov S., Mendling J. (2009) On Labeling Quality in Business Process Models. In: Proceedings of the 8th GI-Workshop Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (EPK). CEUR Workshop Proceedings Vol. 554. Berlin

Malone T. W., Crowston K., Herman G. A. (2003) Organizing Business Knowledge: The MIT Process Handbook. The MIT Press, Cambridge

Mendling J., Reijers H., Recker J. (2010a) Activity labeling in process modeling: empirical insights and recommendations. In: Information Systems 35(4), pp. 467–482

Mendling J. (2008) Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness. Lecture Notes in Business Information Processing Vol. 6. Springer, Berlin

Mendling J., Reijers H. A., Cardoso J. (2007) What Makes Process Models Understandable? In: Procceedings of the 5th International Conference on Business Process Management. Lecture Notes in Computer Science Vol. 4714. Springer, Berlin, pp. 48–63

Mendling J., Verbeek H. M. W., van Dongen B. F., van der Aalst W. M. P., Neumann G. (2008) Detection and prediction of errors in EPCs of the SAP reference model. In: Data & Knowledge Engineering 64(1), pp. 312–329

Mendling J., Reijers H. A., van der Aalst W. M. P. (2010b) Seven process modeling guidelines

(7PMG). In: Information & Software Technology 52(2), pp. 127–136

Nüttgens M., Rump F. J. (2002) Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen – Promise 2002. Lecture Notes in Informatics Vol. 21. GI, pp. 64–77

Peters N., Weidlich M. (2009) Using Glossaries to Enhance the Label Quality in Business Process Models. In: Proceedings of the 8th GI-Workshop Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (EPK). CEUR Workshop Proceedings Vol. 554. Berlin

Recker J., Dreiling A. (2007) 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, pp. 356–366

Rosemann M. (2003) Preparation of process modeling. In: Becker J., Kugeler M., Rosemann M. (eds.) Process Management: A Guide for the Design of Business Processes. Springer, Berlin, pp. 41–78

Rosemann M., van der Aalst W. M. P. (2007) A configurable reference modelling language. In: Information Systems 32(1), pp. 1–23

Sedera W., Rosemann M., Gable G. G. (2002) Measuring Process Modelling Success. In: Proceedings of the 10th European Conference on Information Systems, pp. 331–341

Smirnov S., Weidlich M., Mendling J., Weske M. (2009) Action Patterns in Business Process Models. In: Baresi L., Chi C.-H., Suzuki J. (eds.) ICSOC/ServiceWave. Lecture Notes in Computer Science Vol. 5900, pp. 115–129

Stephens S. (2001) The Supply Chain Council and the Supply Chain Operations Reference Model. In: Supply Chain Management 1, pp. 9–13

Thom L., Reichert M., Iochpe C. (2009) Activity Patterns in Process-aware Information Systems: Basic Concepts and Empirical Evidence. In: International Journal of Business Process Integration and Management 4(2), pp. 93–110

Weidlich M., Mendling J., Weske M. (2010) Efficient Consistency Measurement based on Behavioural Profiles of Process Models. In: IEEE Transactions on Software Engineering To appear.

Nicolas Peters, Matthias Weidlich

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