Henrik Leopold, Sergey Smirnov, Jan Mendling

Recognising Activity Labeling Styles in Business Process Models

Quality assurance is a serious issue for large-scale process modelling initiatives. While formal control flow analysis has been extensively studied in prior research, there is little work on how the textual content of a process model and its activity labels can be systematically analysed. In this context, it is a major challenge to systematically identify and to consequently assure high label quality. As many large process model collections contain more than thousand models, each including several activity labels, there is a strong need for an automatic detection of labels that might be of bad quality. Recent research has shown that different grammatical styles correlate with potential ambiguity of a label. In this paper, we propose an algorithm for recognition of activity labeling styles. The developed algorithm exploits natural language processing techniques, e.g., part of speech tagging and analysis of the grammatical structure. We also study how ontologies, like WordNet, can support the solution. We conduct a thorough evaluation of the developed techniques utilising about 6,000 activity labels from the SAP Reference Model. The evaluation of this algorithm shows that spurious labels can be identified with a significant level of precision and recall. In this way, our approach can be used as a means of quality assurance for process repository management by listing bad quality labels, which a human modeler should correct.

1 Introduction

Business process modelling has become an integral part of process management initiatives in enterprises, in particular for documenting business operations (Davies et al. 2006). In many companies these initiatives have become so large that several thousand process models have to be maintained and that a significant amount of staff members is directly involved in modelling (Rosemann 2006). This fact implies considerable challenges for the quality management of the process repository. Most notably, the consistency of the process models can only be guaranteed when detailed modelling guidelines are provided.

Due to the sheer size of modelling projects, the number of people involved, and the number of models created, there is a strong need for automatic analysis techniques that can scan a process repository for quality issues. While an extensive set of research contributions have been made on verification techniques, (e.g., Verbeek et al. 2001; Wynn et al. 2009), error prediction (Mendling 2008), and comprehension issues (Gruhn and Laue 2007; Mendling et al. 2007), it is only recently that the style of activity labeling has become a focus of research. This quality aspect is of particular importance as labels contribute a great share to semantic and pragmatic usefulness of a model (Krogstie et al. 2006). Different works on process modelling guidelines and use case writing suggest a specific grammatical style of labeling to improve comprehension and to avoid ambiguity (Malone et al. 2003; Miles 1961; Phalp et al. 2007; Sharp and McDermott 2008). As a prerequisite for the efficient quality assurance of activity labels in large collections of process models, research needs to define techniques for the identification of a particular labeling style in order to support the *automatic* detection of labels that might be of bad quality.

A research gap in this area has been identified in recent works on applying standard parser and part of speech taggers for the automatic parsing of activity labels (Becker et al. 2009; Leopold et al. 2009). A problem of this approach is though that part of speech taggers require complete sentences to perform well, but activity labels often contain only two words. In this paper we refine the label parsing approach that we defined in Leopold et al. (2009) by parsing labels on a more fine-granular level. Our contribution is a parsing algorithm that is capable to identify labeling styles that we collected from the SAP Reference Model, a publicly available business process model collection capturing the processes using Event-Driven Process Chains (EPCs) (Keller and Teufel 1998). We evaluate the performance of our approach by automatically classifying the labels of this model collection and checking precision and recall.

The structure of the paper is as follows. Section 2 discusses labeling issues in a broader context and motivates this work using an example of a real-world process model. Section 3 reports which activity labeling styles can be found in practice and how they bear potential for ambiguous interpretation. Section 4 introduces the algorithms for activity labeling styles. Section 5 presents the results of an evaluation involving the SAP Reference Model. Section 6 discusses the findings in its relationship to the related work. Section 7 concludes the paper and gives an outlook on the future research.

2 Motivation

In this section we motivate the problem of activity labeling and automatic recognition of labeling styles. The problem of activity labeling can be generalised to the model element labeling, which has been discussed for system analysis and software engineering. There are various approaches to assure the high quality of labels in a model. They can be roughly categorised as a priori conventions and a posteriori diagnosis techniques. Furthermore, we illustrate the problem of activity labeling in process models by real world examples.

Conventions regulate various aspects of model design in an a priori way, for instance in terms of naming conventions for model element labeling. In software engineering, code conventions are a prominent technique for providing quality assurance. They prescribe the style, practices, and methods for development of software applications in this language, see e.g., Sun Microsystems (1999), Python Software Foundation (2001). Among many aspects regulated by code conventions are naming conventions for program entities. For instance, Sun Microsystems (1999) suggests to use the verbs for method naming. Another example of naming conventions appears in hierarchical input process output (HIPO) methodology. HIPO is a system analysis design and documentation technique developed by IBM in 1970s (IBM Corp 1974). HIPO advocates the usage of an active verb followed by a subject in a module name, where a module realises a certain activity in the program. Notice that the two given examples relate to design on different abstraction level: on the low level, program code, and on high level, system design. However, disregard of the model abstraction level, naming conventions suggest the use of verbs for the naming of model elements that capture an activity. In the domain of business process modelling Sharp and McDermott (2008) and Malone et al. (2003) argue that a consistent application of the verb-object labeling style increases understanding of activity labels. Verb-object labels are verb phrases headed by a verb infinitive and succeeded by a noun phrase. The verb captures an action, while the noun phrase a business object. Consider examples *Transfer plan data*, *Plan integration*, and *Create plan.* An action-noun label states the action as a noun, which can often be confused with a business object.

Once the system designers agree upon the naming conventions, it has to be assured that the conventions are actually followed, and in particular a specific labeling style. In this context, diagnosis techniques are required to facilitate the

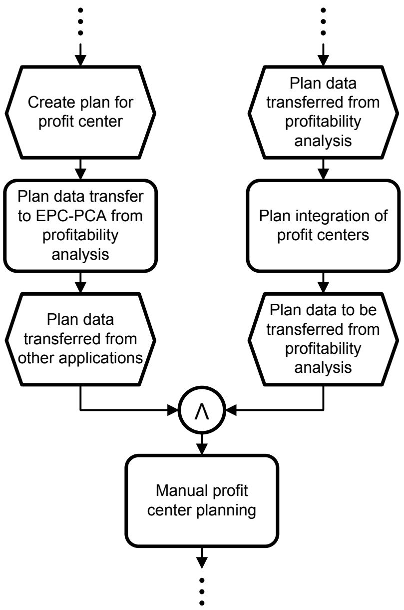
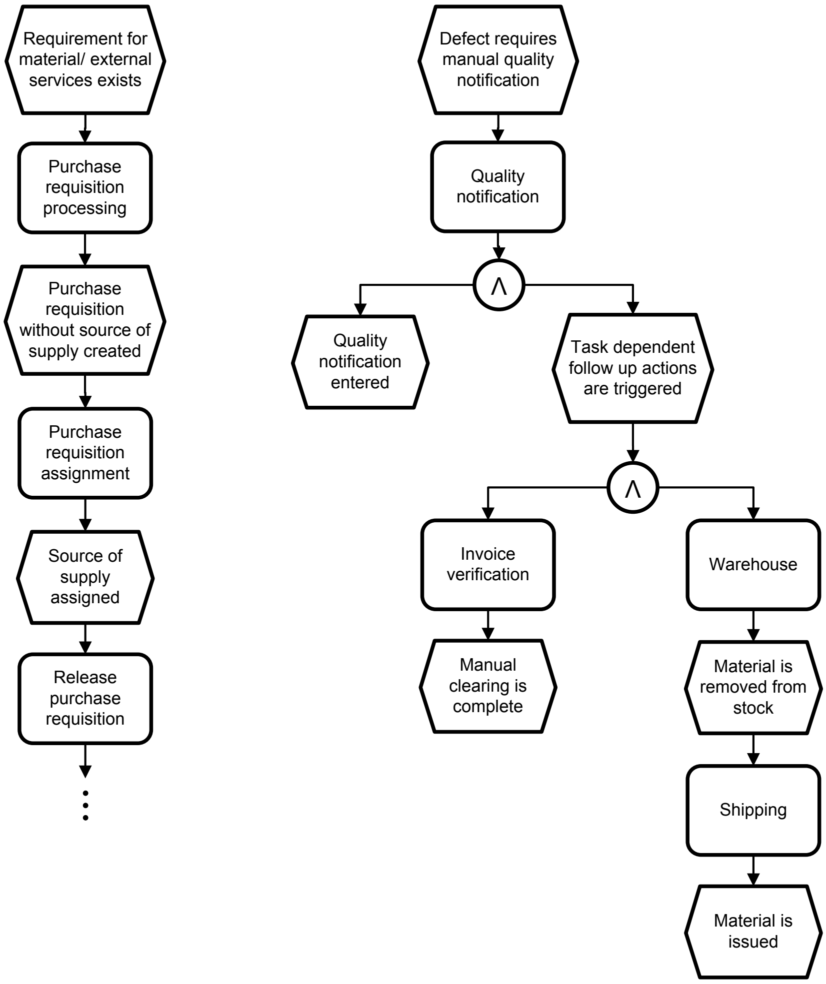


Figure 1: An example of poor labeling in the SAP Reference Model

analysis of single process models or whole process model repositories (Delfmann et al. 2009). Yet, an automatic recognition of labeling styles in business process models is missing so far. The outcome of the diagnostic technique can be used in different ways to provide quality assurance. First, the simplest approach is to report a warning on the possible problem directly to the designer. Second, the diagnostic information can be used for refactoring of poor quality labels. Labels that do not follow the desired style can be shown to the designer such that she can change them to the appropriate style. Towards the definition of automatic refactoring techniques, it is required to first refine a reliable approach to recognise the grammatical structure of the label.

To illustrate the activity labeling problem in process models we refer to a concrete motivating example. Consider a business process fragment presented in Figure 1. It captures a part of a profit center planning process from the SAP Reference Model (Keller and Teufel 1998). One can see, that it is easy to misinterpret activity label *Plan data transfer to EC-PCA from profitability analysis*. If the preceding and succeeding events would be ignored, a model reader might erroneously conclude that the label *Plan data transfer to EC-PCA from profitability analysis* instructs to *plan* a *data transfer*, and label *Plan integration of profit centers* advises to *plan* the *integration of profit centers*. However, event *Plan Data transferred from other Applications* reveals that the action in the activity on the left branch is given by noun *transfer*. Consequently, the activity label does not instruct to *plan* a *data transfer*, but to *transfer plan data*. This example illustrates a high ambiguity that partially stems from the style of labeling: in one case the first word is a verb referring to an action while in other cases the first word is a business object and the action is given as a noun. Implicitly, a reader of a model would assume a consistent usage of labeling styles. For this reason, the Guidelines of modelling emphasise the importance of a consistent design (Becker et al. 2000).

Figure 2 shows two further examples of business process models from the SAP Reference Model. Figure 2(a) depicts a fragment of a business process where a *purchase requisition* is handled. Within this model fragment we observe activity labels *purchase requisition processing*, *purchase requisition assignment* and *release purchase requisition*. In the first two labels the actions are denoted with the nouns *processing* and *assignment*. In the third one the verb *release* corresponds to the action. Obviously, the modelers used several styles for activity labeling. Ambiguity is a potential threat to label understanding. For instance, consider the word *purchase*, which can be both a noun and a verb. This source of ambiguity is called zero derivation, since a verb is linguistically created from a noun without adding a postfix like -*ize* in *computerize*. It has been pointed out that different styles are prone to different degrees of ambiguity (Mendling et al. 2009), which emphasises the importance of labeling styles for human



(a) Fragment of process model *Purchase Requisition* (b) Model of process *Return Deliveries*

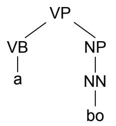
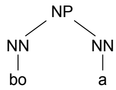
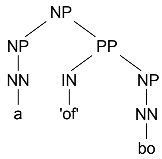
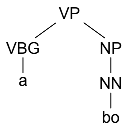
Figure 2: Labeling in two business process model examples from the SAP Reference Model

understanding as stressed early in Sharp and McDermott (2008) and Malone et al. (2003). If an action noun is used, there is likely an ambiguity, when it is combined with a zero derivation noun. If we consider the *purchase requisition processing* label, it is hard to tell if *purchase* or *processing* stands for an action. As zero derivation is an essential part of the language that cannot always be avoided, it is a useful strategy to employ and enforce a suitable labeling style.

Figure 2(b) highlights some further potential problems of labels. It captures a model of the *return deliveries* business process. In this model, we can observe activity labels that do not signify any action (e.g., *warehouse*) or activities with actions, but without any object (e.g., *shipping*). Again understanding of such activities requires the reader to interpret the context of the model.

Diagnosis techniques that are able to recognise different labeling styles can the valuable to reveal such labeling issues. This can include the detection of inconsistent usage of labeling styles, within a single model or in the whole repository, as well as identification of labels that do not comply with the verb-object style.

Table 1: Activity labeling styles

Name	Structure	Example
Verb phrase		Create invoice
Noun phrase		Invoice creation
Noun phrase with *of* prepositional phrase		Creation of invoice
Verb phrase (gerund)		Creating invoice
Irregular	-	LIFO: Valuation: Pool level

3 Activity Labeling Styles

Development of effective algorithms for recognition of labeling styles requires a thorough understanding of current labeling practices. We approached this problem in a bottom-up way by investigating the different verb-object and action-noun labels of the SAP Reference Model. This model collection includes models of business processes, as they are supported by the SAP R/3 software package in its version from the year 2000. The collection is organised in 29 functional branches of an enterprise, including sales, accounting, and other functional areas. As we reason about the activity label using its context, i.e., labels of neighbouring events, we have studied a subset of the SAP Reference Model: 604 Event-Driven Process Chains[1]. Each of the considered EPCs contains several activities, events, and the control flow. Table 1 shows that we found five activity labeling styles, each having a particular structure.

The labels of *verb phrase* style contain an action that is followed by a business object. Examples of *verb phrase* labels are *Create invoice* and *Validate order*. In the first case the action is *create* and the business object is *invoice*, while in the second example the action is *validate* and the business object is *order*. Notice that a business object may be absent. Consider labels *Analyse* or *Notify*. As these labels are also verb phrases, we relate them to the same style. A special case are verb phrases which contain a prepositional phrase, e.g., *Create order for received request*. A prepositional phrase in such labels brings additional information to the reader; it is optional.

The labels of *noun phrase* style are start with a business object followed by an action. Examples of label adhering to this style are *Vendor evaluation* and *Schedule approval*. In the first case the action is *evaluate* and the business object is *vendor*, while in the second example the action is *approve*

[1]In earlier work we used the larger set of models from the SAP Reference Models as the techniques proposed in (Leopold et al. 2009) do not require events to be in the model.

Table 2: Properties of activity label styles

Label class	Property
Verb phrase	the leading word is a verb
Noun phrase	none
Noun phrase with *of* prepositional phrase	label contains a prepositional phrase with *of* as a leading preposition
Verb phrase (gerund)	the leading gerund signifies an action and is followed by a business object
Phrase with coordinating conjunction	the phrase contains a coordinating conjunction, e.g., *and* or *or*
Irregular	label contains characters ':' or ' - '

and the business object is *schedule*. Notice that a business object may be absent. Consider labels *Analysis* or *Notification*. As these labels are also noun phrases, we relate them to the same substyle. Another special case are noun phrases which contain a prepositional phrase, e.g., *Revenue planning in work breakdown structure*. A prepositional phrase in such labels brings additional information to the reader; it is optional.

The labels of *Noun phrase with of prepositional phrase* style are a specific kind of a noun phrase. However, the action is represented by a noun which comes first and is succeeded by a prepositional phrase. The prepositional phrase is headed by a preposition *of* and refers to a business object. Examples are *Creation of specification* and *Settlement of order*. For the two given examples the actions are *create* and *settle*, respectively, and business objects are *specification* and *order*. Similar to the labels of the previous labeling style, the labels of noun phrase with *of* prepositional phrase style can have optional prepositional phrase, e.g., *Creation of specification for budget planning*. Again, the prepositional phrase is optional.

The labels of *verb phrase (gerund)* style are by a verb in *-ing* form. This gerund is succeeded by the business object captured as a noun. The following labels are examples of this class: *Creating version* and *Processing requisition for projects*. For the first label the action is *create* and the business object is *version* while in the second example the action is *process* and the business object is *requisition*. Notice that the label of this style may have an optional prepositional phrase (e.g., as *for projects* in *Processing requisition for projects*).

The styles described above cover almost 97% of all labels in the model collection. However, about 3% of the labels cannot be assigned to one of these substyles and are related to *irregular* style. The specific property of these labels is the use of characters connecting different parts of the label in a sometimes ambiguous way. Hence, these characters do not allow labels to qualify into any of the above named substyles. Examples are *Transfer Posting FI-LC*, *Profit Center Assessment: Plan*, or *LIFO: Valuation: Pool Level*. A majority of irregular labels can be recognised by the use of the characters ":" and "-".

Some labels refer to more than one business object or instruct to perform more than one action. Such labels contain a conjunction, coordinating the relations between homogeneous parts. Examples of conjunctions are *and*, *or*, comma symbol, and slash symbol. A conjunction may combine different parts of a phrase. Hence, the conjunction may appear in all previously defined labeling styles. Consider example labels *Project monitoring and controlling* and *Installation, dismantling and modification of equipment*. The first label refers to two actions and can be decomposed into two labels *Project monitoring* and *Project controlling*, both of which are noun phrases. The

Algorithm 1 Recognition of verb object style

```
1:  recogniseStyle(Set modelActivityLabels)
2:  unrecognisedLabels = ∅;
3:  VERBS = ∅;
4:  NOUNS = ∅;
5:  for all label ∈ modelActivityLabels do
6:     if label.words[1] can be an imperative verb then
7:        if label.environment.getPOS(label.words[1]) == VERB then
8:           VERBS = VERBS ∪ {label.words[1]};
9:           label.style = VERB_OBJECT;
10:       else if label.environment.getPOS(label.words[1]) == NOUN then
11:          NOUNS = NOUNS ∪ {label.words[1]};
12:          label.style = recogniseSubstyle(label);
13:       else
14:          unrecognisedLabels = unrecognisedLabels ∪ {label};
15: for all label ∈ unrecognisedLabels do
16:    if label.words[1] can be an imperative verb then
17:       if label.words[1] ∈ VERBS then
18:          label.style = VERB_OBJECT;
19:       else if label.words[1] ∈ NOUNS then
20:          label.style = recogniseSubstyle(label);
21:       else
22:          if WordNet.getBestPOS(label.words[1]) == VERB then
23:             label.style = VERB_OBJECT;
24:          else
25:             label.style = recogniseSubstyle(label);
```

label in the second example can be decomposed into three labels, which are noun phrases with *of* prepositional phrase style: *Installation of equipment*, *Dismantling of equipment*, and *Modification of equipment*.

Table 2 summarises the identified labeling styles along with their key properties.

4 Labeling Style Recognition

In this section we present a method for automatic recognition of labeling styles. The method is described by two algorithms. While Algorithm 1 specifies an overall approach to recognition of verb-phrase activity labels, Algorithm 2 focuses on recognition of style where an action is given as a noun.

An assumption of Algorithm 1 is that a label either belongs to a verb-object style or to any other kind of action-noun styles. Hence, the algorithm classifies the activity labels of a process model into one of these two categories. The input of the algorithm is the set of activity labels in one model *modelActivityLabels*. The result of the algorithm is the set of activity labels with recognised styles. Algorithm 1 has two phases. In the first phase the environment is used to recognise labeling styles and collect information for resolving ambiguous labels (lines 5–14). In the second phase the gathered information is used to recognise the labeling styles of the remaining activity labels (lines 15–24).

Within the first phase each activity label is analysed by means of the environment: its prece-

Algorithm 2 Recognition of action-noun substyles

```
1: recogniseSubstyle(Label label)
2: if label contains ':' OR ' - ' then
3:    return UNCLASSIFIED;
4: if label contains prepositions then
5:    hasPrepositions = true;
6:    pIndex = getFirstPrepositionIndex(label);
7: if label contains conjunctions then
8:    hasConjunctions = true;
9:    cIndex = getConjunctionIndex(label);
10: if first word in label has suffix 'ing' then
11:    hasSuffixING = true;
12: verbSize = getVerbSise(label);
13: if hasSuffixING and label.size > verbSize and (!hasPrepositions or pIndex > verbSize + 1) then
14:    if verb == action derived from label context then
15:       return GERUND;
16: if hasPrepositions and label.getWordAt(pIndex) == 'of' then
17:    return PREPOSITION_OF;
18: return NOUN;
```

ding and succeeding events. The event labels are tagged by a part of speech tagger, Stanford Tagger (Toutanova and Manning 2000). If the first word of the activity label appears in the labels of preceding/succeeding events, and it is tagged as a verb, the word is included into *VERB* set. The activity label is assigned to the verb-object style (lines 7–9). If the leading word appears in the labels of preceding/succeeding events, and it is tagged as a noun, the word is included into *NOUN* set. The activity label is assumed to follow an action-noun style, and it is further analysed (lines 10–12). If the environment does not help to learn the label style, the label has to be analysed in the second phase.

Sets *VERBS* and *NOUNS* accumulate information about the usage of words in the model labels. Set *VERBS* contains label leading words that are usually used as verbs within the given model, while set *NOUNS* contains the same information about nouns.

The second phase inspects those labels with a still unrecognised labeling style. Once the leading word of a label is contained in set *VERBS*, the style of the corresponding label is recognised as verb-object. If the leading word appears among *NOUNS*, the labeling style is further investigated by function *recogniseSubstyle*. If the label has not been assigned to a style yet, the algorithm makes use of WordNet information about the most probable part of speech for a given word (lines 21–24).

Algorithm 2 formalises label style recognition. The input of the algorithm is an action-noun label *label*, the output—the labeling style of *label*. We assume that all the flags are initiated with *false*.

First the algorithm examines, if the label contains characters that allow to classify the label as irregular (see lines 3–5). If the label contains such characters, the style of the label is irregular and the algorithm terminates. Otherwise, the algorithm continues seeking for prepositions (lines 6–8) and conjunctions (lines 9–11). If conjunctions or prepositions are found, respective flags *hasConjunctions* and *hasPrepositions* are set to true. If conjunctions/prepositions are available, the position of the first conjunction/preposition is stored in *pIndex*/*cIndex*.

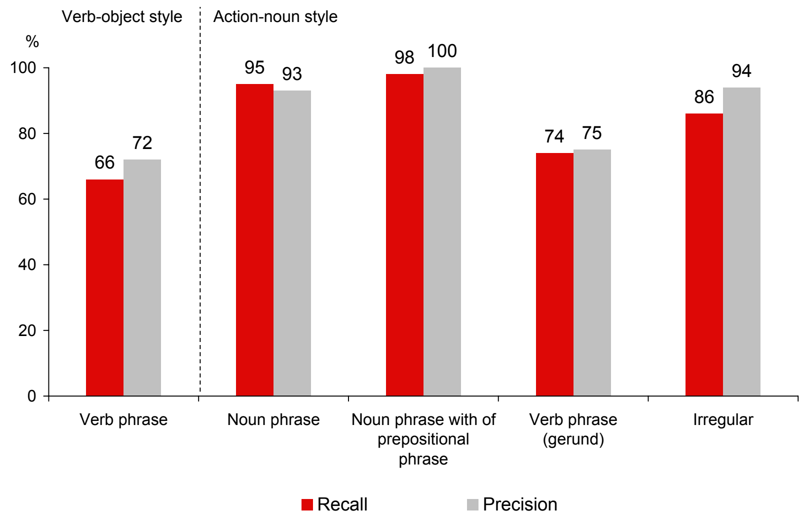


Figure 3: Precision and recall of algorithms for label style recognition

The algorithm proceeds checking, if the label starts with a gerund (lines 12–18). It is verified, if the first word of the label has an *-ing* suffix. Next, WordNet (Miller 1995) is used to learn, if the first word is a verb and which infinitive it has. An assessment whether the gerund represents an action requires a deeper investigation: if the first word of a label is a gerund, it does not imply that this word also represents the action. Consider label *Planning scenario processing*. Although *planning* is a gerund, it might also be a part of a business object. In order to resolve this ambiguity, we consider event nodes preceding and succeeding the activity with the inspected label. Returning to the example, we notice that the activity is preceded by an event labeled with *Planning scenario is processed*. A part of speech analysis of this label identifies *planning* and *scenario* as nouns and *process* as a verb. Hence, we can infer that *processing* captures an action.

If the algorithm qualifies a label to be a gerund, it terminates. In the opposite case, the algorithm proceeds checking prepositions in the label (lines 19–21). A label containing prepositions and the first of which is *of* is qualified as a *noun phrase with of prepositional phrase*. If the label is categorised to none of the enumerated styles, the algorithm refers it to a *noun phrase* style.

5 Empirical Evaluation

To validate the proposed algorithm, we have conducted an experiment. The goal of the experiment was to test how well the algorithm recognises the different labeling styles. We have designed a test collection that includes the process models of the SAP Reference Model, and a classification of activity labels according to styles as identified by two researchers. Figure 3 illustrates the classification of activity labels. Human interpretations of activity labels is captured by a mapping from an activity label to a labeling style. This information is stored in a spreadsheet, which is read by an application in the evaluation phase. Within the evaluation we compared recognition of label styles by the algorithm and by humans.

To evaluate the style recognition algorithm we measured the *precision* and *recall* of the algorithm (Baeza-Yates and Ribeiro-Neto 1999). The precision of the style recognition algorithm is the number of correctly recognised labels of the given style retrieved by the algorithm divided by the total number of labels retrieved. The recall is the number of correctly recognised labels of the given style retrieved by the algorithm divided by the total number of existing labels of this style.

Figure 3 presents the values of precision and recall obtained for the SAP Reference Model. It can be seen that noun phrase labels can be very accurately recognised with precision and recall of almost 100%. Verb phrases, both with gerund and with infinitive, are more difficult to identify, partially due to zero-derivation ambiguity, with values around 70%. These values indicate that the recognition techniques presented in this paper are well suited to identify labels of potentially bad quality, which can be then checked and reworked by process designers.

6 Related Work

Our work can be related to four major streams of related work: quality frameworks, process model labeling, process comparison, and natural language approaches for models.

Process model quality is discussed in different works on *quality frameworks*. The SEQUAL framework builds on semiotic theory and defines several quality aspects (Krogstie et al. 2006; Lindland et al. 1994). In essence, syntactic quality relates to model and modelling language, semantic quality to model, domain, and knowledge, and pragmatic quality relates to model and modelling and its ability to enable learning and action. The semantic and pragmatic quality clearly point to the relevance of labeling activities. The Guidelines of modelling define an alternative quality framework that is inspired by general accounting principles (Becker et al. 2000). The guidelines include the six principles of correctness, clarity, relevance, comparability, economic efficiency, and systematic design, where several of them have implications for good labeling. Also the ISO 9126 (ISO 1991) quality standard has been suggested as a starting point for model quality (Moody 2005; Selçuk Güceglioglu and Demirörs 2005). Our approach complements these more general works by introducing an algorithm that can be used to enforce labeling styles that are considered to be of good quality.

The verb-object style is widely promoted in the literature on *labeling of activities in process models* (Malone et al. 2003; Miles 1961; Sharp and McDermott 2008). Similar conventions are advocated as guidelines for the creation of understandable use case descriptions (Phalp et al. 2007). But in contrast to its promotion in the process modelling domain, it has been observed that verb-object labeling in real process models is not consistently applied, which might be a result of the lack of diagnosis techniques. For instance, the practical guide for process modelling with ARIS (Davis 2001, pp.66-70) shows models with both actions as verbs and as nouns. Also the use of ontologies as a point of reference for activity labels has been promoted (Fillies et al. 2003). Then, there are several recent works on the automatic analysis of activity labels. For instance, it has also been shown that shorter activity labels improve model understanding (Mendling and Strembeck 2008), which is consistent with readability assessments on sentence length (Flesch 1951; Gretchen 2000). The concept of part of speech tagging is also investigated for interactive process modelling support (Leopold et al. 2009) and for autocompletion (Becker et al. 2009). The general significance of labeling styles has been established in Mendling et al. (2009). The authors show that verb-object style labels are less likely to be considered as ambiguous and perceived as more useful than labels following another style. Our proposed algorithm complements this stream of normative and empirical research with an approach to efficiently recognise bad style activity labels even in large process repositories.

Linguistic analysis of activity labels is also an important step in process model comparison (see Aalst et al. 2002; Grossmann et al. 2005; Pankratius and Stucky 2005; Preuner et al. 2001). It also relates to the general matching problem as being discussed for conceptual schemas (Euzenat and Shvaiko 2007; Rahm and Bernstein 2001). Identifying correspondence between activities of a pair of process models is a prerequisite for calculating a degree of similarity or integrating the two models. This can be done manually or at least partially automatically. Linguistic analysis of labels plays an important role in the work by Ehrig et al. (2007), Dongen et al. (2008), and Dijkman et al. (2009). Our approach has the potential to be informative for this stream of research. The automatic recognition of different labeling styles could be useful to improve automatic approaches to matching labels of process models.

The enforcement of the verb-object style might help to close the gap between *natural language and formal language processing*. And indeed, the relationship between process models and natural language has been discussed and utilised in various works. In Fliedl et al. (2005) the authors investigate in how far the three steps of building a conceptual model (linguistic analysis, component mapping, and schema construction) can be automated using a model for pre-design. Further text analysis approaches have been used to link activities in process models to document fragments (Ingvaldsen et al. 2005) and to compare process models from a semantic perspective (Ehrig et al. 2007). A consistent usage of the verb-object style can be helpful for model verbalisation and paraphrasing, see Halpin and Curland (2006), and Frederiks and Weide (2006). Such verbalisation is an important step in model and requirements validation (Nuseibe and Easterbrook 2000). For instance, verb-object style labels can easily be verbalised using the *You have to* prefix, which we also used in our analysis. In this way, automatic recognition of labeling styles can provide for a better validation of process models.

7 Conclusion and Future Work

Recent research revealed the high impact of the labeling style in business process models on the overall model understanding (Mendling et al. 2009). The problem in this area is that automatic techniques have been missing for an accurate recognition of these different styles. In this paper we present algorithms for automatically classifying a label as verb-object style or several subtypes of action-noun styles. For the evaluation of the techniques, we use the SAP Reference Model. The results show that high precision and recall can be achieved automatically using the techniques that we propose. In this way, our technique can be a useful aid to enforce a particular style of labeling or to support quality assurance of a process repository.

There are several complementary directions of research to our work. Business process models contain other model elements beyond activities, like events and data objects, that should be subject to label quality assurance. It is part of our future research agenda to identify how part of speech tagging techniques can be applied for those labels as well. Our results also depend on using English as a language for labeling activities. It will be an interesting task to analyse other languages, like German or Russian, to see whether part of speech tagging can be utilised with the same accuracy. Finally, we plan to use tagging information for building taxonomies for process model collections. Identifying nouns from the label will be a crucial step for this application.

References

van der Aalst W., van Hee K., van der Toorn R. (2002) Component-Based Software Architectures: A Framework Based on Inheritance of Behavior. In: Science of Computer Programming 42(2–3), pp. 129–171

Baeza-Yates R., Ribeiro-Neto B. (1999) Modern Information Retrieval. Addison-Wesley, Boston

Becker J., Rosemann M., von Uthmann C. (2000) Guidelines of Business Process Modeling. In: Business Process Management. Lecture Notes in Computer Science Vol. 1806. Springer, Berlin, pp. 30–49

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: Procceedings of the 17th European Conference on Information Systems. Verona

Davies I., Green P., Rosemann M., Indulska M., Gallo S. (2006) How Do Practitioners Use Conceptual Modeling in Practice? In: Data & Knowledge Engineering 58(3), pp. 358–380

Davis R. (2001) Business Process Modelling with ARIS: A Practical Guide. Springer, Berlin

Delfmann P., Herwig S., Lis L. (2009) Konfliktäre Bezeichnungen in Ereignisgesteuerten Prozessketten – Linguistische Analyse und Vorschlag eines Lösungsansatzes. In: GI-Workshop EPK 2009: Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten. Berlin, pp. 178–194

Dijkman R. M., Dumas M., García-Bañuelos L. (2009) Graph Matching Algorithms for Business Process Model Similarity Search. In: Business Process Management, pp. 48–63

van Dongen B. F., Dijkman R. M., Mendling J. (2008) Measuring Similarity between Business Process Models. In: Proceedings of the 20th International Conference on Advanced Information Systems Engineering (CAiSE '08). Lecture Notes in Computer Science Vol. 5074. Springer, Montpellier, pp. 450–464

Ehrig M., Koschmider A., Oberweis A. (2007) Measuring Similarity between Semantic Business Process Models. In: Procceedings of the 4th Asia-Pacific Conference on Conceptual Modelling. Australian Computer Science Communications, Ballarat, Victoria, pp. 71–80

Euzenat J., Shvaiko P. (2007) Ontology Matching. Springer, Berlin

Fillies C., Wood-Albrecht G., Weichhardt F. (2003) Pragmatic Applications of the Semantic Web using SemTalk. In: Computer Networks 42(5), pp. 599–615

Flesch R. (1951) How to Test Readability. Harper & Brothers, New York

Fliedl G., Kop C., Mayr H. (2005) From Textual Scenarios to a Conceptual Schema. In: Data & Knowledge Engineering 55(1), pp. 20–37

Frederiks P., Weide T. (2006) Information Modeling: The Process and the Required Competencies of Its Participants. In: Data & Knowledge Engineering 58(1), pp. 4–20

Gretchen H. (2000) Readability and Computer Documentation. In: ACM Journal of Computer Documentation 24(3), pp. 122–131

Grossmann G., Ren Y., Schrefl M., Stumptner M. (2005) Behavior Based Integration of Composite Business Processes. In: van der Aalst W., Benatallah B., Casati F., Curbera F. (eds.) Business Process Management 2005. Lecture Notes in Computer Science Vol. 3649. Springer, Nancy, pp. 186–204

Gruhn V., Laue R. (2007) What Business Process Modelers Can Learn from Programmers. In: Sci. Comput. Program. 65(1), pp. 4–13

Halpin T., Curland M. (2006) Automated Verbalization for ORM 2. In: On the Move to Meaningful Internet Systems 2006. Lecture Notes in Computer Science Vol. 4278. Springer, Berlin, pp. 1181–1190

IBM Corp (1974) HIPO – A Design Aid and Documentation Technique GC20-1851 White Plains

ISO (1991) Information Technology - Software Product Evaluation - Quality Characteristics and Guide Lines for Their Use, ISO/IEC IS 9126

Ingvaldsen J., Gulla J. A., Su X., Rønneberg H. (2005) A Text Mining Approach to Integrating Business Process Models and Governing Documents. In: On the Move to Meaningful Internet Systems 2005. Lecture Notes in Computer Science Vol. 3762. Springer, Berlin, pp. 473–484

Keller G., Teufel T. (1998) SAP(R) R/3 Process Oriented Implementation: Iterative Process Prototyping. Addison-Wesley, Berlin

Krogstie J., Sindre G., Jørgensen H. (2006) Process models representing knowledge for action: a revised quality framework. In: European Journal of Information Systems 15(1), pp. 91–102

Leopold H., Smirnov S., Mendling J. (2009) On Labeling Quality in Business Process Models. In: GI-Workshop EPK 2009: Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten. CEUR Vol. 554, pp. 42–57

Lindland O., Sindre G., Sølvberg A. (1994) Understanding Quality in Conceptual Modeling. In: IEEE Software 11(2), pp. 42–49

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

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., Strembeck M. (2008) Influence Factors of Understanding Business Process Models. In: Business Information Systems. Lecture Notes in Business Information Processing Vol. 7. Springer, pp. 142–153

Mendling J., Reijers H. A., Cardoso J. (2007) What Makes Process Models Understandable? In: Business Process Management 2007. Milan, pp. 48–63

Mendling J., Reijers H. A., Recker J. (2009) Activity Labeling in Process Modeling: Empirical Insights and Recommendations. In: http://eprints.qut.edu.au/19625/

Miles L. D. (1961) Techniques of Value Analysis and Engineering. McGraw-Hill, New York

Miller G. A. (1995) WordNet: a Lexical Database for English. In: Communications of the ACM 38(11), pp. 39–41

Moody D. (2005) Theoretical and Practical Issues in Evaluating the Quality of Conceptual Models: Current State and Future Directions. In: Data & Knowledge Engineering 55(3), pp. 243–276

Nuseibe B., Easterbrook S. M. (2000) Requirements Engineering: A Roadmap. In: International Conference on Software Engineering. ACM Press, Park Falls, pp. 35–46

Pankratius V., Stucky W. (2005) A Formal Foundation for Workflow Composition, Workflow View Definition, and Workflow Normalization based on Petri Nets. In: Hartmann S., Stumptner M. (eds.) Procceedings of the 2nd Asia-Pacific Conference on Conceptual Modelling. Conferences in Research and Practice in Information Technology Vol. 43. Australian Computer Society, Newcastle

Phalp K. T., Vincent J., Cox K. (2007) Improving the Quality of Use Case Descriptions: Empirical Assessment of Writing Guidelines. In: Software Quality Journal 15(4), pp. 383–399

Preuner G., Conrad S., Schrefl M. (2001) View Integration of Behavior in Object-Oriented Databases. In: Data & Knowledge Engineering 36(2), pp. 153–183

Python Software Foundation (2001) Style Guide for Python Code. Last Access: http://www.python.org/dev/peps/pep-0008/

Rahm E., Bernstein P. A. (2001) A Survey of Approaches to Automatic Schema Matching. In: Very Large Data Bases Journal 10(4), pp. 334–350

Rosemann M. (2006) Potential Pitfalls of Process Modeling: Part A. In: Business Process Management Journal 12(2), pp. 249–254

Selçuk Güceglioglu A., Demirörs O. (2005) Using Software Quality Characteristics to Measure Business Process Quality. In: Business Process Management. Lecture Notes in Computer Science Vol. 3649. Springer, Berlin, pp. 374–379

Sharp A., McDermott P. (2008) Workflow Modeling: Tools for Process Improvement and Applications Development. Artech House Publishers, Norwood

Sun Microsystems (1999) Code Conventions for the Java Programming Language. Last Access: http://java.sun.com/docs/codeconv/

Toutanova K., Manning C. (2000) Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In: Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 63–70

Verbeek H., Basten T., Aalst W. (2001) Diagnosing Workflow Processes using Woflan. In: The Computer Journal 44(4), pp. 246–279

Wynn M. T., Verbeek H., Aalst W., Hofstede A., Edmond D. (2009) Soundness-preserving Reduction Rules for Reset Workflow Nets. In: Inf. Sci. 179(6), pp. 769–790

Henrik Leopold, Jan Mendling
Humboldt-Universität zu Berlin,
Germany
henrik.leopold@student.hu-berlin.de
jan.mendling@wiwi.hu-berlin.de

Sergey Smirnov
Hasso Plattner Institute,
Potsdam,
Germany
sergey.smirnov@hpi.uni-potsdam.de