

# Refactoring of Activity Labels in Business Process Models

Henrik Leopold<sup>1</sup>, Sergey Smirnov<sup>2</sup>, and Jan Mendling<sup>1</sup>

<sup>1</sup> Humboldt-Universität zu Berlin, Germany

{henrik.leopold,jan.mendling}@wiwi.hu-berlin.de

<sup>2</sup> Hasso Plattner Institute, Potsdam, Germany

sergey.smirnov@hpi.uni-potsdam.de

**Abstract.** Many companies have recently expanded their business process modeling projects such that often thousands of process models are designed and maintained. The problem with such initiatives is that staff with limited experience takes part in modeling, with a detrimental effect on the quality of the models. The quality of activity labels, i.e., the textual descriptions in a process model, is an important quality aspect in this context. Activity labels can be related to different styles according to their grammatical structure. There are several guidelines that suggest using a so-called verb-object labeling style and a recent study establishes its superiority in terms of human comprehension. Meanwhile, real-world process models often include labels that do not follow this style, but rather mention the action as a noun. In this paper, we investigate the potential to automatically improve the label quality. We define and implement an approach for automatic refactoring of labels following action-noun style into verb-object labels. We evaluate the proposed techniques using a collection of real-world process models—the SAP Reference Model.

## 1 Introduction

Business process modeling has become an integral part of process management in large enterprises. Many of these companies design and maintain up to several thousand process models that capture their operations like sales, production, accounting, and strategic planning [21]. The sheer size of these initiatives poses new challenges to quality assurance of process models. When modeling experts were still the ones to create process models for dedicated purposes, there was little concern with model quality. Nowadays, laymen and casual modelers design a great share of process models in these large scale initiatives with detrimental consequences for model quality. Due to the large number of models, there is a growing need for automatic techniques to check and improve the quality of process models.

Process model quality has been approached from different angles. Formal properties such a soundness of control flow are understood quite well and can be efficiently decided using Petri net analysis techniques [1]. More user-centric

characteristics, like error probability and comprehension, have been traced back to expertise and model complexity [14, 15]. Also the small pieces of text that capture the names of activities (activity labels) have been investigated from a usability perspective. Such activity labels represent actions, which take place during the execution of a business process. Typically, an activity label captures an action and a business object, on which the action is performed, like *Validate address* or *Creation of order*. In essence three classes of activity labels have been found in practice: verb-object labels, action-noun labels, and a rest category [16]. In a label following the verb-object style, an activity is defined as a verb phrase, e.g., *Validate address* or *Handle order*. Other labels follow an action-noun style, where the label is a noun phrase. Examples are *Comparison of object* or *Asset maintenance*. The interesting point is that [16] shows that verb-object labels are superior to action-noun labels in terms of perceived ambiguity. Therefore, it is desirable that all labels follow the verb-object style.

In this paper we address the problem of automatically refactoring action-noun labels to verb-object labels. A key challenge in this context is the identification of actions and business objects in action-noun labels. As activity labels in process models are only fragments of a sentence, we use contextual information from neighboring control flow elements instead of standard natural language parsing techniques. The algorithms of our approach have been implemented and evaluated using the SAP Reference Model, a large collection of real-world business process models [11]. The results emphasize the potential of our approach.

The rest of the paper is structured as follows. Section 2 illustrates the research problem and presents the main substyles of action-noun style that we identified in the SAP Reference Model. Section 3 presents the main contributions of this work—algorithms for label substyle recognition, action and business object derivation, and refactoring methods. In Section 4 we evaluate the presented algorithms. Related work is discussed in Section 5. Section 6 concludes the paper.

## 2 Background

In this section we outline the problem of label refactoring. The section starts with a motivating example that introduces different labeling styles. Furthermore, we present the results from a study of action-noun labels in the SAP Reference Model. This study identifies the main labeling action-noun substyles. For each substyle we describe its structure and provide supporting examples of labels.

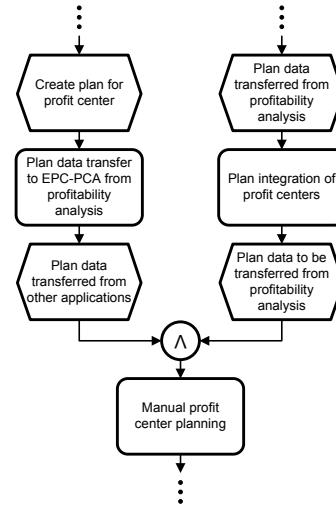
## 2.1 Motivation

The problem of activity label quality can be motivated by the following example. Consider a business process fragment presented in Fig. 1. It captures a part of a profit center planning process. One can see, that it is easy to misinterpret activity label *Plan data transfer to EC-PCA from profitability analysis*. Ignoring the preceding and succeeding events, a model reader might 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: once 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.

[16] argues that consistent application of a verb-object labeling style increases understandability 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. To enhance the overall quality of labels, we propose to refactor labels of potential ambiguity. In particular, we address widely used action-noun labels and present methods for their analysis and refactoring.

## 2.2 Label Classification

Development of effective algorithms deriving actions and business objects from activity labels requires a thorough understanding of current labeling practices. We approached this problem in a bottom-up way by investigating the different 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 organized in 29 functional branches of an enterprise, including sales, accounting, and other functional areas. The SAP Reference Model includes 604 Event-driven Process Chains (EPCs), each containing several activities. Table 1 shows that we found



**Fig. 1.** An example of poor labeling in the SAP Reference Model

**Table 1.** Substyles of action-noun activity labeling style

Name	Structure	Example	Share, %
Noun phrase	<pre>       NP      /  \     NN   NN               bo   a </pre>	Invoice creation	78.8
Noun phrase with <i>of</i> prepositional phrase	<pre>       NP      /  \     NP   PP    /  \ /  \   NN IN  NP             a  'of NN              bo </pre>	Creation of invoice	15.0
Verb phrase (gerund)	<pre>       VP      /  \     VBG  NP                a    NN                  bo </pre>	Creating invoice	5.1
Irregular	-	LIFO: Valuation: Pool level	1.1

four substyles of action-noun labels, each described according to their structure and illustrated by an example.

As it follows from the title, the labels of *noun phrase* style are noun phrases where a business object comes first and an action follows. 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* 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 and can be omitted, as it does not bring useful information regarding action or business object to the reader.

The labels of *Noun phrase with of prepositional phrase* style are noun phrases as well. 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 substyle, 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 optional prepositional phrase in such labels can be ignored within analysis task.

The labels of *verb phrase (gerund)* substyle are verb phrases headed by a gerund. The action is captured as a gerund, succeeded by the business object

captured as a noun. The following labels illustrate exemplars 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 substyles described above cover almost 99% of all action-noun labels in the model collection. However, about 1% of the action-noun 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, linking together parts of the label in a special 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 significant majority of irregular labels can be recognized 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 coordinate 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 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*.

### 3 Automatic Refactoring of Action-noun Activity Labels

This section presents a stepwise approach to automatic refactoring of activity labels from action-noun style into verb-object style. The refactoring process includes the following phases:

1. Label style recognition,
2. Derivation of an action and a business object from the label,
3. Composing a verb-noun label.

#### 3.1 Label Style Recognition

The recognition process is driven by a set of label properties. Each label is evaluated against the set of properties and, according to evaluation results, is categorized into a particular style. Table 2 enumerates the action-noun label substyles and points to their featured properties.

Algorithm 1 formalizes label style recognition. The input of the algorithm is an action-noun label *label*, the output is *prop*—an object storing the label properties with a substyle among them. We assume that all the boolean properties in *prop* object are initiated with *false*.

**Table 2.** Properties of action-noun label substyles

Label class	Property
Noun phrase	none
Noun phrase with <i>of</i> prepositional phrase	label contains a prepositional phrase with <i>of</i> 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., <i>and</i> or <i>or</i>
Irregular	label contains characters '?' or '-'

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*.

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 [18] 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 categorized to none of the enumerated substyles, the algorithm refers it to a *noun phrase* substyle.

### 3.2 Derivation of Action and Business Object

Recognition of a label substyle enables the next step—derivation of an action and a business object. Algorithm 2 presents a derivation method for *noun phrase*, *noun phrase with of prepositional phrase*, and *verb phrase (gerund)* substyles. Labels of irregular substyle are not addressed, as relations between the words

**Algorithm 1** Recognition of action-noun substyles

---

```

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

```

---

in such labels are extremely complex. Algorithm 2 also does not address labels that contain coordinating conjunctions. The input of the algorithm is an action-noun label *label* and a corresponding *LabelProperties* object *prop* storing the properties of *label* obtained with Algorithm 1. The output of the algorithm is *prop* with *action* and *bObject* properties set.

The algorithm starts with an analysis of labels following noun phrase style (lines 4–16). It checks for an optional prepositional phrase. If the label has a prepositional phrase, the phrase is omitted and not studied any more. If the label has only one word, e.g., *Deployment* or *Classification*, this word is recognized as an action. Otherwise, the algorithm checks if the last two words of the label constitute a phrasal verb, for instance, *set up* and *carry forward*. If the first two words are recognized as a phrasal verb, this verb is perceived as an action. The rest of the label, if it exists, is recognized as a business object (lines 11–13). If the phrasal verb is not revealed, the last word is recognized as an action while the rest as a business object (lines 15–16).

Algorithm 2 continues with analysis of *verb phrase (gerund)* labels (lines 17–29). Analysis of these labels resembles the analysis of labels of noun phrase style. The key difference is that the action is expected to appear in the beginning of the label while the business object in the end.

---

**Algorithm 2** Derivation of an action and a business object from a label

---

```

1: deriveActionAndBusinessObject(Label label, LabelProperties prop)
2: if !prop.hasConjunctions then
3:   size = label.words.size;
4:   if prop.style == NOUN then
5:     if prop.hasPrepositions then
6:       label = label.words[1] + ... + label.words[prop.pIndex - 1];
7:     if size == 1 then
8:       action = label.words[1];
9:     else
10:      if label.words[size - 1] + label.words[size] is a phrasal verb then
11:        prop.action = label.words[size - 1] + label.words[size];
12:        if size! = 2 then
13:          prop.bObject = label.words[1] + ... + label.words[size - 2];
14:        else
15:          prop.action = label.words[size];
16:          prop.bObject = label.words[1] + ... + label.words[size - 1];
17:      else if prop.style == GERUND then
18:        if prop.hasPrepositions then
19:          label = label.words[1] + ... + label.words[prop.pIndex - 1];
20:        if size == 1 then
21:          action = label.words[1];
22:        else
23:          if label.words[1] + label.words[2] is a phrasal verb then
24:            prop.action = label.words[1] + label.words[2];
25:            if size! = 2 then
26:              prop.bObject = label.words[3] + ... + label.words[size];
27:            else
28:              prop.action = label.words[1];
29:              prop.bObject = label.words[2] + ... + label.words[size];
30:          else if prop.style == PREPOSITION_OF then
31:            prop.action = label.words[1] + ... + label.words[pIndex - 1];
32:            pPhrase = label.words[pIndex + 1] + ... + label.words[size];
33:            if pPhrase contains prepositions then
34:              nIndex = index of the next preposition after pIndex;
35:              prop.bObject = label.words[pIndex + 1] + ... + label.words[nIndex - 1];
36:            else
37:              prop.bObject = label.words[pIndex + 1] + ... + label.words[size];
38: return prop;

```

---

The algorithm concludes with an analysis of activity labels following *noun phrase with prepositional phrase* style (lines 30–37). The label part preceding preposition *of* is recognized as an action. The label part between preposition *of* and the next preposition is treated as a business object.

Algorithm 2 does not address labels with conjunctions. Therefore, we provide an outlook on how such labels can be analyzed. The first step is to identify, if the conjunction coordinates actions or business objects. This can be achieved using information about the label subtype and the position of the conjunction in the label. Afterwards, an algorithm similar to Algorithm 2 can be applied to derive actions and business objects from coordinated components of the label. Notice that a conjunction may appear in the optional prepositional phrase, e.g., *Creation of proposal for sales and profit planning*. In this case the conjunction is ignored, as it does not coordinate neither actions, nor business objects.

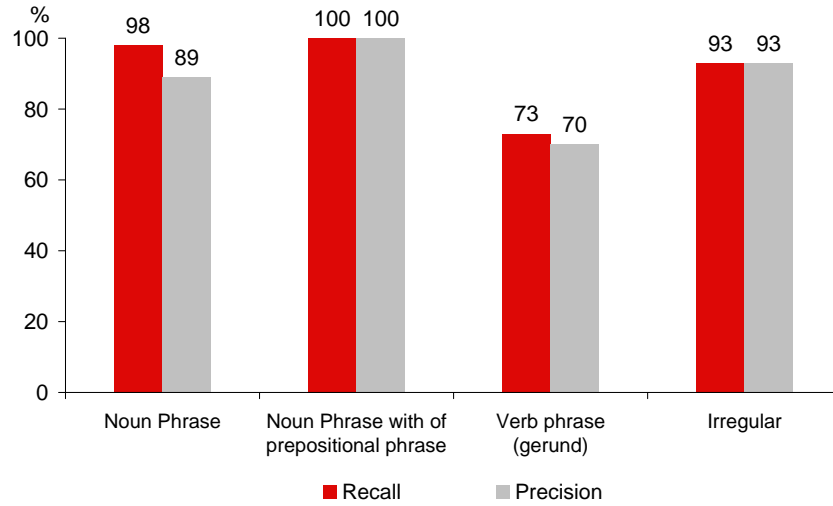
### 3.3 Label Refactoring

Refactoring aims to transform an action-noun label into a verb-object label signifying the same action performed on the same business object. Derivation of actions and business objects from activity labels enables construction of labels in verb-object style. In fact, after the analysis of the previous steps the task becomes a trivial concatenation of a verb representing an action and a noun phrase representing a business object. Notice that the optional prepositional phrase derived from the label at the derivation stage can be preserved in the verb-object label. To achieve this, the prepositional phrase is concatenated to the label after the business object.

As it follows from the title, action-noun labels capture actions with nouns. At the same time, a verb-object label expects a verb to represent the action. To learn which verb corresponds to the noun capturing the action, we make use of *nominalization*—a linguistic phenomenon of producing a noun from another part of speech via the addition of derivational affixes. In the context of this task we are interested in nominalization relations between nouns and verbs. Technically, nominalization relations can be obtained from WordNet. As an example, consider action-noun label *Invoice verification*. The action is given by *verification* and the business object by *invoice*. Using nominalization we can learn that the action signified by the label is *verify*. Concatenation of the verb *verify* and the business object *invoice* results in verb-object label *Verify invoice*.

## 4 Empirical Evaluation

To validate the proposed algorithms, we have conducted an experiment. The goal of the experiment was to learn, how well the proposed algorithms approximate a human interpretation of activity labels. To evaluate the algorithms we have designed a test collection that includes the SAP Reference Model as a process model collection, and human interpretations of activity labels. Human interpretations of activity labels are captured by two mappings: one mapping from an



**Fig. 2.** Precision and recall of algorithms for label style recognition

activity label to a set of corresponding actions and another mapping from an activity to a set of business objects. This information is stored in a spreadsheet, which is read by an application in the evaluation phase.

Within the evaluation we compared:

1. recognition of label substyles by the algorithm and by humans;
2. derivation of actions and business objects by the algorithm and by humans.

To evaluate the substyle recognition algorithm we measured the *precision* and *recall* of the algorithm [2]. The precision of substyle recognition algorithm is the number of correctly recognized labels of the given substyle retrieved by the algorithm divided by the total number of labels retrieved. The recall is the number of correctly recognized labels of the given substyle retrieved by the algorithm divided by the total number of existing labels of this substyle. Fig. 2 presents the values of precision and recall obtained for the SAP Reference Model.

The evaluation of the algorithms for action/business object derivation makes use of precision: the share of correctly derived items among the overall number of derived items. The precision value for action derivation is 88%, for business object derivation 85%, and for label refactoring 85%. As the precision values are reasonably high, we conclude that the proposed algorithms are capable of automatic derivation of actions and business objects from activity labels.

## 5 Related Work

The research reported in this paper relates to two major streams of related work: guidelines for process model labeling and natural language approaches for conceptual models.

The verb-object style is widely promoted in the literature for *labeling activities of process models* [13, 17, 22], but rather as informal guidelines. Similar conventions are advocated as guidelines for the creation of understandable use case descriptions, a widely accepted requirements tool in object-oriented software engineering [20]. Despite the promotion in the process modeling domain, it has been observed that verb-object labeling in real process models is not consistently applied. For instance, the practical guide for process modeling with ARIS [4, pp.66-70] shows models with both actions as verbs and as nouns. Our work helps to automatically refactor such action-noun labels. It also complements prior work on the automatic identification of verb-object labels in [12]. The concept of part of speech tagging is also investigated for interactive process modeling support. In a recent paper, the authors employ it for auto-completion [3].

The enforcement of the verb-object style might help to close the gap between *natural language and formal language processing*. The relationship between process models and natural language has been discussed and utilized in various works. In [6] 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 [10] and to compare process models from a semantic perspective [5]. Most beneficiary is the verb-object style for model verbalization and paraphrasing (see [7, 9]). Such verbalization is an important step in model and requirements validation [19]. For instance, verb-object style labels can easily be verbalized using the *You have to* prefix. In this way, automatic parsing enables a better validation of process models.

## 6 Conclusions and Future Work

In this paper we have proposed a method for automatic refactoring of action-noun activity labels into labels of verb-object style. We performed an evaluation of the proposed approach with a collection of real-world process models from the SAP Reference Model. To enable this evaluation, we have created a test collection that makes use of process models in the SAP Reference Model and human interpretations of activity labels in this collection.

There are two directions of the future work. On the one hand, it is the improvement of the existing refactoring technique and the development of alternative methods based on the employment of natural language processing tools (for instance, [8]). Evaluation and comparison of various refactoring techniques also belongs to this branch of research. As a concrete next step, it is of great interest to evaluate the proposed refactoring technique against other test collections. On the other hand, a number of applications call for algorithms deriving actions and business objects from activity labels including business process compliance verification, matching of process models, and model quality assurance. In particular, we are interested in the integration of action/business object derivation techniques into the automatic modeling support approach presented in [23].

## References

1. W.M.P. van der Aalst. *Business Process Management*, volume 1806 of *LNCS*, chapter Workflow Verification: Finding Control-Flow Errors Using Petri-Net-Based Techniques, pages 161–183. Springer Verlag, 2000.
2. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. May 1999.
3. J. Becker, P. Delfmann, S. Herwig, L. Lis, and A. Stein. Towards Increased Comparability of Conceptual Models - Enforcing Naming Conventions through Domain Thesauri and Linguistic Grammars. In *Proceedings of ECIS*, June 2009.
4. R. Davis. *Business Process Modelling with ARIS: A Practical Guide*. 2001.
5. M. Ehrig, A. Koschmider, and A. Oberweis. Measuring Similarity between Semantic Business Process Models. In *Proceedings of APCCM*, volume 67, pages 71–80, Australia, 2007.
6. G. Fliedl, C. Kop, and H.C. Mayr. From Textual Scenarios to a Conceptual Schema. *Data & Knowledge Engineering*, 55(1):20–37, 2005.
7. P.J.M. Frederiks and T.P. van der Weide. Information Modeling: The Process and the Required Competencies of Its Participants. *Data & Knowledge Engineering*, 58(1):4–20, 2006.
8. The Stanford Natural Language Processing Group. Stanford Log-linear Part-Of-Speech Tagger. <http://nlp.stanford.edu/software/tagger.shtml>.
9. T.A. Halpin and M. Curland. Automated Verbalization for ORM 2. In *OTM*, volume 4278 of *LNCS*, pages 1181–1190. Springer, 2006.
10. J.E. Ingvaldsen, J. A. Gulla, X. Su, and H. Rønneberg. A Text Mining Approach to Integrating Business Process Models and Governing Documents. In *OTM*, volume 3762 of *LNCS*, pages 473–484. Springer, 2005.
11. G. Keller and T. Teufel. *SAP(R) R/3 Process Oriented Implementation: Iterative Process Prototyping*. Addison-Wesley, 1998.
12. H. Leopold, S. Smirnov, and J. Mendling. On Labeling Quality in Business Process Models. In *EPK*, volume 554 of *CEUR*, pages 42–57, 2009.
13. Th. W. Malone, K. Crowston, and G. A. Herman. *Organizing Business Knowledge: The MIT Process Handbook*. The MIT Press, 2003.
14. J. Mendling. *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness*, volume 6 of *LNBIP*. 2008.
15. J. Mendling, H. A. Reijers, and J. Cardoso. What Makes Process Models Understandable? *Business Process Management*, pages 48–63, 2007.
16. J. Mendling, H. A. Reijers, and J. Recker. Activity Labeling in Process Modeling: Empirical Insights and Recommendations. *Information Systems 2009*.
17. L. D. Miles. *Techniques of Value Analysis and Engineering*. McGraw-hill, 1961.
18. G. A. Miller. WordNet: a Lexical Database for English. *Commun. ACM*, 38(11):39–41, 1995.
19. B. Nuseibeh and S.M. Easterbrook. Requirements Engineering: a Roadmap. *22nd International Conference on Software Engineering* pages 35–46, 2000.
20. K. Th. Phalp, J. Vincent, and K. Cox. Improving the Quality of Use Case Descriptions: Empirical Assessment of Writing Guidelines. *Software Quality Journal*, 15(4):383–399, 2007.
21. M. Rosemann. Potential Pitfalls of Process Modeling: Part A. *Business Process Management Journal*, 12(2):249–254, 2006.
22. A. Sharp and P. McDermott. *Workflow Modeling: Tools for Process Improvement and Applications Development*. Artech House Publishers, 2008.
23. S. Smirnov, M. Weidlich, J. Mendling, and M. Weske. Action Patterns in Business Process Models. In *ICSOC*, Stockholm, November 2009.