On the Automatic Labeling of Process Models

Henrik Leopold[1], Jan Mendling[1], and Hajo A. Reijers[2]

[1] Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany
{henrik.leopold,jan.mendling}@wiwi.hu-berlin.de
[2] Eindhoven University of Technology
PO Box 513, 5600 MB Eindhoven, The Netherlands
h.a.reijers@tue.nl

Abstract. Process models are essential tools for managing, understanding and changing business processes. Yet, from a user perspective they can quickly become too complex to deal with. Abstraction – aggregating detailed fragments into more coarse-grained ones – has proven to be a valuable technique to simplify the view on a process model. Various techniques that automate the decision of which model fragments to aggregate have been defined and validated by recent research, but their application is hampered by the lack of abilities to generate meaningful names for such aggregated parts. In this paper, we address this problem by investigating naming strategies for individual model fragments and process models as a whole. Our contribution is an automatic naming approach that builds on the linguistic analysis of process models from industry.

1 Introduction

Business process management is a concept for enabling companies to cope with the increasing dynamics and challenges in a competitive business environment. A key element of process management is to map business processes to models in order to leverage understanding, analysis and improvement of processes. Today, many larger enterprises possess an extensive documentation of their business process in terms of several thousand models, often at a significant level of detail [1]. In order to make large and detailed models easier to understand, recent research has developed automatic abstraction techniques to generate coarse-grained model parts from more detailed ones [2,3].

The essential idea of abstraction is to identify fragments of a model that can be aggregated into a single activity. While this is valuable to reduce the structural complexity of a large model, existing techniques do not address how a suitable name for an aggregated part can be established. When using abstraction to render a high-level view of a process model for a human reader, which is the most popular use case for abstraction [4], this is troublesome. In this paper, we address the naming problem of aggregated model parts from the perspective of naming *a whole process model*. A complete process model is as much a collection of activities with mutual control-flow dependencies as an aggregated process

H. Mouratidis and C. Rolland (Eds.): CAiSE 2011, LNCS 6741, pp. 512–520, 2011.

model part is, although it is evidently not part of any higher-level model. Since in many industrial settings entire process models *themselves* carry names that convey indications of the business procedure that they capture, the underlying process model naming conventions are a valuable source of inspiration on how to name model parts. Our contribution is an automatic approach for generating name suggestions for a process model based on its events and activities, which is applicable to process model fragments as well. From a practical point of view, this approach paves the way to an integrated and automated abstraction of process models, in pursuit of the communication advantages associated with skilfully decomposed process models [5].

Against this background, the paper is structured as follows. Section 2 discusses the problem of assigning a meaningful name to a process and identifies a list of naming strategies in models from practice. Section 3 describes the different phases of our approach to generate process model names. Section 4 discusses our contribution in the light of related work. Finally, Section 5 summarizes the findings and provides an outlook on future research.

2 Naming of Process Models in Practice

Before considering an automatic approach for generating process model names, we have to understand how modelers assign names to process models. Guidelines exist on how activity names should be composed, e.g. [6] suggest a verb-object style putting the action first followed by the corresponding business object. While such guidelines advocate a certain *grammatical* structure of naming, they do not deal with its *content* by refraining to mention how to choose a particular verb or business object in the name of the model. In Section 2.1, we introduce Event-Driven Process Chains (EPCs), the process modeling language that we consider in this paper, and discuss directions for choosing a name for a model. In Section 2.2, we inspect three sets of process models from practice in order to identify strategies of naming. These strategies provide us with the foundation to automatically generate names for a fragment or the whole EPC.

2.1 Event-Driven Process Chains

An EPC is a graph-structured process model, which consists of different types of nodes: functions, events, and connectors. Functions define the business activities that have to be executed while events define the pre-conditions and post-conditions for starting a function. Figure 1 shows an example EPC from the SAP reference model with two functions (rounded boxes) and four events (hexagons). Functions and events are connected via control flow arcs in an alternating way. Complex routing is defined using connectors (circles). In the example, we observe an OR-split connector (symbol $\vee$) creating two end events. An EPC has at least one start event (no incoming arc) and at least one end event (no outgoing arc).

To illustrate the naming problem addressed in this paper, we re-consider the example EPC from Figure 1. One approach for naming the entire process would be to consider the different activities of the model. The two functions relate

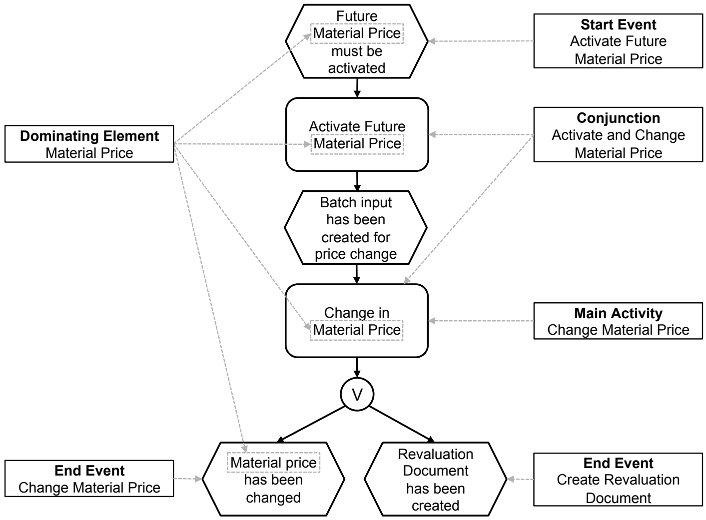

Fig. 1. Potential sources of process names for model *Change in Material Price*

to the business object *material price*, in respective connection with the actions *activate* and *change*, which therefore could be used as central elements of the overall name. Another option would be to look at the names of the end events since they refer to what is actually accomplished by executing the process. Hence, the *creation* of the *revaluation document* may represent another naming option. This discussion aims to emphasize that there are many options, and they are difficult to rank without prior domain knowledge. In the following part, we aim to systematically identify different strategies for choosing a name for a process.

2.2 Classification of Naming Strategies

We collected an extensive set of process models from practice in order to identify naming strategies. We used three different model collections, aiming in this respect for broad diversity in the underlying domains. First, we had access to the *SAP Reference Model* [7, pp. 145-164], a collection containing 604 EPCs organized in 29 functional branches of an enterprise such as sales, accounting and other areas, with a total number of 2,433 activity labels. Second, we have used a model collection from a *European utility vehicle manufacturer* consisting of one main procurement process with nine sub-processes and altogether 115 activity labels. Third, we inspected a model collection containing the incident management process from an *international IT service provider*. The process is captured as an EPC on three abstraction layers, containing 88 sub-processes and 293 activity labels.

We analyzed the names in the model sets by first identifying action and business objects in the name of the process model. Then, we used a self-developed tool that identified linguistic relations between the name parts and the activity and event labels. As a final step, we inspected each model manually. Based on this procedure, we were able to develop a classification of naming strategies. Five different approaches were observed including Dominating Element, Main Activity, Start or End Events, Conjunction of Activities, and Semantic Naming. Figure 1 illustrates some of these strategies, which we now aim to describe.

1. **Dominating Element:** If one particular business object or action is mentioned more often in activity labels than any other business object or action, this element is considered to be a dominating element. In the analyzed process model collections dominating elements were often used for naming the process model if such a dominating element existed. The example in Figure 1 has *Material Price* as such a dominating element.
2. **Main Activity:** Some of the analyzed processes contain one particular activity that is of central importance. The remaining activities have the character of side activities supporting, preparing or evaluating the result of this activity. Figure 1 is a good example since *Change in Material Price* has such characteristics. The process also contains an activity which is concerned with activating the future material price. However, from the choice of the modeler we can assume that this activity only plays a subordinate role, while the focus is on the activity of changing the material price.
3. **Start or End Events:** Especially when the state at the beginning or at the end of process define the overall goal of the process, the name of the whole model may be closely related to them. In Figure 1 this is visible in the start event *Future Material Price must be activated* and the end event *Material price has been changed.*
4. **Conjunction of Activities:** If the same action is performed on different business objects or different actions are applied on the same business object, these activities can be easily described in terms of a conjunction. Even whole activity labels may be connected if the resulting name is not too complex. For Figure 1, this would yield *Activate and Change Material Price.*
5. **Semantic Naming:** The previously introduced concepts always explicitly refer to the textual description of at least one element in the process model. By contrast, the concept of semantic naming does not refer to one or more model elements, but uses the broader context of the activities for naming the process model. This can be appropriate if there is a part-of relationship between the activities and the name of the process [8]. Hence, the process name, which is itself representing an activity, subsumes the given activities in the model. As an example, consider the SAP Process *Shipping.* It consists of five events and the two activities *Delivery for Returns* and *Goods Receipt Processing for Returns.* Apparently, the action shipping is not mentioned in any of the process elements. Nevertheless, *shipping* can be considered as a more general concept in semantic terms, which implies delivery and goods receipt. Clearly, the derivation of semantic names requires external knowledge, e.g. in terms of an ontology, and cannot be directly derived from the activity names.

3 An Automatic Approach to Generate Process Names

In this section we present an automatic approach for the generation of process model names, which builds on the strategies identified in the previous section. The main idea of the approach is to derive a set of potentially useful names for a given process model based on its activities and events. Subsequently, a modeler can select the most suitable name.

We organize our approach in three steps. Phase 1 serves as preparation for the main information extraction. In this phase all activities, start events and end events of the given process model are annotated with their respective action and business object. For this step, we use an algorithm for automatically identifying action and business objects from activity labels as presented in [9], and extended it with a capability to analyze different start and end event structures as defined in [10]. In Phase 3, the name candidates are transformed automatically to the verb-object style based on the techniques defined in [9]. In the following subsections, we introduce the specific techniques to generate the different proposals in Phase 2, as well as their interdependence. All of these assume that annotations of actions and business objects are available from Phase 1. The reader may wish to refer to [11] for the pseudo algorithms that abstract from our implementation in Java.

The *Dominating Element Extraction* technique investigates whether the given process model includes a dominating action and dominating business object. Therefore, for each element type, i.e. action or business object, the occurrence of the elements among all activities of the model is checked. If there exists an element that has a higher occurrence than all other elements among this type, it is saved as dominating element. If a dominating element has been identified, it can be used as input for the *Subordinate Element Extraction* technique. In case no dominating element could be detected, the further steps are limited to executing the *Event Extraction* and the *Main Activity Extraction* techniques, which do not require the input of a dominating element.

If one type of dominant element was detected, the *Subordinate Element Extraction* technique identifies those actions or business objects with which the dominant element is connected in the given process model. Therefore, all activities containing the dominating element are scanned and the complementing element is both extracted and saved to a list of subordinate elements. If, for instance, the dominating action *analyze* was derived from the two activities *Order Analysis* and *Program Analysis*, the subordinate elements are given by the business objects *order* and *program*. Hence, all activities containing the dominating element are selected and the subordinate elements are derived.

We introduce two techniques for constructing a process name based on the set of subordinate elements and the dominating element: *Lexical Conjunction* and *Logical Conjunction*. In case of the Lexical Conjunction the subordinate actions or business objects are replaced with a newly introduced element. In particular, the lexical database WordNet is consulted to detect common *holonyms* and *hypernyms* of the subordinate elements. If a proper holonym – a word that is more generic than a given word – or a hypernym is found – a word that is more specific than a given word – a name proposal is constructed using the dominating element and the according holonym or hypernym. In case of the Logical Conjunction, the subordinate elements are simply connected using the logical operators *and* or *or*.

Another technique that builds on the identified dominating elements is *Label Repository*. This technique uses the activities of other process models to build up a label repository. If a dominating element was identified with the *Dominating Element Extraction* technique, such a label repository can be consulted to find a corresponding element which is likely to be connected with the dominating element. As an example, consider the SAP process model *Capacity Planning* containing the dominating business object *Capacity*. As this business object in isolation is not a very comprehensive process name, the repository can be consulted. By browsing the repository for labels containing *Capacity*, we obtain, amongst others, the process name *Capacity Planning* which perfectly matches the original process name.

The *Event Extraction* technique derives potentially useful names from start and end events. Therefore, start and end events are inspected on their merit to provide information about the model content. That decision is based on the usage of particular signal terms given in the event label. For instance, it is not very probable that the term *was* in the start event *Asset was found* indicates what is *to be* performed in the process; rather, it represents a state that was required for triggering the first activity. By contrast, the term *is to be* in the start event *Asset is to be created* captures the necessity for the execution of a *subsequent* action within the process of consideration. Hence, (1) the identified start events are reduced to those where the signal term indicates that the event actually contains information about what is *going* to happen and (2) the end events are restricted to those that signal what *has* happened in the process. Based on an extensive classification of these terms from the investigated process model collections, this decision can be made in an automated fashion.

Referring to the main activity approach as briefly mentioned in Section 2.2, we further introduce the *Main Activity Extraction* technique. The objective of this technique is to automatically decide whether a considered activity represents a main activity for the given process model or not. In order to be able to make this decision for an individual activity, it is necessary to automatically derive the context of the process and subsequently decide about the role of the activity. This approach utilizes the insight of our analysis that approximately 85% of the main activities are found either at the beginning or at the end of the process. Accordingly, the main activity extraction presumes the existence of a main activity in the first or last position and selects the according activity labels as process name proposals.

In order to obtain an all-encompassing approach, we combined all techniques. To some extent the order of executing the techniques is fixed as some depend on the on the input of other, like the *Lexical Conjunction*. However, techniques such as *Main Activity Extraction* can be executed independently from other techniques and can be executed at any time.

4 Related Work

Several approaches have been proposed for process model abstraction. The work by Polyvyanyy et al. builds on an algorithm for aggregating activities based on a slider and specific abstraction criteria [3]. Abstraction criteria are discussed in [12,13]. A recent paper presents an abstraction approach based on behavioral profiles [14]. For a set of activities, this approach generates the control flow of the aggregated model. Both approaches do not generate names for aggregated activities, such that our work is complementary. A different approach based on meronymy relations is presented in [15]. This approach inspects meronymy relations between activity labels to find aggregation candidates. It integrates the problems of finding aggregation candidates and aggregation names. Our work is more general in that sense that it is able to derive names for arbitrary process model fragments, even if they do not share a meronymy relation.

The linguistic analysis of activity labels is also an import task in process model matching and similarity calculation [16,17,18]. Different approaches of matching process models are integrated in [19]. This area is also related to research on semantic annotation of business process models [20]. Recent research has also started using natural language processing techniques for generating process models from text. Gonçalves et al. generate process models from group stories [21]. The approach by the University of Klagenfurt combines linguistic analysis with user feedback [22]. The Rapid Business Process Discovery (R-BPD) framework uses natural language techniques for constructing BPMN models from corporate documentations or web-content [23]. Anaphora resolution is tackled in a recent approach to generate BPMN models [24].

5 Conclusion

In this paper, we have addressed the problem of automatically generating names for process models. Our work is motivated by the fact that existing works on process model abstraction require telling names for structurally aggregated process fragments. Our overall contribution is an automatic naming approach that builds on the linguistic analysis of the elements of process models from industry. The work presented in this paper has significant implications for research and practice. The automatic generation provides the basis not only for proposing names of whole processes, but also for process fragments. In this regard, our approach can be used for instance to dynamically generate abstractions of different granularity as the user is interacting with the modeling tool.

The main task for future research is the validation of the presented approach. This may include the comparison of the given with the generated names but also an applicability assessment by humans. In addition, we aim to further investigate the usability of different naming strategies. Currently, if a single name for an abstracted fragment is needed, a system can only make a random suggestion from the set of name proposals. We expect that the strategy itself, but also the length of the suggested name has a significant impact on the perceived usefulness. Based on such insight, we will be able to select the best name from a set of suggestions.

References

1. Rosemann, M.: Potential pitfalls of process modeling: part a. Business Process Management Journal 12(2), 249–254 (2006)
2. Eshuis, R., Grefen, P.: Constructing customized process views. Data Knowl. Eng. 64(2), 419–438 (2008)
3. Polyvyanyy, A., Smirnov, S., Weske, M.: Process model abstraction: A slider approach. In: Proceedings of EDOC (2008)
4. Smirnov, S., Reijers, H.A., Nugteren, T., Weske, M.: Business Process Model Abstraction: Theory and Practice. Technical Report 35 (2010)
5. Reijers, H.A., Mendling, J.: Modularity in process models: Review and effects. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 20–35. Springer, Heidelberg (2008)
6. Mendling, J., Reijers, H.A., Recker, J.: Activity labeling in process modeling: Empirical insights and recommendations. Inf. Syst. 35(4), 467–482 (2010)
7. Keller, G., Teufel, T.: Sap R/3 Process Oriented Implementation. Addison-Wesley Longman Publishing Co., Inc., Boston (1998)
8. Reijers, H., Limam, S., van der Aalst, W.: Product-based workflow design. Journal of Management Information Systems 20(1), 229–262 (2003)
9. Leopold, H., Smirnov, S., Mendling, J.: Refactoring of process model activity labels. In: Hopfe, C.J., Rezgui, Y., Métais, E., Preece, A., Li, H. (eds.) NLDB 2010. LNCS, vol. 6177, pp. 268–276. Springer, Heidelberg (2010)
10. Decker, G., Mendling, J.: Process instantiation. Data Knowl. Eng. 68(9), 777–792 (2009)
11. Leopold, H.: Modularization of business process models using natural language techniques. Master's thesis, Humboldt-Universität zu Berlin (2010)
12. Eshuis, R., Grefen, P.: Constructing Customized Process Views. Data Knowl. Eng. 64(2), 419–438 (2008)
13. Günther, C.W., van der Aalst, W.M.P.: Fuzzy Mining – Adaptive Process Simplification Based on Multi-perspective Metrics. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 328–343. Springer, Heidelberg (2007)
14. Smirnov, S., Weidlich, M., Mendling, J.: Business process model abstraction based on behavioral profiles. In: Maglio, P.P., Weske, M., Yang, J., Fantinato, M. (eds.) ICSOC 2010. LNCS, vol. 6470, pp. 1–16. Springer, Heidelberg (2010)
15. Smirnov, S., Dijkman, R., Mendling, J., Weske, M.: Meronymy-based aggregation of activities in business process models. In: Parsons, J., Saeki, M., Shoval, P., Woo, C., Wand, Y. (eds.) ER 2010. LNCS, vol. 6412, pp. 1–14. Springer, Heidelberg (2010)
16. Ehrig, M., Koschmider, A., Oberweis, A.: Measuring similarity between semantic business process models. In: Proceedings of APCCM, pp. 71–80 (2007)
17. van Dongen, B.F., Dijkman, R., Mendling, J.: Measuring similarity between business process models. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 450–464. Springer, Heidelberg (2008)
18. Dijkman, R., Dumas, M., van Dongen, B., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. Inf. Systems 36, 498–516 (2010)
19. Weidlich, M., Dijkman, R., Mendling, J.: The iCoP framework: Identification of correspondences between process models. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 483–498. Springer, Heidelberg (2010)
20. Lin, Y., Ding, H.: Ontology-based semantic annotation for semantic interoperability of process models. In: Proc. of CIMCA/IAWTIC, pp. 162–167 (2005)

21. de AR Gonçalves, J.C., Santoro, F.M., Baiao, F.A.: A case study on designing processes based on collaborative and mining approaches. In: Int. Conf. CSCWD (2010)
22. Kop, C., Vöhringer, J., Hölbling, M., Horn, T., Mayr, H.C., Irrasch, C.: Tool supported extraction of behavior models. In: ISTA, pp. 114–123 (2005)
23. Ghose, A.K., Koliadis, G., Chueng, A.: Rapid business process discovery (R-BPD). In: Parent, C., Schewe, K.-D., Storey, V.C., Thalheim, B. (eds.) ER 2007. LNCS, vol. 4801, pp. 391–406. Springer, Heidelberg (2007)
24. Friedich, F., Mendling, J., Puhlmann, F.: Process model generation from natural language text. In: Proc. of CAISE. LNCS (2011)