Enabling Reuse of Process Models through the Detection of Similar Process Parts

Fabian Pittke[1,3], Henrik Leopold[1], Jan Mendling[2], and Gerrit Tamm[3]

[1] Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany
henrik.leopold@wiwi.hu-berlin.de
[2] WU Vienna, Augasse 2-6, A-1090 Vienna, Austria
jan.mendling@wu.ac.at
[3] SRH University Berlin, Ernst-Reuter-Platz 10, 10587 Berlin, Germany
fabian.pittke@srh-uni-berlin.de, gerrit.tamm@srh-uni-berlin.de

Abstract. Many companies use business process modeling to support various improvements initiatives leading to an increasing number of process models. Typically, these models are stored in a collection containing several hundreds of process models. In many cases, process models are overlapping, although parts could be easily reused saving costs and efforts. Different labeling styles and evolving process models complicate the detection of reusable model parts. In this paper, we propose a novel approach for the detection of equivalent and similar process model parts that exploits semantic comparison of activity labels and behavioral comparison of control flow. We evaluate our approach on the SAP Reference Model, a collection with 604 process models. The evaluation reveals insights for the thresholds of semantic and behavioral similarity of process models as well as their influence for similar process part detection. Hence, we identify five candidate groups with specific similarity properties that contain reoccurring process parts.

Keywords: Business Process Modeling, Similar Process Part Detection, Semantic Similarity, Behavioral Similarity.

1 Introduction

Due to the increasing popularity of business process modeling many companies face a steadily increasing amount of process models. In some cases, such process model collections range up to thousands of process models [1]. As a result, these companies struggle with the effective maintenance of their process model collections [2].

A corresponding problem of growing model collections is the increasing overlap across process models. Hence, the implementation of consistent changes becomes more and more challenging and cost intensive. Moreover, size and number of overlapping process models is often larger than it necessarily had to be. Reoccurring parts could be easily extracted in form of separate process models that reduce the complexity of the models themselves and of the overall collection. Recent

M. La Rosa and P. Soffer (Eds.): BPM 2012 Workshops, LNBIP 132, pp. 586–597, 2013.

research addressed this problem by detecting clones in process model repositories [3]. Although the detection of clones is undoubtedly a very useful step, it is not complete. Moreover, most approaches only focus on structural aspects and disregard semantic aspects. As a consequence, semantically equal process model parts which are not exact matches remain undetected and impede reuse.

There are at least two issues that complicate non-exact reuse. First, the labeling of process model elements in practice is heterogeneous and modelers use different labeling patterns to express the same semantics [4]. Thus, equivalent process parts with differently labeled elements would not be recognized as the same. Second, minor changes, for instance the insertion of an additional activity or the usage of other words, impede the identification of similar process parts. In this paper, we address this problem by introducing an approach for the identification of semantically equivalent and similar process parts in process model repositories. Our approach exploits the semantic comparison of activity labels and the behavioral comparison of control flow aspects. To demonstrate the applicability of our approach, we conduct an evaluation with the SAP Reference Model. We also provide insights into the sensitivity of the similarity thresholds for semantic and behavioral similarity and their influence for process part detection. While our approach detects clones for high similarity thresholds, it still detects meaningful candidates, when loosen the clone requirement.

The remainder of this paper is structured as follows. Section 2 illustrates the motivation of our work. Section 3 defines our approach for detecting similar process model parts. Section 4 presents the results from our empirical evaluation with the SAP Reference Model. Section 5 discusses related work before Section 6 concludes the paper and gives an outlook on future research.

2 Problem Illustration

The main challenge associated with the identification of similar process parts is given by appropriately covering semantic aspects. While the identification of clones can be accomplished on a structural level [3], this is not possible for process parts deviating in labeling style and behavior.

Figure 1 shows a typical example of two similar process models from the SAP Reference Model. Thereby, functions with a bold line represent activities which are not covered by the other process. Bold font indicates that the function has a corresponding function in the other model, but captures the semantics in a linguistically different way. Considering the models, we observe that both have additional functions and that two functions differ in the label. However, it is also obvious the these models are semantically very close. For instance the label *Processing of Shipping Notifications / Confirmations* gives the same instruction as the label *Shipping Notifications / Confirmation Processing*. In this case the two functions simply make use of different label styles resulting in a different position of the action *to process* [4]. While this is a rather syntactical difference, the labels *Scheduling Agreement Delivery Schedule* and *Schedule Line (Schedule Agreement)* represent a semantically more complex example.

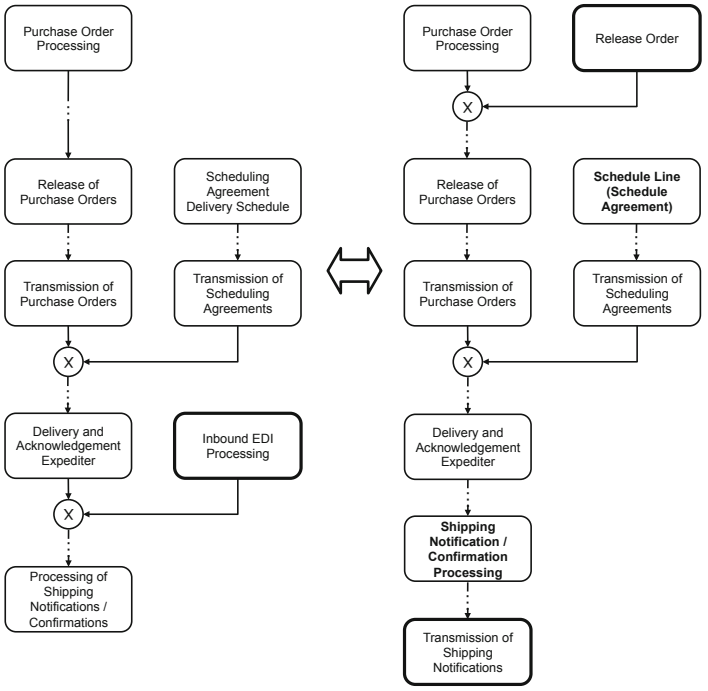

Fig. 1. Almost Completely Similar Process Parts from the SAP Reference Model

The comparison of process models is a widely discussed area and many metrics have been defined. Thereby, many metrics build on the process model structure [5,6] or the execution semantics [7,8]. However, the calculation of a semantic similarity of process model labels which goes beyond the consideration of synonyms has not been addressed so far. However, because of the potential of similar process parts to increase the reuse in process model collections, we consider this to be an important step.

3 Approach for the Detection of Similar Process Parts

This section introduces our three-step approach for the detection of similar process parts. It includes (1) activity label annotation, (2) semantic label similarity calculation, and (3) behavioral similarity calculation.

3.1 Activity Label Annotation

In order to accomplish a comparison which goes beyond a simple string comparison we annotate each activity with its semantic components. As pointed out by [9] each activity can be characterized by three components: an action, a business object, on which the action is performed, and an additional information

fragment that provides further details if required. As an example, consider the activity *Send Contract to Customer* which contains the action *to send*, the business object *contract* and the additional information fragment *to customer*. To reliably accomplish the annotation of these components, we employ a technique defined in prior work [4]. The introduced technique builds on the observation that activities follow regular patterns, so called label styles. The most prominent label style is the verb-object style where the action is captured as an infinitive verb at the beginning of the label. Examples are *Notify Customer* or *Print Document*. However, many labels follow actually other styles such as the action-noun style. In these cases the action is not given as a verb but as a noun at the end of the label. Examples are *Order Verification* or *Product Shipment*. The given examples demonstrate that knowledge about label styles can be used to annotate activities. Once the label style is identified, the derivation of the components is a straightforward step.

3.2 Semantic Label Similarity Calculation

The calculation of the semantic process similarity builds on the annotation of the comprised activities. The fundamental idea is to first calculate the semantic similarity between all activities of the input models and then compute an overall similarity score.

The calculation of the semantic similarity between two activities is accomplished as follows. As a result of the annotation, we can use the semantic components of two given activity labels to compute their semantic closeness. Hence, we consider the similarity between the actions, the business objects and the additional fragments. Thereby, the similarity between two components is given by the closeness of the two concepts in the WordNet taxonomy [10]. This closeness can be calculated using a metric defined by Lin [11].

For calculating this semantic similarity between two activity labels l_1 and l_2, we introduce three functions: a component similarity function sim_c, a coverage function cov, and a label similarity function sim_l combining the latter two to an overall result.

The function sim_c calculates the semantic closeness of two label components l_{c_1} and l_{c_2}. In general, the result of the Lin measurement is returned. If only one label includes the considered component, the value is set to zero.

$$sim_c(l_1, l_2) = \begin{cases} 0 & \text{if } l_{1_c} = \emptyset \vee l_{2_c} = \emptyset \\ Lin(l_{1_c}, l_{2_c}) & \text{if } l_{1_c} \neq \emptyset \wedge l_{2_c} \neq \emptyset \end{cases} \tag{1}$$

The coverage function cov is used to determine the number of components in a label l. Assuming that a label always refers to an action, the result of cov ranges from 1 to 3. Note that the index a in the definition denotes the action, bo the business object and add the additional information fragment.

$$cov(l) = \begin{cases} 1 & \text{if } l_a \neq \emptyset \wedge l_{bo} = \emptyset \wedge l_{add} = \emptyset \\ 2 & \text{if } l_a \neq \emptyset \wedge (l_{bo} \neq \emptyset \vee l_{add} \neq \emptyset) \\ 3 & \text{if } l_a \neq \emptyset \wedge l_{bo} \neq \emptyset \wedge l_{add} \neq \emptyset \end{cases} \tag{2}$$

In order to calculate an overall result from the individual similarity results, we introduce the function sim_l. It calculates the arithmetic mean of the similarity values for action, business object and the additional information fragment. This is accomplished by dividing the sum of sim_a, sim_{bo} and sim_{add} by the maximum coverage among l_1 and l_2. As a result, we obtain the overall similarity score for two given labels.

$$sim_l(l_1, l_2) = \frac{sim_a(l_1, l_2) + sim_{bo}(l_1, l_2) + sim_{add}(l_1, l_2)}{\underset{l \in \{l_1, l_2\}}{\arg\max}\, cov(l)} \tag{3}$$

In order to calculate the similarity for the complete process models, we adapt a metric proposed by [5]. By calculating sim_l for all activity pairs which can be derived from the input models we can identify the best activity pairs based on their sim_l value. Accordingly, we use the relation of these pairs and the overall number of activities in both models to yield an overall process similarity score. Let $M^{opt}_{sim_l}$ be an optimal equivalence mapping derived from sim_l. Further A represents the set of activities in a given process models P. As a result, we can define the process model similarity metric sim_p models as follows:

$$sim_p(p_1, p_2) = \frac{2 \cdot \sum_{(l_1, l_2) \in M^{opt}_{sim_l}} sim_l(l_1, l_2)}{|A_1| + |A_2|} \tag{4}$$

3.3 Behavioral Similarity Calculation

Besides containing similarly labeled activities process parts also require these activities to occur in a similar order. To measure such a control-flow oriented similarity of two process models, we use the concept of behavioral profiles [12] and the respective behavioral metric [8].

Behavioral profiles are an abstract representation of control flow aspects. They capture behavioral characteristics by describing the relation between pairs of activities. The possible relations are grounded in the weak order relation between two activities, which holds when there exists a path from activity x to activity y, denoted with $x \succ_P y$. With this background, behavioral profiles can be defined as follows. Let A be the set of all activities of a given process model P. Each pair $(x, y) \in (A \times A)$ has one of the following relations:

- strict order relation $\rightsquigarrow_P$, *iff* $x \succ_P y$ *and* $y \not\succ_P x$.
- exclusiveness relation $+_P$, *iff* $x \not\succ_P y$ *and* $y \not\succ_P x$.
- interleaving order relation $||_P$, *iff* $x \succ_P y$ *and* $y \succ_P x$.

Thus, a behavioral profile B_P of a process model P is defined as $B_P = \{\rightsquigarrow_P, +_P, ||_P\}$. Also note, that for each pair (x, y) in strict order relation also fulfills the inverse strict order relation for (y, x), i.e. $x \rightsquigarrow_P y \Leftrightarrow y \rightsquigarrow_P^{-1} x$.

The behavioral profile metric is based upon these basic relations of a behavioral profile. Let B_P and B_Q be two behavioral profiles of the process models P and Q. Hence, the behavioral similarity is defined as,

Algorithm 1. Checking similarity of two process models with given thresholds

```
1: isSimilar(ProcessModel m1, ProcessModel m2, float thresholdSimP,
   float thresholdSimBP)
2: similar = false;
3: sim_p = 0;
4: List activityPairs = new List();
5: for i = 1 to m1.getActivities().getLength() do
6:    currentActivity = m1.getActivities().getItem(i);
7:    tempActivity = null;
8:    for j = 1 to m2.getActivities().getLength() do
9:       maxSim = 0;
10:      sim = sim_l(currentActivity, m2.getActivities().getItem(j));
11:      if sim > maxSim then
12:         maxSim = sim;
13:         tempActivity = m2.getActivities().getItem(j);
14:   if maxSim > 0 then
15:      sim_p = sim_p + maxSim;
16:      activityPairs.add(currentActivity, tempActivity);
17:      cleanLists(currentActivity, tempActivity);
18:   else
19:      cleanLists(currentActivity);
20: sim_p = sim_p/(m1.getActivities().size() + m2.getActivities().size());
21: sim_bp = getBehavioralSimilarity(activityPairs, m1, m2);
22: if sim_p ≥ thresholdSimP then
23:    if sim_bp ≥ thresholdSimBP then
24:       similar = true;
25: return similar;
```

$$sim_{bp}(B_P, B_Q) = 1 - \sum_h w_h \cdot sim_h(B_P, B_Q) \tag{5}$$

with $h \in \{+, \rightsquigarrow, ||, \rightsquigarrow', ||'\}$ *and weighting factors* $w_h \in \mathbb{R}$, $0 < w_h < 1$ *such that* $\sum_h w_h = 1$.

sim_h refers to the elementary behavioral similarity metrics introduced in [8].

3.4 Approach for Detecting Similar Process Parts

Our approach builds upon the three steps described above. Activity label annotation is performed for each activity label revealing action, business object and the additional fragment. Afterwards, all available process models are compared with each other resulting in a similarity score of sim_p. In this step, we additionally identify activities that form a semantic pair. Using these activity pairs we can determine the behavioral similarity of the two process models. If the model pair fulfills certain thresholds for sim_p and sim_{bp}, it is considered to be similar. The approach is formalized in Algorithm 1 taking two process models $m1$ and

$m2$ as well as thresholds for sim_p and sim_{bp} as input and returning a boolean score reflecting the similarity and the consistency of the detected process part.

The algorithm starts with some basic initializations (line 2-3). Afterwards, the algorithm processes all possible pairs of activities from $m1$ and $m2$ (line 4, line 7) as follows. The similarity scores between $currentActivity$ from $m1$ and the activities from $m2$ are calculated. If the similarity of $currentActivity$ and the respective activity from $m2$ is higher than the highest similarity score ($maxSim$) calculated so far, $maxSim$ is updated by the new similarity score and the respective activity from $m2$ temporarily stored (lines 10-13). These steps are repeated for all activities from $m2$. Afterwards, the algorithm checks, whether the maximal calculated similarity is bigger than zero (line 13). If this is the case, the similarity of the input models is increased by the respective similarity (line 15) and the activity pair contributing the similarity score is added to $activityPairs$, a list that contains all pairs of activities with the highest pairwise similarity score (line 16). Additionally, the activity lists of $m1$ and $m2$ are cleaned from the pair activities to prevent one activity occurring in multiple activity pairs (line 17 and line 19). After the calculation of the final model similarity score (line 20), the algorithm proceeds with the calculation of the behavioral similarity using the two process models $m1$ and $m2$ as well as the list $activityPairs$ as input (line 21). If sim_p and afterwards sim_{bp} exceed the two thresholds, two boolean variable $similar$ is set to $true$ (lines 22-24). The algorithm terminates with the output of $similar$ indicating a similarity or not (line 25).

4 Evaluation

In this section we evaluate the introduced approach for detecting similar process parts.To this end, we test our technique on the SAP Reference Model, a model collection containing 604 Event-Driven Process Chains [13].The comprised process models are organized in 29 functional branches, as for instance procurement, sales or financial accounting. In Section 4.1 we present the general results from our test run. In Section 4.2 we discuss the relation between sim_p and sim_{bp} as well as their influence for the detection of process model parts.

4.1 General Results

We conduct a pair-wise comparison of all models in the SAP Reference Model to identify similar processes. Accordingly, we conducted in total $\binom{604}{2} = 182.106$ comparisons and computed sim_p and sim_{bp} for each model pair in the collection. The results are depicted in Figure 2. It illustrates the number of retrieved process model pairs depending on varying thresholds for sim_p. Apparently, the more we decrease the threshold for the sim_p, the more process model pairs are identified by the algorithm.

We also computed the average sim_{bp} values for the respective sim_p as illustrated in Figure 3. We observe a proportional relation between the decrease of sim_p and the average sim_{bp} value, i.e. sim_{bp} is dependent on sim_p. This is the

case, because sim_{bp} requires the correspondences between the elements of the two input models that is provided by sim_p [8]. In consequence, weaker correspondences between the model pairs also result in a weaker behavioral similarity.

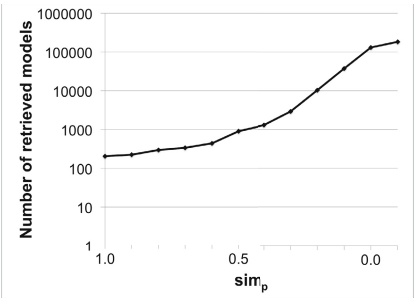

Fig. 2. Retrieved models for sim_p thresholds

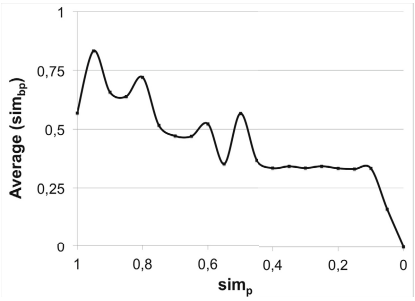

Fig. 3. Avg. sim_{bp} for thresholds sim_p

4.2 Similarity Thresholds for Detecting Similar Process Parts

Since different thresholds for the metrics sim_p and sim_{bp} entail completely different results, we investigated how these metrics should be configured in order to obtain the desired outcome. As illustrated in Figure 2, a step-wise decrease of sim_p threshold leads to an increase of the retrieved model pairs, while it reduces the average score of sim_{bp}. Apparently, sim_p has a significantly stronger influence on the detection of similar process parts than sim_{bp}.

Taking the results of Section 4.1 into consideration, we can identify several candidate groups based on a given threshold for sim_p. We summarized all candidate groups in Table 1. The first candidate group amounts to 204 model pairs and contains process models that are perfect clones. Process models of this group have an equal number of activities and equivalent activity labels.

The next candidate group has a sim_p score between 0.6 and 1.0. These models tend to include activity clones as well as semantically similar activities. In general, they differ in the number of activities. The activities themselves are either clones or semantically (very) close activities sharing similar actions and business objects. The example models in Figure 1 represent candidates of this group. For the example sim_p amounts to 0.62.

Candidate group 3 comprises models with an increasing semantic distance. While the semantic closeness of these models is still meaningful for higher scores in this range, models tend to become more and more distant for lower bound thresholds. Hence, the respective activities tend to either share a similar business object or the same action. Figure 4 provides an example of process models that is on the edge of group 3 and 4. The value of sim_p amounts to 0.27 and we note only weak semantic relations between the activities. Consider for instance the activities *Appropriation Request Processing* and *Process Inquiry*. Obviously, both activities share the action *process* that is performed on a business object leading to the correspondence.

As the largest candidate group 4 covers models with small semantic correspondences where activities only share a similar business object or a similar activity. Normally, human perception would ignore these models to be similar. Group 5 consists of all model pairs with no semantic correspondence scoring zero for sim_p.

Table 1. Candidate groups of sim_p

Group Number	Similarity Tendency	Threshold	Group size
1	Perfect Model Clones	$sim_p = 1.0$	204
2	Clone or semantically close models	$0.6 \leq sim_p < 1.0$	237
3	Semantically similar to distant models	$0.3 \leq sim_p < 0.6$	2,481
4	Semantically distant models	$0.0 < sim_p < 0.3$	128,240
5	Semantically dissimilar models	$sim_p = 0.0$	50,944

As already stated above, the decrease of the sim_p threshold corresponds with a decrease of the sim_{bp} value. We observed a high discrepancy between sim_p and $avg(sim_{bp})$ for group 1 (about 43%) and for group 2 (about 22%). In general, most process models only comprise two of the three used behavioral relations for sim_{bp}, which leads to a smaller score of sim_{bp}. The divergence of candidate group 3 is rather small (8%), whereas group 4 shows a high discrepancy again (about 65%). Again, two behavioral relations primarily contribute to sim_{bp} leading to a relatively high sim_{bp} compared to sim_p. Considering the example from Figure 4 sim_{bp} amounts to 0.42, which implies a similar control-flow. There is no discrepancy in group 5, because of the fact that no pair of corresponding activities is found, if $sim_p = 0.0$.

Due to this high discrepancies for higher or smaller thresholds for sim_p we conclude that sim_{bp} is not appropriate for the initial identification of similar process parts. This is supported by the fact that the correspondence of process models which is required for the calculation of sim_{bp} is strongly dependent on sim_p. We conclude that sim_{bp} is more appropriate to verify the correctness of an identified process part, while sim_p is able to identify meaningful candidates in a given collection. In other words, sim_p imposes a necessary and sim_{bp} a sufficient condition for process part detection.

5 Related Work

The work presented in this paper is related to three major streams of research: process model reuse, process model similarity and process model matching.

In disciplines such as software engineering reuse has a long tradition [14,15,16]. Identifying and reusing code fragments and software components does not only save time but also increases the maintainability of the resulting software artifacts. This is line with the Service-Oriented Architecture paradigm, where business

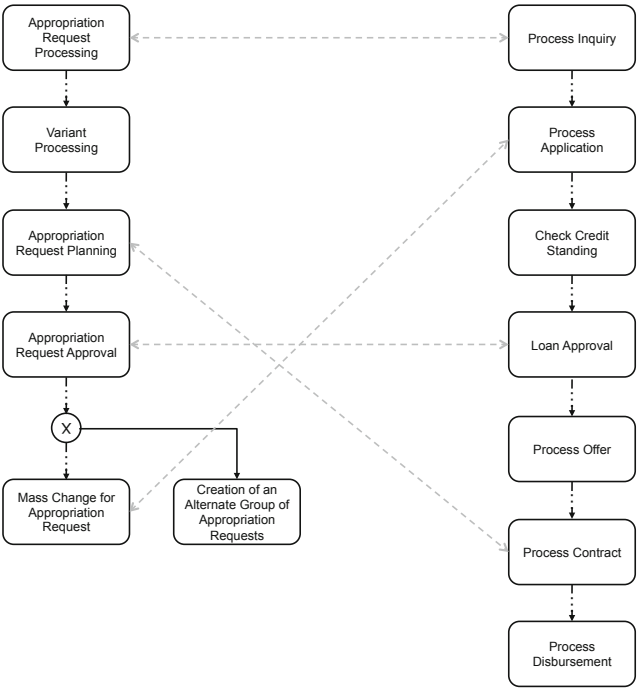

Fig. 4. Semantically distant process models with their corresponding activities

functionality is bundled and centrally provided [17,18]. Recently, this potential was also recognized for business process models. [3] proposed an approach for identifying clones in process model repositories. Our approach deviates from this technique, because it does not aim for detecting clones but explicitly targets the identification of similar process parts. Hence, our approach includes techniques for determining the semantic closeness of models based on their control flow and activity labels. Another technique was introduced by [19]. The authors propose an approach to support to the design and modeling of workflows by introducing a repository, which can be used for adapting workflow cases.

Techniques for determining the overall similarity of process models have been proposed by different authors. An overview is provided by [5]. Some of these works build on an ontology in order to match the labels [20]. Other approaches make use of control-flow based aspects. For instance, [8] use behavioral profiles to determine the similarity of processes. However, this approach assumes that the correspondences between the activities are already given. In general, the vast majority of these approaches focus on structural aspects and do not take semantic aspects into account. As this is crucial for the identification of similar process parts, we use a metric which builds on the semantic comparison of element labels.

The alignment or so-called matching of process models is closely connected with the similarity computation. Usually similarity scores are used to identify potential correspondences between two models which are then used as input for the matching technique [21,22]. However, while matchers try to find the best match for a given pair of models, our approach aims for quantifying the semantic similarity between them. Accordingly, a perfect match is not a prerequisite for our approach. In our context it is more important to identify models which have a certain degree of similarity. As a result, the identification of reuse candidates is automatically accomplished.

6 Conclusion

In this paper, we have proposed an approach that detects similar parts of process models. We exploit semantic and behavioural similarity aspects and challenged our approach against the SAP Reference Model. Our evaluation proved that the approach is applicable to real-world process models. It also revealed five candidate groups and configurable similarity thresholds enabling process part detection. We concluded that semantic similarity represents a necessary requirement for the detection of process parts, while the behavioral similarity formulates a sufficient condition ensuring correctness and consistency of the identified process parts.

There are several directions of our future work. First, we aim at improving our approach. This especially applies for the identification of model correspondences using semantic similarity techniques. Accordingly, we plan to incorporate more sophisticated matching algorithms in order to obtain a more precise sim_p value. Second, we plan to test our approach on further process model collections, as for instance the BIT process library [23] or the process repository of "Nationale Prozessbibliothek"[1]. In addition, we aim for testing our approach in an industrial setting. We think that the resulting feedback will help us to tailor approach to the actual needs of organizations. A third direction for future work is given by combining our approach with other technique facilitating reuse. Particularly, we plan to integrate the approach with automatic identification of services [24].

References

1. Rosemann, M.: Potential Pitfalls of Process Modeling: Part A. Business Process Management Journal 12(2), 249–254 (2006)
2. Reijers, H.A., Mans, R.S., van der Toorn, R.A.: Improved model management with aggregated business process models. Data Knowledge Engineering 68(2), 221–243 (2009)
3. Uba, R., Dumas, M., García-Bañuelos, L., La Rosa, M.: Clone Detection in Repositories of Business Process Models. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 248–264. Springer, Heidelberg (2011)
4. Leopold, H., Smirnov, S., Mendling, J.: On the refactoring of activity labels in business process models. Information Systems 37(5), 443–459 (2012)

[1] `www.prozessbibliothek.de`

5. Dijkman, R.M., Dumas, M., van Dongen, B.F., Käärik, R., Mendling, J.: Similarity of Business Process Models: Metrics and Evaluation. Information Systems 36(2), 498–516 (2011)
6. Grigori, D., Corrales, J., Bouzeghoub, M., Gater, A.: Ranking BPEL Processes for Service Discovery. IEEE Transactions on Services Computing 3(3), 178–192 (2010)
7. Zha, H., Wang, J., Wen, L., Wang, C., Sun, J.: A workflow net similarity measure based on transition adjacency relations. Computers in Industry 61(5), 463–471 (2010)
8. Kunze, M., Weidlich, M., Weske, M.: Behavioral Similarity – A Proper Metric. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 166–181. Springer, Heidelberg (2011)
9. Mendling, J., Reijers, H.A., Recker, J.: Activity Labeling in Process Modeling: Empirical Insights and Recommendations. Information Systems 35(4), 467–482 (2010)
10. Miller, G.A.: WordNet: a Lexical Database for English. Communications of the ACM 38(11), 39–41 (1995)
11. Lin, D.: An information-theoretic definition of similarity. In: Proc. 15th International Conf. on Machine Learning, pp. 296–304. Morgan Kaufmann, San Francisco (1998)
12. Weidlich, M., Mendling, J., Weske, M.: Efficient consistency measurement based on behavioural profiles of process models. IEEE TSE 37(3), 410–429 (2011)
13. Keller, G., Teufel, T.: SAP(R) R/3 Process Oriented Implementation: Iterative Process Prototyping. Addison-Wesley (1998)
14. Bellon, S., Koschke, R., Antoniol, G., Krinke, J., Merlo, E.: Comparison and evaluation of clone detection tools. IEEE TSE 33(9), 577–591 (2007)
15. Jacobson, I., Griss, M., Jonsson, P.: Software Reuse: Architecture, Process and Organization for Business Success, vol. 43. ACM Press (1997)
16. Heineman, G.T., Councill, W.T.: Component-Based Software Engineering: Putting the Pieces Together. Addison-Wesley Professional (2001)
17. Erl, T.: Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall PTR, Upper Saddle River (2005)
18. Marks, E.A., Bell, M.: Service-Oriented Architecture: A Planning and Implementation Guide for Business and Techonology. John Willey & Sons Inc. (2006)
19. Madhusudan, T., Zhao, J., Marshall, B.: A case-based reasoning framework for workflow model management. Data and Knowledge Engineering 50(1), 87–115 (2004)
20. Ehrig, M., Koschmider, A., Oberweis, A.: Measuring Similarity between Semantic Business Process Models. In: APCCM 2007, Ballarat, Victoria, Australia, vol. 67, pp. 71–80. Australian Computer Science Communications (2007)
21. 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)
22. La Rosa, M., Dumas, M., Uba, R., Dijkman, R.: Merging Business Process Models. In: Meersman, R., Dillon, T.S., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6426, pp. 96–113. Springer, Heidelberg (2010)
23. Fahland, D., Favre, C., Jobstmann, B., Koehler, J., Lohmann, N., Völzer, H., Wolf, K.: Instantaneous Soundness Checking of Industrial Business Process Models. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 278–293. Springer, Heidelberg (2009)
24. Leopold, H., Mendling, J.: Automatic Derivation of Service Candidates from Business Process Model Repositories. In: Abramowicz, W., Kriksciuniene, D., Sakalauskas, V. (eds.) BIS 2012. LNBIP, vol. 117, pp. 84–95. Springer, Heidelberg (2012)