

Towards Multi-perspective Process Model Similarity Matching

Michael Heinrich Baumann, Michaela Baumann,
Stefan Schönig^(✉), and Stefan Jablonski

University of Bayreuth, Bayreuth, Germany
{michael1.baumann,michaela1.baumann,
stefan.schoenig,stefan.jablonski}@uni-bayreuth.de

Abstract. Organizations increasingly determine process models to support documentation and redesign of workflows. In various situations correspondences between activities of different process models have to be found. The challenge is to find a similarity measure to identify similar activities in different process models. Current matching techniques predominantly consider lexical matching based on a comparison of activity labels and 1-to-1-matchings. However, label based matching probably fails, e.g., when modellers use different vocabulary or model activities at different levels of granularity. That is why we extend existing methods to compute candidate sets for N-to-M-matchings based on power-sets of nodes. Therefore, we impose higher demands on process models as we do not only consider labels, but also involved actors, data objects and the order of appearing. This information is used to identify similarities in process models that use different vocabulary and are modelled at different levels of granularity.

Keywords: Business process model · Process similarity · Model matching

1 Introduction

Organizations increasingly determine business process models for supporting the documentation and redesign of actual workflows as well as information system implementation [9]. In order to cover all the different peculiarities of a process typically several expert modellers from diverse business domains are involved in modelling activities [5].

In various situations correspondences between elements of different process models have to be found, e.g., when analysts of different departments modelled the same process or when merging similar processes of recently merged companies [1]. Furthermore, it is conceivable to detect correspondences in conjunction with process improvement, pattern identification or increase of efficiency. The challenge is to find a similarity measure to identify similar activities in different process models [1]. Therefore, current process model matching techniques

predominantly consider lexical matching scores based on a comparison of activity labels that appear in process models [12]. Furthermore, these methods only consider 1-to-1-matchings, i.e., only single nodes are compared per model [1].

Think about analysts of different departments who model the same process. Some analysts use a more technical vocabulary than others and some analysts tend to get more granular when modelling the process. As a consequence, it is likely that activity labels of resulting models considerably deviate from each other or that activities are divided in different chunks. In such situations, label based matching methods probably fail, i.e., lead to a low recall [6].

The intention of the work at hand is to find an adequate similarity measure to identify similar activities in process models that use rather different vocabulary and are modelled at different levels of granularity. Therefore, we extend existing process model matching methods to compute candidate sets for N-to-M-matchings based on power-sets of nodes. That is why we compute similarity measures to identify similar sets of activities in different process models. Of course, this implicates a considerably higher complexity. Furthermore, it is useless or even impossible to use only label matching to analyse sets of activities. Therefore, we impose higher demands on process models as we do not only want to match activities based on their labels, but also by analysing involved actors, data objects and their order of appearing in the model. Our approach is based upon the different perspectives of the perspective-oriented process modelling approach [7]. Consider the simple example of Fig. 1. Here, we identified a similarity between the activities A and B of the first model and the activity C in the second model since the combination of data objects produced by A and B relates to the set of data objects produced by C.

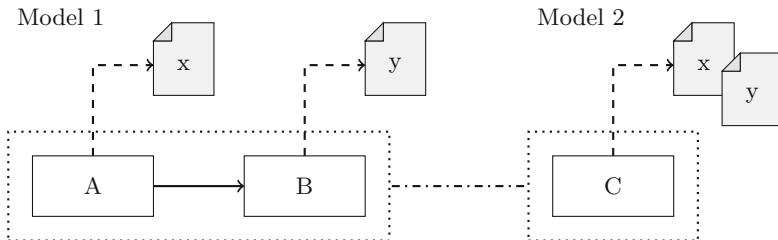


Fig. 1. Example process model similarity matching based on data objects

This information can help to reduce computation time and to identify similar activities in process models that use different vocabulary and are modelled at different levels of granularity.

2 Background and Related Work

The problem of matching two process models of a general form has already been discussed in several papers, like in [1–3, 10–16]. A lot of modeling notations

are available to capture business processes, e.g., Event-driven Process Chains (EPCs), UML Activity Diagrams and the Business Process Modeling Notation (BPMN) [9]. In the work at hand, we seek to abstract as much as possible from specific notations and therefore a process model is given according to following definition.

Definition 1 (process model). *Let $\mathcal{L} \subset \{s_1 s_2 \dots s_n \mid s_i \text{ is a character } \forall i \in \{1, 2, \dots, n\}, n \in \mathbb{N}\}$ be a set of labels. Then, a process graph is a tuple (N, E, λ) , where*

- N is a set of nodes,
- $E \subseteq N \times N$ is a set of edges and
- $\lambda : N \rightarrow \mathcal{L}$ is a function, that maps nodes to labels.

To determine the similarity between such models, first, the similarity of two nodes, i.e., of their labels has to be specified. This is usually done with a construct called string-edit similarity which is a measure for how strong one string resembles another one. It is defined with help of the so-called string-edit distance, *sed*. The string-edit distance of two strings is the minimal number of atomar string operations, that means insertion, deletion and substitution of one character, that is needed to transform one string into the other. Thus, the string-edit distance is an integer with a value not more than the length of the longer string.

Definition 2 (string-edit similarity). *For two strings s and t , the string-edit similarity Sim is given through*

$$Sim(s, t) = 1 - \frac{sed(s, t)}{\max(|s|, |t|)}.$$

Sim takes values between 0 and 1, where 1 can only be reached, if sed equals 0, that means the two compared strings are the same. As mentioned in [1, 3], it is also possible to do stemming before computing the string-edit similarity in order to get better values. Stemming is the name for a collection of several techniques, like deleting symbols, fillers, often and repeatedly used words, reducing words to their stem, translating words into one language, sorting words, etc. to get a standardized basis for the strings that have to be matched [17]. If possible, one can even use synonym dictionaries or a thesaurus, as suggested in [3], or ontologies to get optimal results for comparing two strings. Of course, the problem of homonyms cannot be solved this way, and there is no help when it comes to spelling errors or neologisms. In [6] another label-based similarity is proposed, namely the basic bag-of-words similarity which may be combined with label pruning.

The next step in getting an optimal matching of two models $G_1 = (N_1, E_1, \lambda_1)$ and $G_2 = (N_2, E_2, \lambda_2)$ is to consider a partial and injective mapping $M : N_1 \rightarrow N_2$, that maps nodes of G_1 to nodes of G_2 . This mapping is partial, as not all nodes of G_1 have to be mapped, and injective, as not all nodes of G_2 have to be met and those that are in the image of M may only have a one-elemental

inverse image. If $|N_1| < |N_2|$, not all nodes in N_2 can be met by M . With this mapping M , all nodes and edges of the two graphs G_1 and G_2 are element of one of the following sets.

Definition 3 (Substituted and deleted nodes). *For a mapping M as defined above, the set*

$$subn := \{n \in N_1 \cup N_2 \mid n \text{ is in the image or the inverse image of } M\}$$

is the set of all substituted/mapped nodes. Accordingly,

$$skipn := (N_1 \cup N_2) \setminus subn$$

is the set of all deleted nodes.

Definition 4 (Substituted and deleted edges). *Consider the mapping M . For all edges of G_1 and G_2 we say that an edge $(n_1, m_1) \in E_1$ is deleted from G_1 if there is no $(n_2, m_2) \in E_2$ with $M(n_1) = n_2$ and $M(m_1) = m_2$, and vice versa. Then*

$$skipe := \{(n, m) \mid (n, m) \text{ is deleted}\}$$

is the set of all deleted edges and

$$sube := (E_1 \cup E_2) \setminus skipe$$

is the set of all substituted/mapped edges.

Subsequently, the graph-edit similarity, a value of how good two graphs match, is computed with the shares of deleted nodes and edges compared to their total number and an average distance of the substituted edges where the variables $skipn$, $skipe$, $subn$ and $Sim(\cdot, \cdot)$ are defined as in Definitions 2, 3 and 4. These values are given as

$$\begin{aligned} - fskipn &= \frac{|skipn|}{|N_1| + |N_2|} \text{ (share of deleted nodes)} \\ - fskipe &= \frac{|skipe|}{|E_1| + |E_2|} \text{ (share of deleted edges)} \\ - fsubn &= \frac{2 \cdot \sum_{(n_1, n_2) \in M} (1 - Sim(n_1, n_2))}{|subn|} \\ &\text{(average distance of substituted nodes)} \end{aligned}$$

All these shares are element of the interval $[0, 1]$ and especially $fsubn$ takes values near 1, when there's not much similarity between the two compared graphs. Combining these three values with some weight factors $wskipn$, $wskipe$ and $wsubn$ that are element of $[0, 1]$ and sum up to 1 leads to the graph-edit similarity defined as

Definition 5 (graph-edit similarity induced by M). *For two models G_1 and G_2 and a mapping M the graph-edit similarity induced by M , $GSim_M$, is given through*

$$GSim_M(G_1, G_2) = 1 - (wskipn \cdot fskipn + wskipe \cdot fskipe + wsubn \cdot fsubn).$$

To get the best matching, that means the best mapping M between the two models, the graph-edit similarity induced by M has to be maximized with respect to M . The resulting value is called graph-edit similarity.

Definition 6 (graph-edit similarity). *The graph-edit similarity $GSim$ for G_1 and G_2 is obtained as*

$$GSim(G_1, G_2) = \max_M GSim_M(G_1, G_2).$$

For the implementation of this maximization problem, efficient algorithms are used, like Greedy or A*-Algorithms (see e.g. [2]), as the problem of finding the best M is of exponential order. With one of these algorithms it is now possible to efficiently match two process models on the same level of abstraction using a similar vocabulary. Obviously, by comparing process models with a strongly differing number of nodes the method presented so far will not provide satisfying results. Furthermore, a lot of information contained in the models is not considered. In [3] there are mentioned some possibilities to expand this matching technique and not only use the nodes' labels, as they might not lead to the desired results, but also their context with predecessor and successor relations. In [1] only the idea of expanding this method to more than one node in a successive/iterative way is mentioned. Nevertheless, all methods described so far are based on mapping single nodes to single nodes. Reference [8] introduces 1-to-n matchings, however, it does not imply other perspectives of business process models and only focuses on sequence flows during analysis. To eliminate the discussed disadvantages of existing techniques a method based on mapping sets of nodes to sets of nodes will be introduced.

3 Extended Definitions for Graph Matching

The work at hand expands previous ideas of single node matching to a procedure where sets of nodes are matched. Therefore, we impose higher demands on the process models as we do not only want to match nodes based on their description, but also based on involved positions, data objects and on their order of appearing in the model. Therefore, our approach is based upon the different perspectives of the perspective-oriented process modelling approach defined in [7]. We need these additional perspectives to get better matches, as the descriptions differ very much when comparing sets of nodes, and to reduce a combinatoric explosion, that results from the exponential number of matches we have to check. We also make some additional assumptions for these perspectives. Involved positions are arranged in some kind of tree, that represents their hierarchical structure. This tree can be seen as a combination of an organigram and maybe a population, which we use to reduce complexity of the model and to avoid introducing another mapping. Furthermore, all data objects appearing in the process models need to have a unique identifier. Based on these assumptions we can define an extended process model.

Definition 7 (Extended process model). Let $\mathcal{B} \subset \{s_1 s_2 \dots s_{n_{\mathcal{B}}} \mid s_i \text{ is a character } \forall i \in \{1, 2, \dots, n_{\mathcal{B}}\}, n_{\mathcal{B}} \in \mathbb{N}_0\}$ be a set of descriptions, \mathcal{A} a hierarchical tree of positions with nodes a and levels e , and $\mathfrak{D} = \{D_1, \dots, D_{n_{\mathfrak{D}}}\}$ a finite set of data objects. Then a process graph is a tuple (N, E, λ) with

- N being a set of nodes,
- $E \subseteq N \times N$ a set of edges and
- $\lambda : N \rightarrow \mathcal{B} \times \mathcal{A} \times \mathcal{P}(\mathfrak{D})$ a function, that maps nodes to entities.

For all process models to be matched, the sets \mathcal{B} , \mathcal{A} and \mathfrak{D} have to be the same. Note, that $\mathcal{P}(\cdot)$ indicates the power set.

Taking two process models, represented by their graphs, we define the extended graph-edit-similarity under consideration of the following sets.

Definition 8 (Set of deleted and substituted nodes). Let $G_i = (N_i, E_i, \lambda_i)$, $i = 1, 2$ be two models and $P_i \subset \mathcal{P}(N_i) \ni \emptyset$ a complete and disjoint partition of N_i (i.e. $\bigcup_{p \in P_i} p = N_i$ & $\forall p, p' \in P_i : p \cap p' = \emptyset, p \neq p'$), $i = 1, 2$. Further, let $M : P_1 \rightarrow P_2$ be a bijective function ($\emptyset \mapsto p_2$ and $p_1 \mapsto \emptyset$ means, that p_2 and p_1 are deleted, respectively, $p_1 \in P_1$, $p_2 \in P_2$), where $\neg(\emptyset \mapsto \emptyset)$. Then

$$\begin{aligned} \text{skipn} = & \{n_1 \in N_1 \mid n_1 \in p_1 \in P_1 : p_1 \xrightarrow{M} \emptyset\} \\ & \cup \{n_2 \in N_2 \mid n_2 \in p_2 \in P_2 : \emptyset \xrightarrow{M} p_2\} \end{aligned}$$

is the set of deleted nodes and

$$\text{subn} = (N_1 \cup N_2) \setminus \text{skipn}$$

is the set of substituted nodes.

Definition 9 (Set of deleted and substituted edges). Let $E_i^* = \{(a, b) \in E_i \mid \exists p_i \neq p'_i \in P_i : a \in p_i, b \in p'_i\}$ be a set of edges, that connect nodes from different elements of P_i , i.e., start node a is in p_i and end node b is in p'_i with $p_i \neq p'_i \in P_i$. Thus, with the assumption that $p_1 \neq p'_1, p_2 \neq p'_2$ we name with

$$\begin{aligned} \text{sube} = & \{(a, b) \in E_1^* \mid a \in p_1 \in P_1, b \in p'_1 \in P_1, M(p_1) = p_2, M(p'_1) = p'_2, \\ & \exists a' \in p_2 \in P_2, b' \in p'_2 \in P_2 : (a', b') \in E_2^*\} \\ & \cup \{(a', b') \in E_2^* \mid a' \in p_2 \in P_2, b' \in p'_2 \in P_2, M(p_1) = p_2, M(p'_1) = p'_2, \\ & \exists a \in p_1 \in P_1, b \in p'_1 \in P_1 : (a, b) \in E_1^*\} \end{aligned}$$

the set of substituted edges, i.e., the set of edges, that remain connectors of mapped pairs of nodes. Like in the definition above, let

$$\text{skipe} = (E_1^* \cup E_2^*) \setminus \text{sube}$$

be the set of deleted edges.

As one can see, we do not need to consider those edges, that connect nodes both being inherent in the same element of P_i . The similarity can now be defined analogously to that in the section before. However, we still need to determine the similarity of two sets of nodes p_1 and p_2 , “ $Sim(p_1, p_2)$ ”. Therefore, we consider different perspectives of nodes separately from each other and define a similarity value for each perspective. These values can then be combined with respect to some weight factor resulting in a global similarity value. In the next section, we look upon the four perspectives given in the process model and how we compute a similarity value for each perspective.

4 Similarity Between Sets of Nodes

We will start with the nodes’ description, i.e., the activity label, as we can apply a modification of the similarity concepts of Sect. 2, i.e., a modification of string-edit similarity. For positions, we examine their hierarchical structure in form of the given trees. Data objects have the function of an exclusion criterion due to their unique identifiers. Finally, we examine the order of sets of nodes where we need the concept of partial orders. In this paper, we focus on sequential process models which should be extended in future.

The Functional Perspective. As mentioned above, for the functional perspective, i.e., the nodes’ description, we can apply the well-known concepts of label matching, like string-edit similarity. The only difference is that we have to apply it to a set of nodes, i.e., to a set of strings. For this, we concatenate the descriptions of each node of the two sets with whitespace and in their order of appearance in the model.

Definition 10 (Extended string-edit similarity). *Let P_1 be a partition of graph G_1 and P_2 a partition of G_2 . Then, with $p_1 \in P_1$ consisting of nodes n_1, \dots, n_k with description strings s_1, \dots, s_k and $p_2 \in P_2$ consisting of nodes m_1, \dots, m_l with description strings t_1, \dots, t_l , we indicate with $s_1 \vee \dots \vee s_k$ and $t_1 \vee \dots \vee t_l$ the concatenated descriptions of p_1 and p_2 . The string-edit similarity of p_1 and p_2 is then defined as*

$$BSim(p_1, p_2) = 1 - \frac{sed(p_1, p_2)}{\max(|p_1|, |p_2|)},$$

where $sed(p_1, p_2) = sed(s_1 \vee \dots \vee s_k, t_1 \vee \dots \vee t_l)$ is the string-edit distance of p_1 and p_2 . $|p_i|$ stands for the length of the respective, underlying, concatenated string.

The range of $BSim$ is in $[0, 1]$ with a value of 1 if p_1 and p_2 have the same descriptions and a value close to 0 if they differ very much. Of course it is clear that comparing two sets of nodes with a strongly different number of elements, the result of $BSim$ has no chance to come close to 1. Thus, we strongly recommend using stemming-techniques like mentioned in Sect. 2.

The Data/Dataflow Perspective. Our intention is to compute a number that is 0 if the compared sets use completely different data objects and increases to 1 if all used objects appear in both sets. Furthermore, this perspective is meant to fulfill some important function in the context of practicability of our approach to distinctly reduce computation time of the hyper-exponential problem. That means if under a certain mapping M at least one assignment $p_1 \mapsto p_2$ has 0 similarity in the data/dataflow perspective the whole mapping M gets a similarity value of 0 and must not be considered any longer. It is likely that a lot of mappings can be rejected before their concrete similarity values have to be computed.

Before we define a similarity for occurring data objects of sets of nodes, we first have to specify a value for sets of data objects, as in one node more than one data object can be listed.

Definition 11 (Similarity for sets of data objects). For $D_1, D_2 \subset \mathfrak{D}$, $D_1 \cup D_2 \neq \emptyset$, we set

$$DSim(D_1, D_2) = \frac{|D_1 \cap D_2|}{|D_1 \cup D_2|}.$$

If $D_1 = D_2 = \emptyset$ we set $DSim(D_1, D_2) = 1$.

In fact, we do not have to look at single sets of data objects, but at all data objects in $p_i \in P_j$, which can be a set of sets of data objects. To handle this construct, we join all data object sets of the nodes and call this new set

$$D_{p_i} := \{D \mid \exists n \in p_i \in P_j : (\lambda(n))_3 = \mathcal{D} \wedge D \in \mathcal{D}\} = \bigcup_{n \in p_i} (\lambda(n))_3.$$

Now, we can define a similarity for the data/dataflow perspective for a set of nodes as follows

Definition 12 (Data/dataflow similarity)

$$DSim(p_1, p_2) = DSim(D_{p_1}, D_{p_2}) = \frac{|D_{p_1} \cap D_{p_2}|}{|D_{p_1} \cup D_{p_2}|}, \quad p_i \in P_i.$$

The Organizational Perspective. A similarity value has to be found that is 1 if the executing positions in the compared sets are the same and decreases to 0 the more organizational distance lies between the involved positions. In comparing the positions of two nodes of a given hierarchical tree it is possible to find a minimal number of edges, \tilde{k} , that have to be passed to get from one position to the other and the number of levels, \tilde{e} , that lie between them. Two positions on the same level have $\tilde{e} = 0$. To transform these two values into a similarity value, we set

$$ksim(A, B) := \frac{1}{\tilde{k} + 1} \quad \text{and} \quad esim(A, B) := \frac{1}{\tilde{e} + 1}.$$

Therefore, by comparing a position with itself, we get a value of 1 for both similarities, that means maximal similarity, and a value tending to 0, the more

edges and levels are between two positions. To combine these two similarity measures, we define

$$HSim(A, B) = \alpha ksim(A, B) + (1 - \alpha) esim(A, B),$$

with $\alpha \in [0, 1]$ being a weight factor, that allows to display some preferences for the position similarity. This value $HSim$ has to be extended to work for sets of nodes, that means a set of positions. This is done the following way:

Definition 13 (Organizational similarity). Let $\mathcal{M}_{p_i} \subset \mathcal{A}$ be the set of positions occurring in $p_i \in P_j$, i.e.,

$$\mathcal{M}_{p_i} = \{m \mid \exists n \in p_i \in P_j : (\lambda(n))_2 = m\}.$$

Then, for $p_i \in P_j$, it is

$$HSim(p_1, p_2) = \frac{\sum_{m_1 \in \mathcal{M}_{p_1}, m_2 \in \mathcal{M}_{p_2}} HSim(m_1, m_2)}{|\mathcal{M}_{p_1}| \cdot |\mathcal{M}_{p_2}|}.$$

Hence, we compute the similarity of every pair of positions from the two sets, add this values up and divide through the number of pairings to get an average value for position similarity. Note, that if there is more than one tree representing the hierarchical structure of an organization, a comparison of positions from different trees leads to a similarity value of 0.

The Behavioral Perspective. The behavioral perspective is somehow different to the other perspectives, as it is not a component of λ , but given through the nodes' sequential order in a process model. We examine whether the order of elements from P_1 is maintained, turned around or completely mixed up under the mapping M . To define a similarity with respect to this sequence, we use the partial order on P_i which is a result from the complete, disjoint decomposition of $P_i = \{p_i^1, \dots, p_i^t\} \subset \mathcal{P}(N_i)$ of the i -th model, $i = 1, 2$. Within this partial order, several states may occur, namely

$$\begin{aligned} p_i \succ p'_i &\Leftrightarrow \forall n \in p_i, n' \in p'_i : n \succ n', \\ p_i \prec p'_i &\Leftrightarrow \forall n \in p_i, n' \in p'_i : n \prec n', \\ p_i \sim p'_i &\Leftrightarrow p_i = p'_i \vee \exists n, m \in p_i, n', m' \in p'_i : n \succ n', m \prec m', \\ p_i &\sim \emptyset. \end{aligned}$$

\succ and \prec on N_i is given through the successive order of nodes in N_i . With this notation, we can now assign values to sets $p, p' \in P_1$ by comparing their order to the order of $M(p), M(p') \in P_2$. For this, we first want to distinguish between comparisons involving the empty set and all other pairs of sets and assign following values:

$$\gamma(p, p') = \begin{cases} 0, & \text{if } p \approx p' \vee M(p) \approx M(p') \\ 1, & \text{else.} \end{cases}$$

Next, the individual similarity values are assigned for each possible situation:

$$\nu(p, p') = \begin{cases} 1, & \text{if } p \prec p' \wedge M(p) \prec M(p'), \\ 1, & \text{if } p \succ p' \wedge M(p) \succ M(p'), \\ 1, & \text{if } p \sim p' \wedge M(p) \sim M(p'), \\ 0, & \text{if } p \prec p' \wedge M(p) \succ M(p'), \\ 0, & \text{if } p \succ p' \wedge M(p) \prec M(p'), \\ \frac{1}{2}, & \text{if } p \prec p' \wedge M(p) \sim M(p'), \\ \frac{1}{2}, & \text{if } p \succ p' \wedge M(p) \sim M(p'), \\ \frac{1}{2}, & \text{if } p \sim p' \wedge M(p) \prec M(p'), \\ \frac{1}{2}, & \text{if } p \sim p' \wedge M(p) \succ M(p'), \\ 0, & \text{if } p \approx p' \vee M(p) \approx M(p'). \end{cases}$$

It is possible to assign other plausible values to the different cases, for example $\frac{3}{4}$ in the third line. Function γ is necessary to make sure, later on, that we do not divide through 0.

5 The Extended Graph-Edit Similarity

The next step is to combine the defined similarity values for the different perspectives in addition to the values of *skipn* and *skipe*. Therefore, we transform all these values into normalized distances where 0 means full similarity and 1 greatest possible distance. We result in getting the following equations:

$$f_{skipn} = \frac{skipn}{|N_1| + |N_2|}$$

is the share of deleted nodes and

$$f_{skipe} = \frac{skipe}{|E_1^*| + |E_2^*|}$$

is the share of deleted edges, considering only the relevant ones. With

$$f_{subb} = \frac{\sum_{(p_1, p_2) \in M | p_1 \neq \emptyset \neq p_2} (1 - BSim(p_1, p_2))}{\sum_{(p_1, p_2) \in M | p_1 \neq \emptyset \neq p_2} 1}$$

we get an average normalized distance value for the functional perspective with respect to M . Analogously, we get a value for the organizational perspective through

$$f_{subh} = \frac{\sum_{(p_1, p_2) \in M | p_1 \neq \emptyset \neq p_2} (1 - HSim(p_1, p_2))}{\sum_{(p_1, p_2) \in M | p_1 \neq \emptyset \neq p_2} 1}.$$

For the data perspective, we have to do a distinction of cases to enable it to work as an exclusion criterion, as explained in Sect. 4. That is why we get

$$f_{subd} = \begin{cases} 1, & \text{if } \exists \emptyset \neq p \in P_1 : M(p) \neq \emptyset, DSim(p, M(p)) = 0 \\ \frac{\sum_{(p_1, p_2) \in M | p_1 \neq \emptyset \neq p_2} (1 - DSim(p_1, p_2))}{\sum_{(p_1, p_2) \in M | p_1 \neq \emptyset \neq p_2} 1}, & \text{else} \end{cases}$$

for the data perspective. For the behavioral perspective we also need to distinct between some cases, as there exist some degenerated mappings. Considering such mappings, we get

$$f_{subv} = \begin{cases} \frac{\sum_{p \neq p' \in P_1} (1 - \nu(p, p')) \gamma(p, p')}{\sum_{p \neq p' \in P_1} \gamma(p, p')}, & \exists p \neq p' \in P_1 : \gamma(p, p') \neq 0, \\ 1, & \text{for } f_{skipn} = 1, \\ 0, & \text{else.} \end{cases}$$

Now, these normalized distance measures are simply added with some weight factors and transformed back into a similarity measure with greatest possible similarity = 1 and 0 for no similarity to get the extended graph-edit similarity.

Definition 14 (Extended graph-edit similarity induced by M). *With weight factors $w_{skipn}, w_{skipe}, w_{subb}, w_{subv}, w_{subh} \in [0, 1]$ and $w_{subd} \in (0, 1]$ that sum up to 1 and can be chosen at one's own discretion we get the graph-edit similarity induced by M through*

$$GSim_M(G_1, G_2) = \begin{cases} 0, & \text{if } f_{subd} = 1, \\ 1 - (w_{skipn} \cdot f_{skipn} + w_{skipe} \cdot f_{skipe} \\ \quad + w_{subb} \cdot f_{subb} + w_{subd} \cdot f_{subd} \\ \quad + w_{subh} \cdot f_{subh} + w_{subv} \cdot f_{subv}), & \text{else.} \end{cases}$$

To get the global graph-edit similarity that means the best fitting mapping M we define.

Definition 15 (Extended graph-edit similarity)

$$GSim(G_1, G_2) = \max_M GSim_M(G_1, G_2).$$

For this task, again algorithms like Greedy or A* are used but adjusted to make use of the exclusion criterion, as the problem we focus here is of hyper-exponential order. Using these algorithms, the problem is still of exponential order, but can be made fairly efficient by using the mentioned exclusion criterion as a lot of possibilities are neglected and not completely computed.

6 Case Study and Evaluation

To give a more detailed insight of how the different similarities are computed and applied we present a case study and find the graph-edit similarity of the two

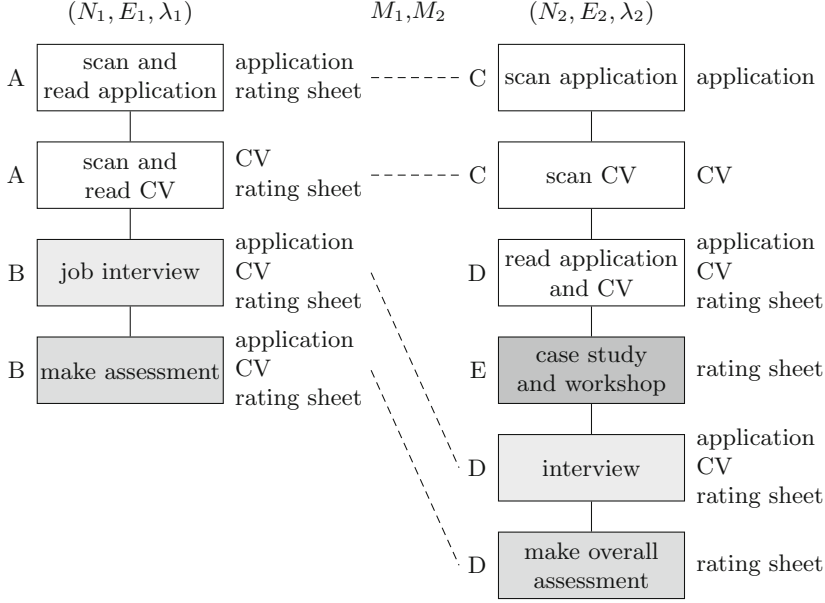


Fig. 2. Two process models for two resembling processes written down differently with 1:1-mapping M_1 and N:M-mapping M_2 . The positions at the left-hand side and the data at the right-hand side of the nodes are relevant for mapping M_2 .

exemplary process models (N_1, E_1, λ_1) and (N_2, E_2, λ_2) under a given mapping M_i . In fact, we will consider two mappings, one 1:1-mapping M_1 and one M:N-mapping M_2 . It becomes obvious that our extended approach provides better results than simple node-to-node mappings. For this, our two sequential process models and mapping M_1 are like in Fig. 2, where the nodes mapped by M_1 are indicated with dashed lines. It can be shown, that under application of stemming techniques this mapping M_1 is the best 1:1-mapping for these two models. For computing the graph-edit similarity we need the two values $f_{skipn} = 0.2$ and $f_{skipe} = 0.5$. Applying stemming techniques on the nodes' labels, we get for the average distance of substituted nodes a value of $f_{subn} \approx 0.54$. With all weights equalling $\frac{1}{3}$ we get for graph-edit similarity with respect to mapping M_1

$$GSim(G_1, G_2) = GSim_{M_1}(G_1, G_2) \approx 1 - \left(\frac{1}{3} \cdot 0.2 + \frac{1}{3} \cdot 0.5 + \frac{1}{3} \cdot 0.54\right) \approx 0.59.$$

For our second mapping M_2 we choose the M:N-mapping according to Definition 8 indicated with different colors in Fig. 2. The mappings λ_1 and λ_2 of G_1 and G_2 are given through

$$\begin{aligned}
- \lambda_1 : & \begin{cases} n_{11} \mapsto (\text{scan and read application, A, } \{\text{application, rating sheet}\}) \\ n_{12} \mapsto (\text{scan and read CV, A, } \{\text{CV, rating sheet}\}) \\ n_{13} \mapsto (\text{job interview, B, } \{\text{application, CV, rating sheet}\}) \\ n_{14} \mapsto (\text{make assessment, B, } \{\text{application, CV, rating sheet}\}) \end{cases} \\
- \lambda_2 : & \begin{cases} n_{21} \mapsto (\text{scan application, A, } \{\text{application}\}) \\ n_{22} \mapsto (\text{scan CV, A, } \{\text{CV, rating sheet}\}) \\ n_{23} \mapsto (\text{read application and CV, B, } \{\text{CV}\}) \\ n_{24} \mapsto (\text{case study and workshop, B, } \{\text{rating sheet}\}) \\ n_{25} \mapsto (\text{interview, B, } \{\text{application, CV, rating sheet}\}) \\ n_{26} \mapsto (\text{make overall assessment, B, } \{\text{rating sheet}\}) \end{cases}
\end{aligned}$$

where

$$\begin{aligned}
- N_1 &= \{n_{11}, n_{12}, n_{13}, n_{14}\}, \\
- N_2 &= \{n_{21}, n_{22}, n_{23}, n_{24}, n_{25}, n_{26}\}, \\
- E_1 &= \{(n_{11}, n_{12}), (n_{12}, n_{13}), (n_{13}, n_{14})\} \text{ and} \\
- E_2 &= \{(n_{21}, n_{22}), (n_{22}, n_{23}), (n_{23}, n_{24}), (n_{24}, n_{25}), (n_{25}, n_{26})\}
\end{aligned}$$

and thus the mapping M_2 is the following:

$$- M_2 : \begin{cases} \{n_{11}, n_{12}\} =: p_{11} \mapsto \{n_{21}, n_{22}, n_{23}\} =: p_{21} \\ \{n_{13}\} =: p_{12} \mapsto \{n_{25}\} =: p_{22} \\ \{n_{14}\} =: p_{13} \mapsto \{n_{26}\} =: p_{23} \\ \emptyset =: p_{14} \mapsto \{n_{24}\} =: p_{24} \end{cases}$$

This leads to edge sets

$$\begin{aligned}
- E_1^* &= \{(n_{12}, n_{13}), (n_{13}, n_{14})\} \text{ and} \\
- E_2^* &= \{(n_{23}, n_{24}), (n_{24}, n_{25}), (n_{25}, n_{26})\}.
\end{aligned}$$

With this, the share of deleted nodes has the same underlying set of nodes, whereas the share of deleted edges changes its denominator with respect to this new edge sets and we get

$$f_{skipn} = \frac{1}{4+6} = 0.1$$

and

$$f_{skipe} = \frac{3}{2+3} = 0.6.$$

The next step is to compute f_{subb} . Concatenating the respective descriptions and using the same stemming techniques as before we get for the stemmed descriptions

- $sed(\text{application CV read scan, application CV read scan}) = 0$,
 $BSim(p_{11}, p_{21}) = 1$
- $sed(\text{interview job, interview}) = 4$,
 $BSim(p_{11}, p_{21}) = 1 - \frac{4}{13} \approx 0.69$
- $sed(\text{assessment make, assessment make overall}) = 8$,
 $BSim(p_{11}, p_{21}) = 1 - \frac{8}{23} \approx 0.65$

This leads to a f_{subb} of value

$$f_{subb} = \frac{(1 - 1) + (1 - \frac{9}{13}) + (1 - \frac{15}{23})}{3} \approx 0.22.$$

For f_{subd} we have to determine $DSim$. It is

- $D_{p_{11}} = \{\text{application, CV, rating sheet}\}$,
- $D_{p_{12}} = \{\text{application, CV, rating sheet}\}$,
- $D_{p_{13}} = \{\text{application, CV, rating sheet}\}$,
- $D_{p_{21}} = \{\text{application, CV, rating sheet}\}$,
- $D_{p_{22}} = \{\text{application, CV, rating sheet}\}$ and
- $D_{p_{23}} = \{\text{rating sheet}\}$.

So, we get

$$- DSim(p_{11}, p_{21}) = 1, DSim(p_{12}, p_{22}) = 1, DSim(p_{13}, p_{23}) = \frac{1}{3}$$

and with that, it is

$$f_{subd} = \frac{(1 - 1) + (1 - 1) + (1 - \frac{1}{3})}{3} = \frac{2}{9} \approx 0.22,$$

as the exclusion criterion does not occur with this mapping M_2 .

For computation of f_{subh} we need the organizational structure for the positions A, B, C, D and E , that is given via the tree in Fig. 3. For the weights, we find it appropriate to give more weight to level similarity, so we choose $\alpha = \frac{1}{4}$. With this, we get

$$\begin{aligned} - HSim(p_{11}, p_{21}) &= \frac{HSim(A, C) + HSim(A, D)}{2} = \frac{0.25 \cdot \frac{1}{5} + 0.75 \cdot 1 + 0.25 \cdot \frac{1}{4} + 0.75 \cdot \frac{1}{2}}{2} \approx 0.62 \\ - HSim(p_{12}, p_{22}) &= HSim(p_{13}, p_{23}) = \frac{HSim(B, D)}{1} = 0.25 \cdot \frac{1}{3} + 0.75 \cdot 1 \approx 0.83 \end{aligned}$$

This leads to a value for f_{subh} of

$$f_{subh} \approx \frac{(1 - 0.62) + (1 - 0.83) + (1 - 0.83)}{3} = 0.24.$$

For the last part of the formula for graph-edit similarity we need to compute f_{subv} . For this, we have to consider the order of the sets p_{ij} under our mapping M_2 .

- $\gamma(p_{11}, p_{12}) = 1, \nu(p_{11}, p_{12}) = 1$,
- $\gamma(p_{11}, p_{13}) = 1, \nu(p_{11}, p_{13}) = 1$,

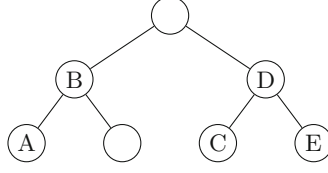


Fig. 3. Organizational structure of the five positions in our example

- $\gamma(p_{11}, p_{14}) = 0, \nu(p_{11}, p_{14}) = 0,$
- $\gamma(p_{12}, p_{13}) = 1, \nu(p_{12}, p_{13}) = 1,$
- $\gamma(p_{12}, p_{14}) = 0, \nu(p_{12}, p_{14}) = 0,$
- $\gamma(p_{13}, p_{14}) = 0, \nu(p_{13}, p_{14}) = 0.$

Therefore, it is

$$f_{subv} = \frac{(1-1) \cdot 1 + (1-1) \cdot 1 + (1-1) \cdot 1}{3} = 0,$$

which means perfect behavioral similarity.

With weights equalling $\frac{1}{6}$, especially $wsubl \neq 0$, we get for the graph-edit similarity with respect to M_2

$$\begin{aligned} GSim_{M_2}(G_1, G_2) &\approx 1 - \left(\frac{1}{6} \cdot (0.1 + 0.6 + 0.22 + 0.22 + 0.24 + 0)\right) \\ &= 0.77 \end{aligned}$$

We can conclude that $GSim(G_1, G_2) \geq 0.77$ as there may exist a mapping M_3 that leads to a better matching of the two graphs than M_2 .

7 Conclusion and Future Work

The contribution of the work at hand is to find an adequate similarity measure to identify similar activities in process models that use rather different vocabulary and are modelled at different levels of granularity. We extended existing process model matching methods to compute candidate sets for N-to-M-matchings based on power-sets of nodes. In order to cope with the increasing complexity we imposed higher demands on process models. Therefore, we did not only consider activity labels but also comprised involved actors, data objects and the order of activities. Using this additional information we reduced computation time and identified similar activities in process models that use different vocabulary and are modelled at different levels of granularity. Table 1 provides a short comparison of traditional 1-to-1-matching techniques and the N-to-M-matching of the work at hand.

For future work it is conceivable to extend the set-of-nodes-matching to general process models containing gateways and the possibility, that not only one specific position, but roles are allowed for the organizational perspective. For the

Table 1. Comparison of 1:1- and N:M-matching techniques

	1:1	M:N
Utilized dimensions	$fskipn, fskipe, fsubn$	$fskipn, fskipe, fsubb, fsubd^*, fsubh, fsubv$
Process model	(N, E, λ) with $\lambda : N \rightarrow \mathcal{L}$	(N, E, λ) with $\lambda : N \rightarrow \mathcal{B} \times \mathcal{A} \times \mathcal{P}(\mathfrak{D})$ with \mathcal{A} being a tree
Findings	1:1-mappings (+ extensions)	1:1-, 1:N-, M:N-mappings
Runtime/complexity	Low	High (improved by special assumptions, etc.)
Robustness	Possibly against inaccurate labels	against inaccurate descriptions, different granularities

weights in the formula of the graph-edit similarity and the weights of computing the organizational similarity, we proposed to choose their values according to everybody's own preferences. It is conceivable that if training graphs with given similarities are available, one may find the best suiting values for the weights with help of statistical methods, like maximum likelihood estimation.

Acknowledgement. The presented work is developed and used in the project “Kompetenzzentrum für praktisches Prozess- und Qualitätsmanagement”, which is funded by “Europäischer Fonds für regionale Entwicklung (EFRE)”.

The work of Michael Heinrich Baumann is supported by Hanns-Seidel-Stiftung e.V.

References

1. Dijkman, R., Dumas, M., García-Bañuelos, L., Käärik, R.: Aligning Business Process Models (2009)
2. Dijkman, R., Dumas, M., García-Bañuelos, L.: Graph matching algorithms for business process model similarity search. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 48–63. Springer, Heidelberg (2009)
3. Dijkman, R., van Dongen, B., Käärik, R., Mendling, J.: Similarity of business process models: metrics and evaluation. Inf. Syst. **36**(2), 498–516 (2011)
4. Minor, M., Tartakovski, A., Bergmann, R.: Representation and structure-based similarity assessment for agile workflows. In: Weber, R.O., Richter, M.M. (eds.) ICCBR 2007. LNCS (LNAI), vol. 4626, pp. 224–238. Springer, Heidelberg (2007)
5. Dijkman, R.: A Classification of Differences between Similar Business Processes (2007)
6. Klinkmüller, C., Weber, I., Mendling, J., Leopold, H., Ludwig, A.: Increasing recall of process model matching by improved activity label matching. In: Daniel, F., Wang, J., Weber, B. (eds.) BPM 2013. LNCS, vol. 8094, pp. 211–218. Springer, Heidelberg (2013)

7. Jablonski, S., Bussler, C.: Workflow Management: Modeling Concepts, Architecture and Implementation. International Thomson Computer Press, London (1996). ISBN: 1850322228
8. 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)
9. Weske, M.: Business Process Management: Concepts, Languages, Architecture. Springer, New York (2007)
10. Branco, M.C., Troya, J., Czarnecki, K., Küster, J.M., Völzer, H.: Matching Business Process Workflows across Abstraction Levels, Models (2012)
11. 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)
12. Leopold, H., Smirnov, S., Mendling, J.: On the refactoring of activity labels in business process models. *Inf. Syst.* **37**(5), 443–459 (2012)
13. Dumas, M., García-Bañuelos, L., Dijkman, R.M.: Similarity search of business process models. *Bull. Tech. Comm. Data Eng.* **32**(2), 23–28 (2009)
14. Ehrig, M., Koschmider, A., Oberweis, A.: Measuring similarity between semantic business process models. In: Proceedings of the 4th Asia-Pacific Conference on Conceptual Modelling, Ballarat, Victoria, Australia, pp. 71–80 (2007)
15. van der Aalst, W.M.P., de Medeiros, A.K.A., Weijters, A.J.M.M.: Process equivalence: comparing two process models based on observed behavior. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 129–144. Springer, Heidelberg (2006)
16. van Dongen, B.F., Dijkman, R., Mendling, J.: Measuring similarity between business process models. In: Bellahsene, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 450–464. Springer, Heidelberg (2008)
17. Lovins, J.B.: Development of a stemming algorithm. *Mech. Transl. Comput. Linguist.* **11**, 22–31 (1968)

Enterprise and Organizational Modeling and Simulation
10th International Workshop, EOMAS 2014, Held at
CAiSE 2014, Thessaloniki, Greece, June 16-17, 2014,
Selected Papers
Barjis, J.; Pergl, R. (Eds.)
2014, XII, 219 p. 88 illus., Softcover
ISBN: 978-3-662-44859-5