On Behavioral Process Model Similarity Matching: A Centroid-based Approach

Michaela Baumann*, Michael Heinrich Baumann†, and Stefan Jablonski*
*Institute for Computer Science
†Institute for Mathematics
University of Bayreuth, Germany
Email: {michaela.baumann,michael.baumann,stefan.jablonski}@uni-bayreuth.de

***Abstract*—As business process models have a broad scope of applications, e.g., in science or in business administration, the problem of handling large amounts of process models arises. One helpful tool for dealing with this amount of models is to reduce it by using similarity measures in order to detect similar models that can be merged. A set of similar models may be replaced by one model. As a pure similarity of labels is often not enough to compare process models, other process perspectives are involved for calculating similarities. The current paper works on the process models' behavior, which is one such perspective. A problem that arises when comparing two models and that is covered in this paper is that one of a differing granularity of process steps. Due to this granularity problem M-to-N mappings are considered. The present paper provides a centroid-based and so easily computable method for calculating behavioral similarity values for process models, which is constructed for M-to-N mappings, and a short evaluation of it.**

Keywords–Business process model; Behavioral process model similarity; M:N-Matching

I. INTRODUCTION

Not only for documentation purposes, business process models have been established in a large amount of organizations. They also serve as supportal means for communication, for training employees, and redesigning actual workflows [1]. These widely spread applications lead to vast process model repositories in enterprises, that have to be managed somehow [2]. One of these management purposes is to find similar models in order to reduce the tremendous amount of repository elements by detecting and merging similar models. Similar models can emerge when the same process is modeled multiple times, either for different end user groups or in different variations for the same user group. The authors of [3] worked out a total of nine categories for application fields of similarity measures, amongst them process merging, facilitating reuse of models [4], and service discovery. Also, process model matching can be used in the fields of compliance and conformance checking, the latter especially in terms of (process) log data, which can be seen as a number of sequential process models. But usually, the models are developed by different persons and thus have different levels of granularity, which means, that process steps in different models are modeled with a different fineness [5]. Especially for human tasks, it is not prescribed how fine-grained the particular steps have to be, and the detail level strongly depends on the purpose the model has to fulfill, on the attributes of the executing agents, or simply on the modeler's preferences. Furthermore, the terminology, i.e., the way of defining names, labels, etc. varies from model to model, and hence a comparison of these models only using their labels is challenging [6]. These two issues often lead to the fact that actually very similar or even equal models are not recognized as such. Because of this and due to the wide variety of modeling languages and notations, like Event-driven Process Chains (EPCs), Petri Nets, Unified Modeling Language (UML) Activity Diagrams, Workflow Nets, the Business Process Model and Notation (BPMN), or the Business Process Execution Language (BPEL), perfect matches, i.e., a true/false answer to the question if two models are the same, cannot be expected. Instead, a degree of similarity, a value between 0 and 1 where 0 means completely different and 1 is an indication for (virtually) identical models, depending on the definition of the respective similarity measure, is desired.

These measures can be defined on different disjoint aspects of process models: on node information, on process structure, and on execution semantics [2]. Node information is attached to each process model element, especially activities, and can again be split up into the description of process model elements, assigned roles or agents, ingoing and outcoming data objects, and operational means. Process structure refers to graph structure when taking a process model as a graph, and execution semantics refers to the question, how, i.e., in which order and under which circumstances (parallel, inclusive, exclusive, loop, etc.), process model elements may be executed. A behavioral similarity usually relies on the execution semantics of a process model. In order to take into account all of this information about process models and to allow for a wide range of modeling notations, we define a process model as instanced in Definition 1. In principle, each process model, whatever modeling notation is used, is a graph consisting of bubbles and directed arcs. Bubbles are elements like activities, events, and gateways, connected through arcs. Execution order is thus more or less prescribed, and human influence, i.e., decisions, are involved through (exclusive or inclusive) gateways. Note that this holds for imperative process models. Declarative process models, like the Case Management Model and Notation (CMMN) [7], take a different approach and allow for a greater human influence.

Definition 1 (Process Model). *A process model is a tuple $G = (N, E, \lambda, \delta)$ where $N = A \cup \{start, end\} \cup S_{AND} \cup S_{XOR} \cup S_{OR}$ is the finite, non-empty set consisting of all model elements, and $E \subseteq N^2$ is the set of all directed edges connecting the elements of N. Function λ assigns a description, a data set, organizational, and operational resources to each of the tasks in A. Function δ assigns constraint descriptions to some edges.*

Set S_{AND} is the set of all (split and merge) parallel gateways. Sets S_{XOR} and S_{OR} are the sets of all (split and merge) exclusive or inclusive gateways, respectively. $start$ denotes the start event of the process and end the end event. Every process has exactly one start and one end event. The activity tasks are summarized in set A. Functions λ and δ are mentioned for the sake of completeness, but are not discussed further, as they are not of importance for the behavior of a process model. Task description, used data, assigned agents, assigned tools and behavior can be treated separately when analyzing process models, as these five perspectives are completely orthogonal to each other [8]. Similarity of descriptions can be determined via string-edit operations, see for example [9], whereas data, organizational and operational similarity can be calculated with set-based methods, like the Jaccard coefficient [10] [11]. A more detailed definition of multi-perspective process models can be found in [12]. The elements of set N are sometimes also called nodes.

Definition 1 allows for many kinds of process models, even if they do not provide information about all process perspectives. For instance, if the non-human resources, that is the operational perspective, is not given in the model, the corresponding co-domain of function λ is left empty. Or if inclusive gateways are not included, then it is set $S_{OR} = \emptyset$. Human influence on the behavior is covered by exclusive (XOR) and inclusive (OR) gateways and the agents' decisions during the execution. At design level, however, this influence, i.e., the decision at run time, does not affect the model behavior. In imperative process models, behavior is strongly restricted.

The focus of the work at hand lies on the behavioral aspect of process models, i.e., on control flow and how two models can be compared with respect to this aspect. During the matching process – this is what we call the process of finding a similarity value between two models – the tasks of two process models are not compared one-to-one (single task compared to single task), but they will be grouped into sets to encounter the problem of differing granularity. In many cases, one-to-one mappings are not able to represent the correct correpondences. For example, when one activity in the first process model is split up into three process steps in the second model (imagine a manager's and a technician's view on a certain process), a one-to-one mapping would not provide a satisfying result. After having established the task sets, centroids, i.e., average positions (see Definition 4), average repeatability, and average optionality are calculated to determine behavioral similarity. As far as the authors know, this distinction of behavior into the three dimensions position, repeatability, and optionality has not yet been done explicitely in previous work.

In [13], process model elements are classified into, among other things, alternative or loop fragments, that resemble optional and repeatable elements. Furthermore, these centroids will be able to punish sets of activities that are widely spread over the whole process model or that have strongly differing manner. See Figure 1 for an example of two process models with schematical positional centroids. The mapped task sets are indicated with different fillings. The resulting behavioral similarity value can then be combined with other similarity values, e.g., description similarity or data similarity, to get a better matching score that is more independent of local errors, i.e., that is more robust against errors in certain process model aspects [10]. To put it together, the method presented in this paper provides two main results: A normalized similarity value for two process models based on their behavior and a mapping that indicates the resembling parts of them, which will be called M-to-N mapping. The mapping is needed to compute the behavioral similarity value. This approach is also known in related work, e.g., in [2] and [14] using 1-to-1 mappings. The advantage of such a method compared to a pure similarity calculation without presupposing a mapping is that the correspondences are provided in the same step and do not have to be detected in a separate step afterwards. The M-to-N mapping, also used in [11] for organizational and operational similarity, is defined in Definition 2.

Definition 2 (M-to-N mapping)**.** *Let* $G_i = (N_i, E_i, \lambda_i, \delta_i)$, $i = 1, 2$ *be two process models, with* $A_i \subset N_i$ *being the set of activities or tasks of each process model and* $P_i \subset \mathcal{P}(A_i) \ni \emptyset$ *a complete and disjoint partition of* A_i, $i = 1, 2$. *A mapping between* G_1 *and* G_2 *is defined as a bijective function* $M : P_1 \to P_2$. *In particular,* $\emptyset \mapsto p_2$ *and* $p_1 \mapsto \emptyset$ *means, that* p_2 *and* p_1 *are deleted, respectively, where* $p_1 \in P_1$, $p_2 \in P_2$, *and* $\neg(\emptyset \mapsto \emptyset)$.

As Definition 2 shows, sets of activities are mapped rather than single tasks, which induces the term M-to-N mapping. These sets of activities are achieved by establishing a partition of set A. In Figure 1, the tasks of the left model are partitioned into four sets (one of them the empty set), as well as the tasks of the right model. Tasks are indicated through rectangles with rounded corners, the diamonds represent gateways. Diamonds filled with "x" are exclusive, filled with "+" are parallel, filled with a small ring inclusive. In Figure 1, the meaning of the gateways is not of importance. Start and end event are denoted by circles. The mapping consists of four elements, a 2-to-3 (dotted) and a 2-to-1 (striped) mapping element, as well as a 1-to-0 (white) and a 0-to-1 (gray) element (the deleted nodes). In cases of process models strongly differing in granularity, a comparison explicitly applying a M-to-N mapping may provide better results than methods presented in most of the previous work. Furthermore, no complex calculations are needed for the centroid-based similarity presented in Section III. Regarding the draft version of this paper [12], requirements for compared process models, like block-structure, have been relaxed.

The remainder of this paper is organized as follows: The next section gives a rough overview of existing similarity measures and process model matching methods. Section III then introduces the behavioral similarity measure in its three dimensions step by step. An extension for penalized similarity measures is given, too. Thereafter, in Section IV, a short evaluation is performed. Section V revises the paper and gives ideas for future work.

II. Background and Related Work

In the literature, many techniques and methods for calculating the similarity, or, on contrast, the distance of process models, are presented. The authors of [3] provide a comparing overview of some of these techniques. Other collections and comparisons of several matching techniques can be found in [15] and [16]. One way of measuring the similarity between a pair of process models is to first define a mapping between

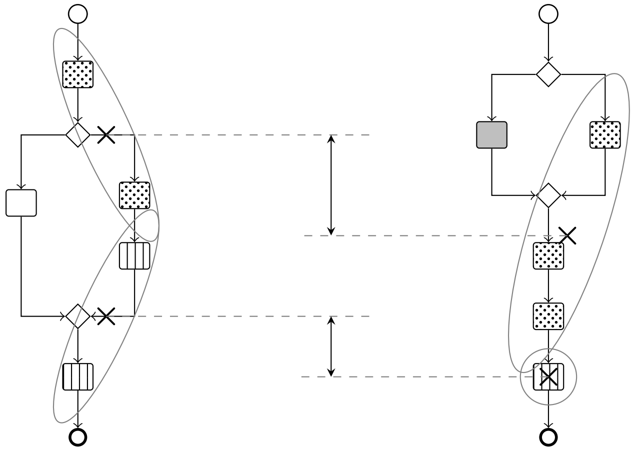


Figure 1: Schematical representation of the comparison of (positional) centroids for mapped sets of process tasks.

these two models. This mapping can either assign one node of the first model to one node of the second model, which often leads to a partial injective function [14], or map a set of nodes of one model to a set of nodes of the second model [10]. Thereby, we will refer to the latter defined mapping as M-to-N mapping, which is defined according to Definition 2 and is just a generalization of the former 1-to-1 mapping definition through an extension to powersets. As the authors of [9] suggest, for many scenarios, e.g., when processes have been developed independent of each other, a M-to-N mapping is preferred to a simple 1-to-1 mapping. M-to-N mappings are capable to overcome problems of granularity levels, which is one of the future tasks stated in [17]. In [6], a method for establishing N-to-1 mappings is presented, but not extended to a M-to-N mapping due to the applied matching techniques.

A. Label-based and structural similarity values

After having established a mapping between the elements of process models, similarity values between these elements can be computed. Depending on the given models, various information is used for this computation step. A similarity value based on the activity labels of a process model is, e.g., presented in [14], [18], and [19]. It makes use of the so-called string-edit (Levenshtein) distance and other string-modifying techniques like stemming [14] or replacing certain words through synonyms [20]. This similarity is often referred to as syntactic, semantic, or linguistic similarity. Another information that can be used for comparing process models is information about authorized agents and assigned input and output data of each activity, which is available, for example, in BPMN process models [21]. In Definition 1, this information can be found in function λ, that maps each node to a four-tuple consisting of node description, authorized agents, tools to be used, and consumed and produced data. This additional information can be analyzed lexically [6] or when applying M-to-N mappings through set-based methods performed on the subjects' or objects' identifiers, as it is done in [10] and [11]. Another important aspect of process models is the arrangement of the model elements. Basically, this arrangement can be categorized into two different similarity metrics: structural/contextual similarity and behavioral similarity [9] [22]. Structural and contextual similarity is, however, not using a preceding mapping, so we leave these kinds of measures out as we want to use a similarity measure showing the model equivalences at the same time. Behavioral matching techniques are based on the execution semantics of a process model [14], which means, that, e.g., parallelism or exclusiveness of model elements as well as their possible execution order is respected.

B. Various Definitions of Behavioral Similarity

Different approaches for measuring behavioral similarity are developed in literature. In [10], a computing method for M-to-N mappings is suggested that makes use of partial order relations of the activity elements, but is limited to serialized process models without any gateways, which is a strong limitation for most models. Behavioral profiles, a set of valid relations (strict, exclusive, interleaving) between every two process model elements, are introduced in [22] and [23] to define different behavioral similarity values. This approach is, however, applicable if the process models are mapped 1-to-1 and hardly transferable to M-to-N mappings. Another common method is to look at the traces of the process models to be compared [2]. Even if there is only a finite set of traces in loop-free process models, the problem of computing the trace-based behavior of a model is NP-hard [9]. An explicit discussion of trace-based methods is presented in [3]. Regarding partial traces is a variant of this trace-based approach and discussed in [24]. To overcome the computational complexity of traces, an approximation via casual footprints can be performed [2]. Casual footprints define the so-called look-back and look-ahead links of single process model elements [9] [20] and not for sets of tasks. Furthermore, this similarity value takes sequential, parallel, or exclusive behavior of the model elements, which is important information about a process model's behavior, only insufficiently into account [9]. A further approach of determining similarity between two process models is given in [25], where process models are compared with respect to some typical behavior that is gained from process event logs. However, as event logs or typical reference models are not always given, the applicability is restricted to particular cases. The aim of the paper is to derive a behavioral similarity function whose calculation is not too difficult, in contrast to the calculation of casual footprints [3], and that is at the same time suitable for M-to-N mappings. All in all, an approach like ours makes use of already existing concepts like embeddability into graph-edit similarity, but is adjusted to situations where previous approaches are not able to detect similarity or to take into account all given information.

Another work dealing with comparison of workflows, workflow systems, and the expressiveness of workflow languages is [26]. It is very broad, but uses simulation and not the models itself for comparison. When using simulation, there is the difficulty of finding significant samples and the fact, that all results are statistical, i.e., hold under a certain significance level. It also introduces a lot of notion and transformation methods to tree and automaton representations for the models. Furthermore, the different process perspectives are not considered separately and concrete correspondences are not worked out explicitely. The authors of [27] also worked out a method to determine behavioral similarity based on arbitrary alignments (thus, also for M-to-N mappings), where overlapping of the mapped node sets is allowed, which is a great advantage of this work. In return, they only allow for acyclic, i.e., loop-free, process models based on causal nets. Repeatability is therefore not considered.

No matter what kind of mapping is established and which kind of information (label, behavior, etc.) is used to calculate similarity, the further progress is always the same. After having computed the similarity value for one particular mapping, this value is maximized over all possible mappings to get the best correspondences between the two models. I.e., $Sim(G_1, G_2) = \max_M Sim_M(G_1, G_2)$ gives the similarity value for the two models G_1 and G_2 where function Sim_M calculates their similarity induced by mapping M [10] [14]. For this optimization step, greedy or A* algorithms working on task set A in the case of 1-to-1 mappings or on the powerset of A in the case of M-to-N mappings can be used [18].

III. The Centroid-based Behavioral Similarity Measure

For our work, we rely on process models given according to Definition 1, that even may contain loops. Models with loops particularly allow for control loops. The similarity calculation further assumes a M-to-N mapping between the two models that shall be compared and makes use of the centroids of the task sets. In particular, a M-to-N mapping M is given according to Definition 2, i.e., a partition P_1 of activities A_1 of the first process model G_1 is mapped bijectively to a partition P_2 of activities A_2 of the second process model G_2, and every element of a partition $p \in P_i$ is a set of activities of the underlying process model, i.e., $p \subseteq A_i$.
The targeted behavioral similarity measure considers the order of nodes given by the control flow, but also takes into account mandatory and optional activities as well as repeatable ones, which we call the three dimensions of behavior as already mentioned in the introduction. A penalty score is added to neglect sets of heterogenous tasks, e.g., widely spread sets of tasks.

A. Positional Similarity

The first behavioral dimension reflects the location of nodes in a process model. This location is specified as a relative position to obtain comparability, i.e., the position of a node is a number in $[0, 1]$, where a value close to zero indicates a position at the beginning of the model and a value close to one a position at the end. In particular, the position of a node is given through the length of the shortest chain (sequence of consecutive directed edges) from the $start$ event to the node, divided by the length of the shortest chain going from $start$ to end while passing the node. This is specified in Definition 3. Function $m(\cdot, \cdot)$ in Definition 3 gives the length of the shortest chain from one node to another. By using these minimal chains the problem of infinitely long chains resulting from loops is avoided. The position of a set of nodes, i.e., the positional centroid, given in Definition 4, is then computed by simply taking the arithmetic average of the single positions, i.e., summing up the single positions and dividing through the number of elements in the set. This again results in a value in $[0, 1]$. Note that the definitions basically apply for all nodes, like events, tasks and gateways, but we will later on use only the activity tasks' positions, as only the tasks are mapped by a M-to-N mapping according to Definition 2.

Definition 3 (Node position)**.** *The position $\pi(n)$ of node $n \in N$ is $\pi(n) = \frac{m(start,n)}{m(start,n)+m(n,end)}$.*

Positions that are fixed in all process models we consider are $\pi(start) = 0$ and $\pi(end) = 1$ as $m(start, start) = m(end, end) = 0$. In the following definition, P denotes a partition of a model whose elements p are sets of nodes, i.e., $p \subseteq N$.

Definition 4 (Centroid of a set of nodes)**.** *The centroid $\pi(p)$ of $p \in P$ is given through $\pi(\emptyset) = \texttt{NULL}$ and*

$$\pi(p) = \tfrac{1}{|p|} \sum\nolimits_{n \in p} \pi(n), \ p \neq \emptyset. \quad (1)$$

All occuring $NULL$-values are ignored in the further calculations in this paper, but lower the overall similarity when combining the behavioral similarity with other kinds of similarity like label- or resource-based similarity. The $NULL$ values occur if nodes are not mapped, like it is the case in Figure 1 for the white and the gray task. The behavioral similarity of two models, represented by their partitions P_1 and P_2, then combines the differences of the centroids of the mapped sets of tasks again as an arithmetic mean.

Definition 5 (Behavioral similarity 1)**.** *For two partitions P_1, P_2 of process models G_1, G_2 induced by a mapping M, the first dimension of behavioral similarity, the position-based similarity, is given through*

$$VSim_M^{\pi}(P_1, P_2) = \tfrac{1}{|P_1|} \sum\nolimits_{p \in P_1} (1 - |\pi(p) - \pi(M(p))|) \,. \quad (2)$$

Figure 1 shows two centroid differences: $|\pi(p_{dotted}) - \pi(M(p_{dotted}))|$ and $|\pi(p_{striped}) - \pi(M(p_{striped}))|$. Low differences, i.e., similar positions, lead to high similarity values due to the modification $1 - |\cdot|$ in formula (2). The formula can also be formulated with the models themselves via $VSim_M^{\pi}(G_1, G_2) := VSim_M^{\pi}(P_1, P_2)$, as mapping M applied on the models induces the partitions.

B. Repeatability and Optionality

Besides the position value π, we can also assign a repeatability value ϱ and an optionality value o (*omikron*) to each node. These additional dimensions of the behavior of process models display the execution with regard to the different gateway types. The approach for these two is similar to that of Section III-A. First, a repeatability/optionality value is defined for single nodes. Then, a repeatability/optionality value for a set of nodes is established through the arithmetic average of the single values. Finally, these values are combined for the partitions induced by the mapping.

Definition 6 (Node repeatability)**.** *The repeatability $\varrho(n)$ of node $n \in N$ is $\varrho(n) = 1$ if n can be executed more than once in one process instance and 0 otherwise.*

The repeatability value provides information if a node can be executed more than once in one process instance, i.e., if it is involved in a XOR-loop. In BPMN, it is possible to mark activities as loop tasks which are also treated as repeatable nodes. Another property of nodes is their optionality, i.e., if a node has to be executed in one process instance or if the process can finish without having executed it. Optionality can be given if XOR- or OR-gateways appear.

Definition 7 (Node optionality)**.** *The optionality $o(n)$ of node $n \in N$ is $o(n) = 1$ if n does not have to be executed to finish an instance of the process successfully, and 0 otherwise.*

Both repeatability and optionality values are boolean. As we do not assume any process log information about executed instances as e.g. shown in [25], there is no statement if an optional node is more or less likely to be executed or how often a repeatable node is executed in average. For future work, one can think of also assigning optionality/repeatability values $\in (0, 1)$, e.g., by using execution probabilities or relative frequencies obtained from process execution logs. Analog to Definition 4, repeatability and optionality is extended to sets of nodes as shown in the following definition.

Definition 8 (Repeatability and optionality of node sets)**.** *For $p \in P$, P a partition of G (i.e., $p \subseteq A$), repeatability $\varrho(p)$ and optionality $o(p)$ of a node set p is given through equation (1) by replacing π through ϱ or o, respectively.*

With this, behavioral similarity for the two remaining behavior dimensions can be formulated.

Definition 9 (Behavioral similarity 2 and 3)**.** *For two partitions P_1, P_2 of G_1, G_2 induced by a mapping M, the behavior similarities based on repeatability and optionality is given through equation (2) by replacing π through ϱ or o, respectively.*

C. Penalty Functions

The positional centroids of a task set consisting of "the first" and "the last" task and of a task set consisting of exactly one task in the middle of the model would be the same, when calculated according to formula (2), namely 0.5. But it is quite obvious, that these two sets of nodes are unlikely to match together (regarding their behavior). This is why we introduce penalty terms for every dimension of behavioral similarity, that lower the similarity value if one or both partition elements p and $M(p)$ are heterogenous activity sets. Especially for favouring homogeneity concerning repeatability and optionality in node sets, penalty functions are important. These functions depend on the underlying mapping M and are denoted with $pen^{\pi}_{M}, pen^{\varrho}_{M}, pen^{o}_{M} \geq 0$. They have to be computed for each partition separately. The resulting penalized similarity is of the form $penVSim^{\xi}_{M}(P_1, P_2) = \Big(VSim^{\xi}_{M}(P_1, P_2) - pen^{\xi}_{M}(P_1) - pen^{\xi}_{M}(P_2)\Big)^{+}$, where $\xi \in \{\pi, \varrho, o\}$ and P_1 and P_2 are the partitions induced by M on the two process models G_1 and G_2. We set $penVSim^{\xi}_{M}(G_1, G_2) := penVSim^{\xi}_{M}(P_1, P_2)$.

As $VSim \in [0, 1]$ it is reasonable to demand for penalty functions $pen \in [0, 0.5]$. A function that meets this requirement and that somehow measures the spread of a set of objects is the variance, in this case the sample variance, that uses the centroids as (sample) means. Therefore, if we apply the unbiased sample variance, we get $pen^{\xi}_{M}(p) = \frac{1}{|p|-1}\sum_{a \in p}(\xi(a) - \xi(p))^2$ with $\xi \in \{\pi, \varrho, o\}$ as penalty value for one partition element $p \in P$ with $|p| \geq 2$. For $|p| = 1$ the penalty value is 0 and for $p = \emptyset$ it is not available, i.e., set to $NULL$. The penalty value for a whole partition P is computed as the average over the single penalty values $pen^{\xi}_{M}(P) = \frac{1}{|P|}\sum_{p \in P} pen^{\xi}_{M}(p)$.

D. (Penalized) Behavioral Similarity

To get one value for behavioral similarity, one has to combine the three dimensions of behavior and their corresponding similarity values $VSim^{\pi}$, $VSim^{\varrho}$, and $VSim^{o}$ or, analog, the penalized similarity values $penVSim^{\pi}$, $penVSim^{\varrho}$, and $penVSim^{o}$. This combination can take place with help of a weighted sum of the three values, where the weights can be chosen according to one's own impression of suitability or, which would be worth futher studies, according to statistical findings including model training and parameter estimation, e.g., maximum likelihood methods. With non-negative weights w^{π}, w^{ϱ}, and w^{o} with $w^{\pi} + w^{\varrho} + w^{o} = 1$ the weighted sum, i.e., the behavioral similarity value for two process models G_1 and G_2 under mapping M, is of the form $VSim_M(G_1, G_2) := \sum_{\xi \in \{\pi, \varrho, o\}} \omega^{\xi} VSim^{\xi}_{M}(G_1, G_2)$.

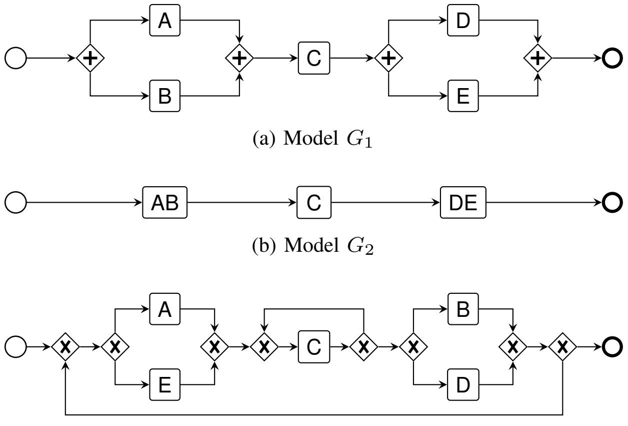
For the penalized behavioral similarity $penVSim_M(G_1, G_2)$, the similarity values for the three behavioral dimensions are replaced by their respective penalized similarity values. Both $VSim$ and $penVSim$ always take values between 0 and 1 where 0 means no similarity and 1 full similarity. The (penalized) behavioral similarity can then again be used for calculating the similarity value including other process model perspectives [11].

IV. Validation

A comparison of three methods to measure behavioral similarity is done in this section. Therefore, three process models G_1, G_2, and G_3, shown in Figure 2, are considered. Models G_1 and G_2 describe the same process, but were modeled by different persons. Model G_3 describes a different process including similar tasks, but with a differing order. In G_3, not all tasks have to be executed and some may be executed several times. Models G_1 and G_2 always have activities A to E executed exactly once. The original label descriptions have been removed and substituted by letters A to E to provide better readability, as the focus lies only on the models' behavior. Information about agents, non-human resources, and data is not shown in the models, either.

For calculating the similarity between Models G_1 and G_2, the *p*artial *i*njective 1-to-1 mapping M^{pi}_1 is established with $\{(A, AB), (C, C), (D, DE)\} = M^{pi}_1$. Tasks B and E from G_1 are not mapped, but results would not differ if B and E instead of A and D would have been mapped. The *b*ijective M-to-N mapping M^{b}_1 according to Definition 2 is established with $\{(\{A, B\}, \{AB\}), (\{C\}, \{C\}), (\{D, E\}, \{DE\})\} = M^{b}_1$. These mappings provide the highest similarity, respectively, when taking into account the activities' descriptions (using string-edit distance). The mappings for the comparison of G_1 and G_3 are the identity functions, namely $M^{pi}_2 = \{(\cdot, \cdot) \mid \cdot \in \{A, B, C, D, E\}\}$ and $M^{b}_2 = \{(\{\cdot\}, \{\cdot\}) \mid \cdot \in \{A, B, C, D, E\}\}$.

For evaluation, three behavioral similarity values are computed for every comparison. One with help of casual footprints (CF) [9], one with smallest casual footprints (smallest CF) as suggested in [3], Section 6.3, IV *discussion*, and one with the



(a) Model G_1

(b) Model G_2

(c) Model G_3

Figure 2: Behavioral similarity values are computed for models G_1–G_2 and G_1–G_3

penalized centroid-based approach as introduced in the present work. The centroid-based method uses M_1^b and M_2^b and the positions (or rather the positional distances) of the respective sets of nodes induced by these mappings. The smallest CF is a modified casual footprint approach with a slightly different, especially simplified, definition. The results of the computation are listed in Table I. The numbers in brackets count the effort of determining the respective similarity values. In particular, the number of computed intermediate values are specified. As the calculation of casual footprints needs another underlying similarity value, called correlation, between the compared nodes, we chose a label-based similarity. For simplicity, it was set $Sim(A, AB) = Sim(D, DE) = 0.5$ and $Sim(\cdot, \cdot) = 1 \; \forall \cdot \in \{A, B, C, D, E\}$.

Obviously, all three methods state that models G_1 and G_2 are more similar than models G_1 and G_3, which is as desired and also assessed by several modeling experts. But differences between the assigned similarity values are substantial. The centroid-based approach states full behavioral similarity between models G_1 and G_2, which could be discussed if this result fits reality, because the first model's parallel gateways seem to be ignored. The casual footprint method instead says similarity is only about 80%, although, as stated above, when finished, all activities A to E are executed exactly once in both models. In contrast, the casual footprint method assigns a similarity of about 64% to G_1 and G_3, although these models describe completely different processes concerning their behavior. The centroid-based approach assigns a relatively low similarity value of about 33%. For both comparisons, the smallest casual footprint approach gives values in between the two other methods. Another even more remarkable difference gets apparent when considering the number of calculated intermediate values (not elementary arithmetical operations). They are shown for all three methods and both comparisons in brackets in Table I. It is apparent, that between the common casual footprint method and the smallest casual footprint approach there is a huge difference in the number of calculated intermediate values, even if the resulting similarity values do not differ that much. For the casual footprint method, the number of intermediate values rises exponentially with the number of model nodes. For the smallest casual footprint approach, this number is only increasing quadratic, which was one of the reasons for the authors of [3] to introduce it. For the centroid-based approach, the number of calculated intermediate values rises linearly with the number of model activity nodes, so the effort is even less, which is a strong point for this method.

TABLE I: Similarity values and number of computed intermediate values (IM).

$Sim.\ (\#IM)$	CF	smallest CF	centroid-based
$Sim(G_1, G_2)$	0.799 (294)	0.885 (90)	1.000 (30)
$Sim(G_1, G_3)$	0.640 (414)	0.632 (108)	0.333 (30)

It should be pointed out again that the centroid-based similarity value gives information about only one aspect of the compared process models. Information about labels, data, and resources is not used for calculating this value. Similarity values concerning these aspects can be calculated separately and then be combined altogether. Instead, the casual footprint method needs a similarity value assigned to each pair of activities which is element of the underlying mapping. Thus, the casual footprint method does not completely separate the different process perspectives orthogonally from each other. However, the results of similarity calculations using the centroid-based method only make sense when combined with other perspective similarities. Otherwise, in the majority of the cases, it would lead to a mapping where first node is mapped to first node, second node to second node, and so on. The behavioral similarity is more like an endorsement of the mappings and similarities obtained for the other perspectives, that also consider the content of the model elements (labels, agents, etc.), which is not the case for the behavioral aspect. This is why a more comprehensive evaluation of the method can only take place together with the other process perspectives, which has to be done nonetheless to fully evaluate the centroid-based approach. As possible evaluation setting, a similar setup as in [9] would probably be a good choice. The draft version for this paper [12] contains a second example for application of the centroid-based similarity measure.

V. Conclusion and Future Work

As shown in Section II, there already exists a variety of techniques for calculating the behavioral similarity of arbitrary process models. The methods presented in the work at hand should not be seen as strictly better, but should rather help in computing *behavioral similarity for M-to-N mappings*, for which behavioral similarity measures applicable on general process models did not exist. A big advantage of the *centroid-based approach* is that average values can *easily be computed*, unlike trace-based and casual footprint methods, and thus, the presented method is suitable even for large practical applications. Furthermore, the idea of splitting process models into several perspectives like label description, data objects, etc., as already pursued in multiple similarity matching papers, is continued in this work by dividing model behavior into the three dimensions *(relative) position, repeatability, and optionality*, which allows for adjusting the weighting of these three dimensions according to the user's needs. The separation of behavior is not done in related work, as far as the authors know. *Penalty terms* are a common means in the field of measure construction, although in the context of process model similarity measuring they have not been used so far. However, the centroid-based similarity measure is not very informative when used on its own. It has to be combined with similarity values for other process perspectives, but this is also the case for, e.g., casual footprints.

Some approaches for future work are already stated in the main part of the paper. Concerning execution specific features of process models, e.g., the optionality value of a node, it is conceivable to use process log information to improve similarity values. Additionally, parameters, weights, and maybe even formulae might be improved by applying machine learning methods on already matched process models. Another big topic for future work would be to implement M-to-N matching methods for all process model perspectives and to run a detailed evaluation that combines all similarity values for the different perspectives. Furthermore, it could be checked if the centroid-based approach provides a real metric, i.e., if it fulfills the corresponding conditions of symmetry, non-negativity, identity, and the triangle inequality. Or, if this is not the case, can it be (easily) adjusted to achieve these properties, as with such metrics it is possible to search huge repositories even faster for similar elements, e.g., with metric trees [22].

ACKNOWLEDGEMENT

The work of Michael Heinrich Baumann is supported by a scholarship of “Hanns-Seidel-Stiftung (HSS)” which is funded by “Bundesministerium für Bildung und Forschung (BMBF)”.

REFERENCES

[1] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management*. Springer, 2013.

[2] M. Dumas, L. García-Bañuelos, and R. M. Dijkman, “Similarity search of business process models,” *IEEE Data Eng. Bull.*, vol. 32, no. 3, pp. 23–28, 2009.

[3] M. Becker and R. Laue, “A comparative survey of business process similarity measures,” *Computers in Industry*, vol. 63, no. 2, pp. 148–67, 2012.

[4] F. Pittke, H. Leopold, J. Mendling, and G. Tamm, “Enabling reuse of process models through the detection of similar process parts,” in *Business Process Management Workshops*, ser. LNBIP, M. La Rosa and P. Soffer, Eds. Springer Berlin Heidelberg, 2013, vol. 132, pp. 586–597.

[5] B. Curtis, M. I. Kellner, and J. Over, “Process modeling,” *Commun. ACM*, vol. 35, no. 9, pp. 75–90, 1992.

[6] M. Weidlich, R. Dijkman, and J. Mendling, “The icop framework: Identification of correspondences between process models,” in *Advanced Information Systems Engineering*, ser. LNCS, B. Pernici, Ed. Springer Berlin Heidelberg, 2010, vol. 6051, pp. 483–498.

[7] Object Management Group, “Case management model and notation version 1.0,” 2014. [Online]. Available: http://www.omg.org/spec/CMMN/1.0/PDF/ [accessed: 2015-07-19]

[8] S. Jablonski and C. Bussler, *Workflow management: modeling concepts, architecture and implementation*. International Thomson Computer Press, 1996.

[9] R. Dijkman, M. Dumas, B. van Dongen, R. Käärik, and J. Mendling, “Similarity of business process models: Metrics and evaluation,” *Information Systems*, vol. 36, no. 2, pp. 498 – 516, 2011.

[10] M. H. Baumann, M. Baumann, S. Schönig, and S. Jablonski, “Towards multi-perspective process model similarity matching,” in *Enterprise and Organizational Modeling and Simulation*, ser. LNBIP, J. Barjis and R. Pergl, Eds. Springer Berlin Heidelberg, 2014, vol. 191, pp. 21–37.

[11] M. Baumann, M. H. Baumann, S. Schönig, and S. Jablonski, “Resource-aware process model similarity matching,” 2014, in press (RMSOC).

[12] M. Baumann, M. H. Baumann, and S. Jablonski, “On behavioral process model similarity matching: A centroid-based approach,” 2015, preprint. [Online]. Available: https://epub.uni-bayreuth.de/id/eprint/2051 [accessed 2015-07-18]

[13] C. Gerth, M. Luckey, J. Küster, and G. Engels, “Detection of semantically equivalent fragments for business process model change management,” in *International Conference on Services Computing (SCC)*. IEEE, 2010, pp. 57–64.

[14] R. Dijkman, M. Dumas, L. García-Bañuelos, and R. Käärik, “Aligning business process models,” in *International Enterprise Distributed Object Computing Conference*. IEEE, 2009, pp. 45–53.

[15] U. Cayoglu *et al.*, “The process model matching contest 2013,” in *4th International Workshop on Process Model Collections: Management and Reuse, PMC-MR*, 2013.

[16] J. Starlinger, B. Brancotte, S. Cohen-Boulakia, and U. Leser, “Similarity search for scientific workflows,” *Proc. VLDB Endow.*, vol. 7, no. 12, pp. 1143–1154, 2014.

[17] R. M. Dijkman *et al.*, “A short survey on process model similarity,” in *Seminal Contributions to Information Systems Engineering*, J. Bubenko, J. Krogstie, O. Pastor, B. Pernici, C. Rolland, and A. Sølvberg, Eds. Springer Berlin Heidelberg, 2013, pp. 421–427.

[18] R. Dijkman, M. Dumas, and L. García-Bañuelos, “Graph matching algorithms for business process model similarity search,” in *Business Process Management*, ser. LNCS, U. Dayal, J. Eder, J. Koehler, and H. A. Reijers, Eds. Springer Berlin Heidelberg, 2009, vol. 5701, pp. 48–63.

[19] C. Klinkmüller, I. Weber, J. Mendling, H. Leopold, and A. Ludwig, “Increasing recall of process model matching by improved activity label matching,” in *Business Process Management*, ser. LNCS, F. Daniel, J. Wang, and B. Weber, Eds. Springer Berlin Heidelberg, 2013, vol. 8094, pp. 211–218.

[20] B. van Dongen, R. Dijkman, and J. Mendling, “Measuring similarity between business process models,” in *Advanced Information Systems Engineering*, ser. LNCS, Z. Bellahsène and M. Léonard, Eds. Springer Berlin Heidelberg, 2008, vol. 5074, pp. 450–464.

[21] BPM-Offensive Berlin, “BPMNPoster,” 2009. [Online]. Available: http://www.bpmb.de/index.php/BPMNPoster [accessed: 2015-07-18]

[22] M. Kunze and M. Weske, “Metric trees for efficient similarity search in large process model repositories,” in *Business Process Management Workshops*, ser. LNBIP, M. zur Muehlen and J. Su, Eds. Springer Berlin Heidelberg, 2011, vol. 66, pp. 535–546.

[23] M. Kunze, M. Weidlich, and M. Weske, “m3–a behavioral similarity metric for business processes,” in *ZEUS*, ser. CEUR-WS, 2011, pp. 89–95.

[24] A. Wombacher, “Evaluation of technical measures for workflow similarity based on a pilot study,” in *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE*, ser. LNCS, R. Meersman and Z. Tari, Eds. Springer Berlin Heidelberg, 2006, vol. 4275, pp. 255–272.

[25] W. van der Aalst, A. de Medeiros, and A. Weijters, “Process equivalence: Comparing two process models based on observed behavior,” in *Business Process Management*, ser. LNCS, S. Dustdar, J. Fiadeiro, and A. Sheth, Eds. Springer Berlin Heidelberg, 2006, vol. 4102, pp. 129–144.

[26] S. Abiteboul, P. Bourhis, and V. Vianu, “Comparing workflow specification languages: A matter of views,” *ACM Trans. Database Syst.*, vol. 37, no. 2, pp. 10:1–10:59, 2012.

[27] A. Polyvyanyy, M. Weidlich, and M. Weske, “Isotactics as a foundation for alignment and abstraction of behavioral models,” in *Business Process Management*, ser. LNCS, A. Barros, A. Gal, and E. Kindler, Eds. Springer Berlin Heidelberg, 2012, vol. 7481, pp. 335–351.