2011

On Measures of Behavioral Distance between Business Processes

Joerg Becker
University of Munster, becker@ercis.de

Philipp Bergener
University of Münster, bergener@ercis.de

Dominic Breuker
University of Münster, breuker@ercis.de

Michael Räckers
University of Münster, raeckers@ercis.de

On Measures of Behavioral Distance between Business Processes

Jörg Becker
University of Münster
Leonardo-Campus 3
D-48149 Münster
+4902518338100
becker@ercis.de

Philipp Bergener
University of Münster
Leonardo-Campus 3
D-48149 Münster
+4902518338067
bergener@ercis.de

Dominic Breuker
University of Münster
Leonardo-Campus 3
D-48149 Münster
+4902518338062
breuker@ercis.de

Michael Räckers
University of Münster
Leonardo-Campus 3
D-48149 Münster
+4902518338075
raeckers@ercis.de

ABSTRACT
The desire to compute similarities or distances between business processes arises in numerous situations such as when comparing business processes with reference models or when integrating business processes. The objective of this paper is to develop an approach for measuring the distance between Business Processes Models (BPM) based on the behavior of the business process only while abstracting from any structural aspects of the actual model. Furthermore, the measure allows for assigning more weight to parts of a process which are executed more frequently and can thus be considered as more important. This is achieved by defining a probability distribution on the behavior allowing the computation of distance metrics from the field of statistics.

Keywords
Business Process Modeling, Process Similarity Measurement, Business Process Distance Measurement, Behavioral Process Similarity

1. INTRODUCTION
Business Process Models (BPMs) enable organizations in Public and Private Sector to get a transparent overview over the relevant extracts of their organization. BPMs are used to gain clarity about the logical sequence of activities in an organization. They are also applied to describe the resulting products and services, the required resources and data, as well as the involved organizational units. They are discussed in Information Systems (IS) literature as a tool to evaluate the overall performance of an organization [1] and to support business process reorganization and optimization by both capturing the as-is situation and designing the to-be process.

10th International Conference on Wirtschaftsinformatik,
16th - 18th February 2011, Zurich, Switzerland

The comparison of business processes with the help of a similarity measure is often an important component in approaches supporting business process management. Examples are the integration of business processes in scenarios of distributed modeling, the identification of similar processes in a huge set of company process models, e.g. to leverage synergy effects, or the benchmarking of processes between organizations [2]. Furthermore, it can be applied to control reorganization projects by comparing to-be and implemented as-is processes or in the context of process mining [3], where the actual behavior of a business process is compared with process models or certain business rules to check for process compliance [4].

The contribution of this paper is to apply distance measure approaches from statistics on the field of business process management. Though a distance measure can be of use in many different contexts, an application scenario in which our approach is of particular interest is the comparison of designed to-be process models with their actual implementations to check for conformance.

In our approach, we introduce a set of related distance measures for business processes. There are two basic characteristics underlying these distance measures. First, it aims at measuring the distance with respect to the behavioral aspects of the business processes. Aspects regarding the modeling language employed to represent the process and the constructs defining this behavior shall be excluded from consideration. This is especially applicable in situations where the information about processes is taken from log files, e.g. to check for conformance or compliance. Second, it takes into account the frequency of the observed behavior, i.e. the number of executions for activities. This allows weighing the important (more frequent) parts of the process stronger than the unimportant ones.

These goals are achieved by taking a probabilistic perspective on the behavior of the process. All the different sequences of activities that may be observed are extracted from the process together with the corresponding probabilities. This delivers a probability distribution on these sequences. Then, using a well-known distance measure from the field of statistics that is based on the so called Bhattacharyya coefficient, our notions of distance between business processes are defined and illustrated by examples.

Often, measures of distance and similarity can be used interchangeably as high distance means low similarity. In our case, the distance will lie between zero and one, making the

question of whether we propose distance or similarity measures a matter of definition. Since the measure we use is often defined as a distance we stay with this convention.

The remainder of this work proceeds as follows. Section two discusses different notions of similarity between business processes by providing an overview of the related work. Section three then introduces our behavioral representation of business processes and derives a probability distribution over its behavior. Following that, section four defines our distance measures and provides examples on how to compute them. Section five then illustrates the defined measures by applying them to an example. Finally, section six concludes and gives an outlook in future research.

2. NOTIONS OF SIMILARITY - RELATED WORK

The first problem arising in business process similarity calculation is matching of elements in different models. Usually, this is based on the labels assigned to the elements, which is why this problem is often referred to as label matching [5]. Due to the use of natural language in these labels, matching them is by no means a trivial task. Simple methods like computing the distances between strings can be employed here. A good overview on such methods is given in [6]. More advanced techniques employ for instance machine learning algorithms or lexical systems such as WordNet [7] to identify higher level relation between words used in the labels. A good survey on approaches to this problem can be found in [8]. In this work, we take the label matching as given. Any of the above mentioned methods could be combined with our approach.

Once a matching of elements is achieved, the computation of business process similarity can be done with respect to two different aspects. On the one hand, the focus can be laid on the actual graphs by which the business processes are represented. This is called the structural aspect of similarity. On the other hand, one can abstract from the particularities of the graphs and restrict the comparison to the interplay of the activities performed in the processes. This is called the behavioral aspect of similarity.

Approaches on structural similarity naturally lead to the well-known field of graph matching, which has a longstanding tradition in computer science [9]. An important concept in that research area is the edit distance of two graphs. It is defined as the cost of transforming one graph into the other by means of elementary change operations like inserting, substituting or deleting nodes. Applications of this concept to the area of business processes can be found for instance in [10, 11].

However, not all approaches focusing on structural aspects use graph matching techniques. In [12], so called features are extracted from the business processes under consideration. Based on the arising feature space, a similarity metric is being derived. In [13] a similarity flooding algorithm is applied to match one process graph onto the other.

One characteristic of these purely structural measures of similarity is that they can identify differences between models even if they describe exactly the same behavior, which may be wanted or unwanted depending on the context of use. Nevertheless, approaches using structurally inspired techniques can also address behavioral aspects. This can, for example, be achieved by building a graph of the process behavior in such a way that ideas from traditional graph matching like edit distances can then be applied to these representations [14, 15].

When viewing business processes – in contrast to the approaches described above – from an entirely behavioral point of view, one is interested in the sequences of activities that a particular business process allows. A widely known approach addressing this aspect is based on a causal footprint representation of the process behavior [16, 17]. This is a graph capturing the possible ordering relations, which means that it specifies which activities can follow on each other and which cannot. The similarity of two processes is then calculated by embedding the causal footprint into a vector space and computing the cosine of the angle between these vectors. A comparable representation of the process behavior is used in [18], where a matrix of so called transition adjacency relations is build. It contains one if a transition can be observed directly after the other and zero if not. The similarity of the behavior is then measured by the similarity of these matrices.

Other approaches taking a behavioral view utilize another traditional field of computer science, namely automata theory [9]. In this area, automata are used to describe languages consisting of words over an alphabet of symbols. Applied to BPM, the behavior of a business process can be understood as a set of activity sequences.

A fundamental concept to compare automata is that of bisimulation [19], which effectively means that, when two automata are bisimilar, their behavior cannot be distinguished by an external observer. Many different notions of bisimulation have been developed over time, but most of them deliver binary yes/no answers only. However, methods for computing the similarity of automata have also been developed that can be interpreted as fuzzy versions of bisimulation, measuring the degree to which the relation holds. See [20, 21] for examples as well as [22] for an application to workflow modeling.

Instead of comparing the automata of languages, one can directly compare the languages themselves, i.e. the sets of possible words [23, 24]. This again involves a notion of edit distance, but this time between languages. In a very rigid case, the distance between two languages is defined as the lowest distance between any of their words. In the context of business processes, this would already result in 100% similarity if there is a single activity sequence shared by the models. To relax this, probabilities can be assigned to each of the words of a language, in which case the comparison can be based on all words of a language, weighted by their probabilities.

The introduction of probabilities assigned to words is, in a sense, closely related to an approach of business process similarity calculation that is, in contrast to any other approach discussed so far, based on observed instances of business processes [25]. The aim of this method is to explicitly address frequent aspects of a business process stronger than infrequent ones. In contrast to our work, it computes two one-sided measures of similarity, called behavioral precision and recall. They measure how well the behavior of one process fits to the other and vice versa.

3. PROBABILITY-WEIGHTED LABELED TRANSITION SYSTEMS

As a model of the business process behavior, we will use a labeled transition system [26] equipped with probabilities on the transitions. We will call the model a probability-weighted labeled transition system (PLTS). Any business process that is supplemented with probabilities on the paths between activities could be transformed into such a representation. The advantage of the PLTS is that the possible paths through the process and the probability of taking it can easily be seen.

A probability-weighted labeled transition system shall be defined as the 6-tuple $PLTS = (S, T, \Sigma, s_o, S_F, p)$ with

- S being a finite set of states
- $T \in (S \setminus S_F) \times \Sigma \times S$ being a finite set of transitions between states emitting an activity label $x \in \Sigma$
- Σ being a finite set of activity labels
- s_o being the unique initial state
- S_F being the set of final states which cannot be left by a transition
- $p: T \to]0,1]$ being a function assigning a positive probability to each of the transitions.

The probability weighting function p is defined such that, for each state s, the sum of the probabilities of transitions leaving this state sums to one:

$$\forall s \in S: \sum_{t=s \xrightarrow{x} \acute{s} \in T} p(t) = 1 \quad .$$

Here, $s \xrightarrow{x} \acute{s}$ denotes the transition $t = (s, x, \acute{s})$. Given this definition of a PLTS, we can define a path of length $n \in \mathbb{N}$ through it, which shall be an n-tuple of transitions $t^{(n)} = \left(s_0 \xrightarrow{x_1} s_1, s_1 \xrightarrow{x_2} s_2, \dots, s_{n-1} \xrightarrow{x_n} s_n\right)$ with $s_n \in S_F$. The path starts at the initial state, wanders through the PLTS and ends in one of the final states.

A path of length n gives rise to a sequence of activity labels $X = (x_1, \dots, x_n) \in \Sigma^n$ having length n by reducing the path to the activity labels the transitions emit. The enumerated behavior can then be defined as the set $EB(PLTS)$ of all possible activity sequences:

$$X = (x_1, \dots, x_n) \in EB(PLTS) \Leftrightarrow$$
$$\exists t^{(n)} = \left(s_0 \xrightarrow{x_1} s_1, s_1 \xrightarrow{x_2} s_2, \dots, s_{n-1} \xrightarrow{x_n} s_n\right) \in T^n : s_n \in S_F$$

This means that the enumerated behavior of a PLTS consists of all the activity sequences that can arise from taking any path through the PLTS. Note that the set can be infinite in the case that the PLTS has loops.

The probabilities $p(t)$ assigned to the transitions t induce an assignment of probabilities $\hat{p}(t^n)$ to paths. It is defined as the product of all the probabilities of the transitions that belong to this path.

$$\hat{p}\left(t^{(n)}\right) = \prod_{i=1}^{n} p\left(s_{i-1} \xrightarrow{x_i} s_i\right)$$

This assignment of probabilities actually induces a probability distribution over paths, as one can see from the following inductive argument. Assume the simplest PLTS is given, consisting of only the initial state s_o and a set of final states S_1. As the transitions are arbitrary, this PLTS can have any number of paths with length one. All these paths will leave s_o and enter one of the states S_1. Since the probabilities for all transitions leaving s_o must sum to one, they define a distribution over all possible paths. Now assume that, for any PLTS having paths of at most length n, a probability distribution over the paths is induced. Then, by adding transitions from the final states of this model to new states, any PLTS can be created that has a path length of $n + 1$. All paths previously having length n and now having length $n + 1$ will be multiplied by the respective probability of an

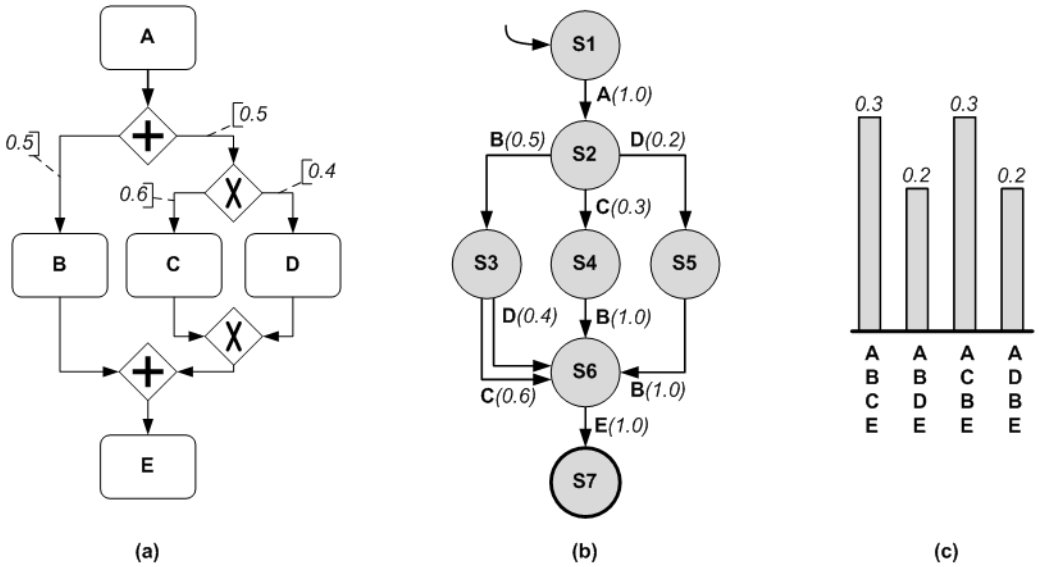

Figure 1: (a) defines a business process in BPMN notation, (b) represents the same process as a PLTS, (c) illustrates the corresponding distribution over activity sequences.

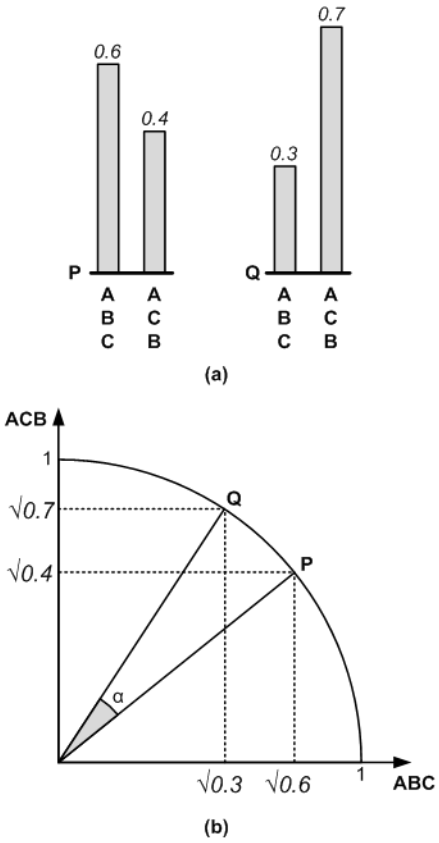

Figure 2: (a) defines two distributions over activity sequences, (b) represents the Bhattacharyya coefficient as the angle between the distributions projected onto a circle.

additional transition to a new state, which sum, for each of the final states of the model with maximum length *n*, to one. Thus, when summing over all the paths, the sum will again turn out to be one.

Finally, given the probability distribution $\hat{p}(t^{(n)})$ over paths, we can define a probability distribution $P(X)$ over the set Σ^* of all possible activity sequences of any length. This is easily accomplished by assigning to each sequence X the sum of all probabilities of paths that emit exactly the activity symbols of this sequence:

$$P: \Sigma^* \to [0,1], with\ P(X) = P(x_1, \dots, x_n) = \sum_{t^* emitting\ X} \hat{p}(t^*)$$

Any other of the infinitely many paths in Σ^* is assigned zero probability. This distribution shall be the probabilistic behavior of a business process.

As an example, consider the business process in figure 1 (a) and (b), giving rise to the distribution shown in figure 1 (c).

4. MEASURES OF BEHAVIORAL SIMILARITY

4.1 The Bhattacharyya Coefficient

The main idea of our behavioral similarity measures will be to measure the difference of distributions over activity sequences as defined in the previous chapter. In statistics, there are numerous different notions of distance between distributions that may be used for this purpose [27]. In our case, two requirements need to be fulfilled. First, we do not want the distance to depend only on extreme values of the distributions such as the maximum distance between the probabilities of two corresponding activity sequences. Rather, all of the sequences shall be taken into account. Second, it must be possible to compare zero probability activity sequences since there will most certainly be sequences in one business process that are completely impossible in others. However, many popular distance measures on distributions, like the Kullback-Leibler divergence or the χ^2 distance, do not handle such singularities.

Having in mind these two requirements, the Bhattacharyya coefficient seems to be a reasonable choice [28]. It is a quantity that measures the similarity of two distributions, i.e. is assumes values between one and zero with one if the distributions are equal. The definition of the Bhattacharyya coefficient is as follows

$$\rho = \sum_{X \in \Sigma^*} \sqrt{P(X) \cdot Q(X)}$$

with P and Q being the distributions to compare. While the summation here is over all of the infinitely many activity sequences of any length, the actual computation has to be done only for sequences to which a positive probability is assigned by both of the distributions, which is unproblematic if this set is finite. The case of infinite sets due to loops will be dealt with later.

The Bhattacharyya coefficient has a straight forward geometric interpretation [29]. Consider a space over $\mathbb{R}$ having a dimension for each of the possibly observable activity sequences $X \in \Sigma^*$. Also assume that there are two distributions P and Q over these activity sequences, assigning probabilities $P(X_1), P(X_2), \dots, P(X_n)$ and $Q(X_1), Q(X_2), \dots, Q(X_n)$ to all the activity sequences. Note that the probabilities are allowed to be zero. Here, $\{X_1, X_2, \dots, X_n\}$ shall be the set of sequences to which at least one distribution assigns a positive probability. Within the space, one can interpret the vectors $\left(\sqrt{P(X_1)}, \sqrt{P(X_2)}, \dots, \sqrt{P(X_n)}\right)$ and $\left(\sqrt{Q(X_1)}, \sqrt{Q(X_2)}, \dots, \sqrt{Q(X_n)}\right)$ as representations of the distributions. Since the vectors contain the square roots of the probabilities and the distributions sum up to one, the vectors will always lie on the unit hypersphere. The Bhattacharyya coefficient can now be interpreted as the cosine of the angle between the two vectors corresponding to the distributions.

For cases in which only two different possible activity sequences are observable, a graphical representation like the one in figure 2 can be given. Here, two distributions P and Q over two activity sequences ABC and ACB respectively are given, as illustrated figure 2 (a). Then, a space having one dimension for each of the activity sequences is given in figure 2 (b) and the relevant part of the unit circle is drawn. As it can be seen, the vectors $\left(\sqrt{0.6}, \sqrt{0.4}\right)$ and $\left(\sqrt{0.3}, \sqrt{0.7}\right)$ lie on that circle. The Bhattacharyya coefficient then calculates to be roughly 0.95, which is the cosine of the angle $\alpha \approx 17{,}56°$ between the vectors.

Bearing in mind this geometrical interpretation, one can reason easily about extreme cases. If two distributions are identical, they will be assigned to exactly the same point, making the angle between them be equal to zero. Thus, the Bhattacharyya

coefficient will be equal to $\cos(0) = 1$, expressing the intuition that the distributions are 100% similar. Contrary, when the one distribution distributes its probability mass on only those activity sequences the other distributions assigns zero probability to, the two vectors will be perpendicular to each other and the coefficient calculates to $\cos(0.5 \cdot \pi) = 0$, expressing that the distributions are entirely different.

Based on the Bhattacharyya coefficient, a distance measure on distributions could be defined as $d(P,Q) = -\ln(\rho(P,Q))$ [30]. It follows from the properties of the Bhattacharyya coefficient that this distance satisfies several desirable properties. Those are:

- $d(P,Q) \geq 0$ *non-negativity*
- $d(P,Q) = d(Q,P)$ *symmetry*
- $d(P,Q) = 0 \Leftrightarrow P = Q$ *identity*

for any choice of distributions *P* and *Q*. However, there is a fourth property this quantity does not satisfy which can be of advantage in various applications. This property is:

- $d(P,R) + d(R,Q) \geq d(P,Q)$ *triangle inequality*

for any choice of distributions *P*,*Q* and *R*. A distance measure fulfilling the triangle inequality is called a distance metric [31]. The important difference of such a metric as compared to non-metric quantities is that it allows sorting the entities being compared by it in a consistent way. When entities are compared by a non-metric distance measure, one could pick an arbitrary reference entity, compare it to all other entities and then sort the entities for instance with increasing distance to the reference entity. The problem is that, for a different reference entity, a new sorting has to be computed separately, whereas a distance metric allows embedding the entities in a metric space in such a way that the distance of entities in that space is consistent for any arbitrary reference point. This allows, for example, the use of powerful search algorithms [32].

Consider for instance the artificial example given in figure 3. For an arbitrary non-metric measure used to sort the distributions in figure 3 (a), a different sorting has to be created for each of the two reference distributions *P* and *Q*. In particular, one can only reason about relations of distributions to the reference distribution. Knowing the distance of *P* to *R* and *P* to *T* implies nothing about the distance of *R* to *T*. It could happen that the total distance of *P* to *T* is actually higher than the sum of the distances of *P* to *R* and *R* to *T*, which is counterintuitive. In figure 3 (b), a distance metric was used such that the distributions can be embedded into a two-dimensional space. In that case it is easy to see that the distance of *P* to *T* cannot be bigger than the sum of the distances *P* to *R* and *R* to *T*, which is due to the triangle inequation.

Luckily, a small modification to the Bhattacharyya coefficient gives rise to a quantity satisfying the triangle inequality [33]. We define

$$\hat{d}(P,Q) = \sqrt{1 - \rho(P,Q)}$$

to be the Bhattacharyya distance metric on distributions.

4.2 Strict Match Distance Measures

In this section, we will define our first two distance measures on business process. We name them the strict match measures since they treat any activity sequences arising from the business processes as being completely different when only a single discrepancy is found. For instance, the sequences ABC and ACB

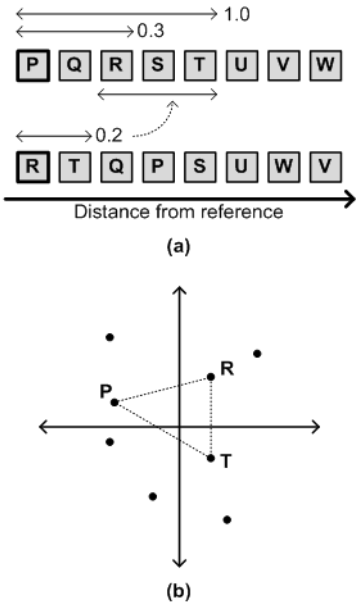

Figure 3: (a) represents two orderings of distributions with respect to a non-metric distance measure (b) represents an embedding of distributions into a space with respect to a metric distance measure.

are treated as being different and the sequences ABC and CBA are treated as being equally different. No distinction based on the similarity of the sequences is made.

We can now define the strict match distance of two business processes BP_1 and BP_2 by calculating the Bhattacharyya distance metric between the distributions on activity sequences $P(X_1)$ and $P(X_2)$ of those two processes to be

$$dist^{strict}_{complete}(BP_1, BP_2) = \sqrt{1 - \rho\big(P(X_1), P(X_2)\big)} \ .$$

This distance metric should be used whenever small discrepancies between rather similar processes shall be measured. The order in which the activities are performed should be critical for the processes as differences in this order are strongly penalized by this distance metric.

As an example, consider the two business processes in figure 4 given as PLTSs. They only have two activity sequences, namely ABCE and ACBE in common. All other activity sequences have probability zero in one of the processes. Furthermore, the probability of the sequence ACBE being observed in the first process differs from that of the second. The Bhattacharyya coefficient computes to $\sqrt{0.3 \cdot 0.3} + \sqrt{0.3 \cdot 0.1} = 0.47$. Thus, the above defined distance metric in this example is equal to $\sqrt{1 - 0.47} = 0.73$.

For some applications, one might not be interested into the behavioral distance with regard to the entire behavior of two processes but rather with regard to the overlaps that exist between the two. In the example of figure 4, the first process contains activity D, while the second does not, and the second contains activity F which is not found in the first process. In such cases, the distance measure can be computed in a slightly different way. Any transition emitting a symbol that is specific to only one the processes is then switched to a "silent mode" which means that it still belongs to the path but its symbol does not appear in the

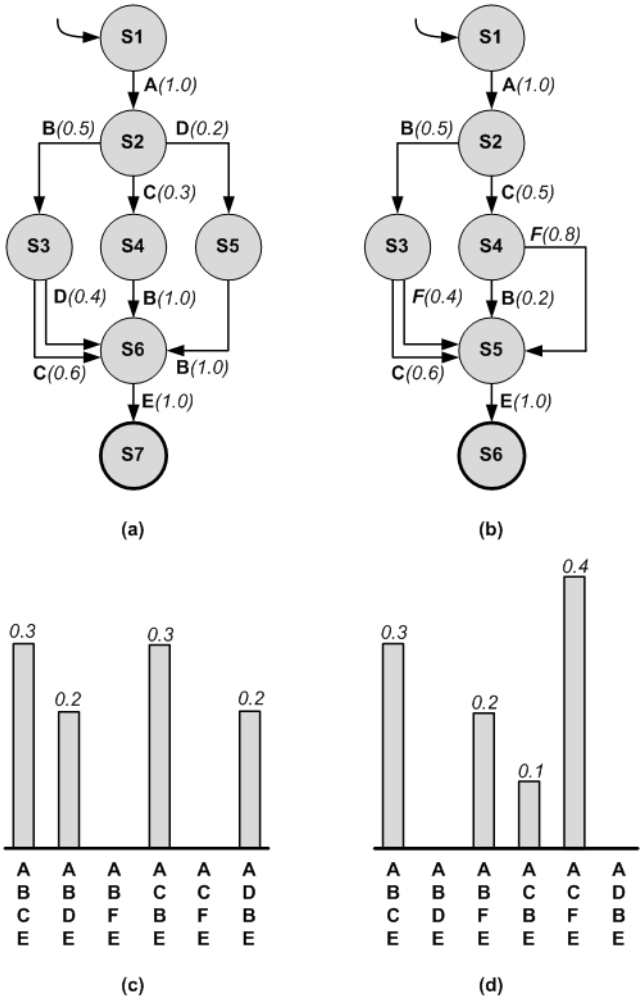

Figure 4: (a) and (b) represent two business processes as PLTSs, (c) and (d) represent the corresponding distributions over activity sequences.

activity sequences anymore. This ensures that all activity sequences only contain activities common to both processes.

We define the rigid distance of the overlap of two business processes BP_1 and BP_2 as

$$dist_{overlap}^{strict}(BP_1, BP_2) = \sqrt{1 - \rho\left(P(\hat{X}_1), P(\hat{X}_2)\right)}$$

where $\hat{X}_1$ and $\hat{X}_2$ are the activity sequences observed when the activities unique to either of the processes are silent.

Figure 5 illustrates this concept using the previous example. The grey shaded transition symbols D and F denote that these transitions are currently silent. Thus, the activity sequences ABDE and ADBE both merge to the single sequence ABE and the sequence ACFE reduces to ACE. The Bhattacharyya coefficient in this example is equal to $\sqrt{0.3 \cdot 0.3} + \sqrt{0.4 \cdot 0.2} + \sqrt{0.3 \cdot 0.1} = 0.76$ giving a distance of $\sqrt{1 - 0.76} = 0.49$.

4.3 Fuzzy Match Distance Measures

The following section is devoted to more relaxed measures of distance. The main difference from the strict match ones is that we will abstain from the assumption that all activity sequences having small differences are already treated as completely different. Rather, we will use the similarity of these sequences to identify them with each other, thereby introducing new notions of distance being more appropriate for application scenarios in which small differences in the order of the activities or the exclusion of some of the activities of a sequence should result in small distances of the behavior.

To derive these measures, consider again the example of figure 4. Both processes have two activity sequences ABCE and ACBE in common. Since we can directly identify them with each other, not special treatment is necessary. The activity sequences ABDE and ADBE however are unique to the first process, but we are now interested in whether we can associate them with similar sequences of the other process.

In general, we can quantify our belief that a particular activity sequence being unique to one business process belongs to any sequence of the other process by calculating any kind of string similarity between the sequences. Usually, these string similarities are based on calculating the minimum number of elementary operations required for transforming one string into the other and summing up the costs of all the operations [6]. The basic operations vary among the algorithms, but possible operations are

- Insertions: Insert one symbol into the string
- Deletions: Remove one symbol from the string
- Substitutions: Replace one symbol with another
- Transpositions: Swap two symbols with each other

In the remainder of this work, we will use the popular Levenshtein distance to compute a similarity of two activity sequences, which uses insertions, deletions and substitutions. However, other choices might be suitable as well. For instance, the Damerau-Levenshtein distance [34], adding transpositions to the set of operations, could be used with the effect that variations in the order of activities would result in less cost. The activity sequence ABC requires two substitutions to be transformed into the sequence ACB, but only one transposition.

To transform the Levenshtein distance $l(X_1, X_2)$ of two activity sequences X_1 and X_2 into a similarity measure, we simply normalize the distance by the maximum distance that could be observed between the sequences, which is $\max(|X_1|, |X_2|)$ with $|X_1|$ and $|X_2|$ being the lengths of the sequences. Thus, we define the sequence similarity to be

$$sim(X_1, X_2) = 1 - \frac{l(X_1, X_2)}{\max(|X_1|, |X_2|)}$$

This definition results, for the example of the sequence ABDE, in a similarity of 0.75 compared to the sequence ABFE and 0.5 compared to ACFE.

The idea of creating the associations among the unique sequences is very simple. First, the similarity of each pair of unique activity sequences is computed where the first sequences stems from one process and the second sequence from the other. Second, associations *R* between activity sequences are created in a greedy way, creating an association between those sequences having the highest similarity. In the example, the first association to be created is the one between sequences ABDE of the first process and ABFE of the second process, leaving the association between ADBE and ACFE as the only possibility for the second association. This procedure is illustrated in figure 6.

In the general case, we will end up with the following relation:

$$R \subset \Sigma^*_{u1} \times \Sigma^*_{u2}$$
$$with\ (X_1, X_2) \in R \wedge (X_1', X_2') \in R \Rightarrow X_1 \neq X_1' \wedge X_2 \neq X_2'$$

where Σ^*_{u1} and Σu^*_2 denote the sets of unique activity sequences in the first and second process respectively. The condition ensures that no activity sequence is mapped to more than one other sequence.

The relation enables us to define a modified version of the Bhattacharyya coefficient which computes the similarity of the distributions according to the created associations but correcting for the dissimilarities of the associated sequences. It shall be

$$\dot{\rho} = \sum_{X \in X_c} \sqrt{P(X) \cdot Q(X)} + \sum_{(X_1, X_2) \in R} sim(X_1, X_2) \cdot \sqrt{P(X_1) \cdot Q(X_2)}$$

where X_c denotes the set of activity sequences common to both processes.

It is then easy to define the fuzzy match distance metrics based on this modified Bhattacharyya coefficient, the first of which is:

$$dist^{fuzzy}_{complete}(BP_1, BP_2) = \sqrt{1 - \dot{\rho}\left(P(X_1), P(X_2)\right)}\ .$$

For the example, the Bhattacharyya coefficient computes to $0.5 \cdot \sqrt{0.2 \cdot 0.4} + \sqrt{0.3 \cdot 0.1} + 0.75 \cdot \sqrt{0.2 \cdot 0.2} + \sqrt{0.3 \cdot 0.3} = 0.76$, which gives a fuzzy distance of $\sqrt{1 - 0.76} = 0.49$.

In a fashion similar to the previous section, we can also define a fuzzy match distance metric that removes all activities unique to one of the processes from the activity sequences. The associations are then created based on this already reduced distribution over sequences. For the sake of completeness, we define it to be

$$dist^{fuzzy}_{overlap}(BP_1, BP_2) = \sqrt{1 - \dot{\rho}\left(P(\hat{X}_1), P(\hat{X}_2)\right)}\ .$$

In our example, no difference can be observed compared to the rigid case. This is due to the fact that there is only one unique sequence, namely ACE in the first process. As there is no other sequence to assign it to, the relation remains empty and $\dot{\rho}$ assumes the same value as ρ, resulting in equal distances.

4.4 Distances in Presence of Loops

Although the distance measures presented in this paper are defined for the case that there are infinitely many activity sequences that can possibly be observed, the actual computation will be infeasible in such cases. This problem arises in all business processes having loops, as for example in the process represented in figure 7 in which the sequence BD may be executed arbitrarily often.

While this is a well-known problem of all approaches to similarity measurement that rest upon activity sequences, our current setting allows circumventing this problem. Since any decision on entering or not entering a loop is weighted by a certain probability, sequences with more loop iterations tend to become more and more unlikely. In the example of figure 7, the probability of the sequence having one iteration of the loop still

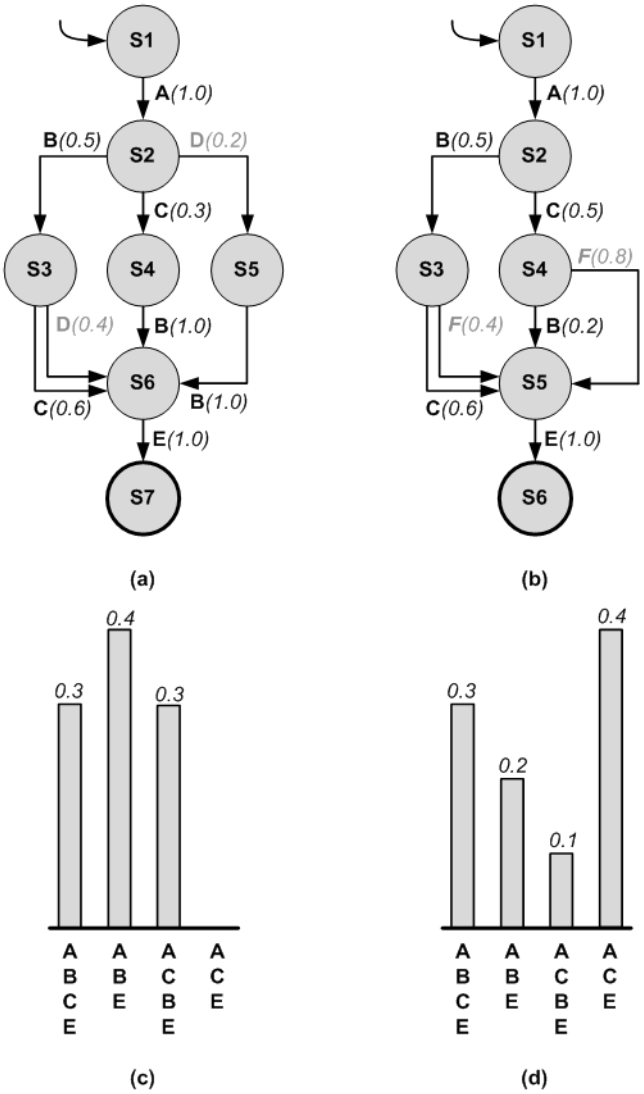

Figure 5: (a) and (b) represent two business processes as PLTSs, (c) and (d) represent the corresponding distributions over activity sequences where activities unique to one of the processes are silent.

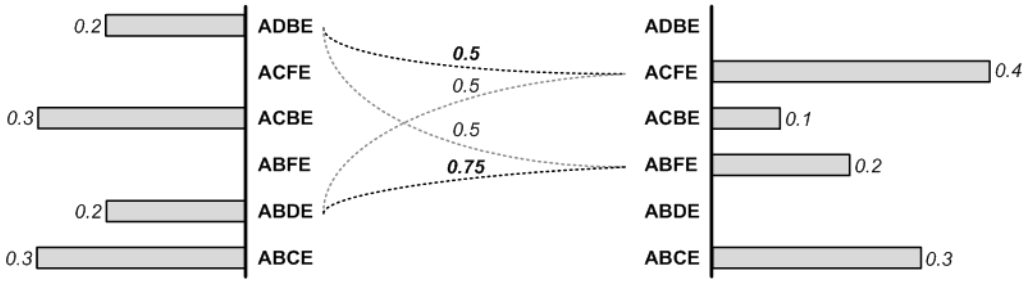

Figure 6: similarities of unique activity sequences, with dominant relations being highlighted.

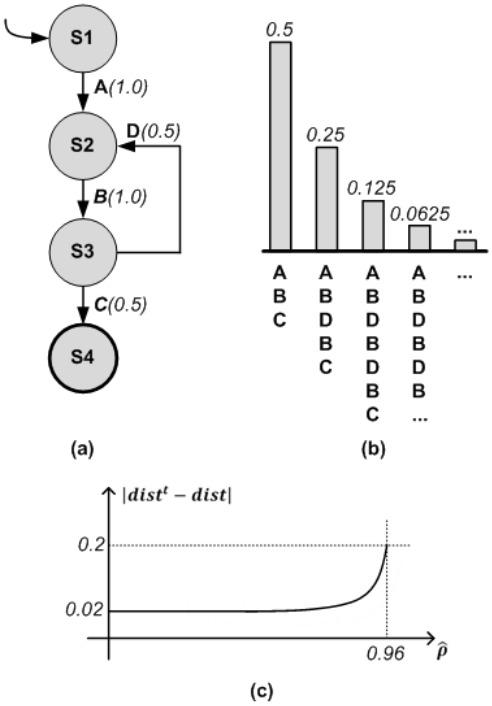

Figure 7: (a) represents business process as PLTS having a loop, (b) represents the corresponding infinite distribution, (c) is a graph of the maximal error of a distance at truncation level 0.96.

amounts for 0.25, while for three iterations it is already down to 0.0625. Since unlikely sequences are of low relevance for the value of the Bhattacharyya coefficient, the remedy to the problem is to just truncate the sum when a certain amount of probability mass is covered.

To formalize this, we first need to define the truncation level *t*. It shall be the amount of probability mass we require to use in the computation of the Bhattacharyya coefficient. Thus, it should assume a value close to one. The set of activity sequences required to fulfill this level shall be named X_t. Thus, the approximated Bhattacharyya coefficient is

$$\hat{\rho} = \sum_{X \in X_t} \sqrt{P(X) \cdot Q(X)} + error \ .$$

In the worst case, the error term comprises only one missing activity sequence X_m to which both processes assign probability $1 - t$. Hence, the error is bounded by *t*:

$$0 \leq error \leq \sqrt{P(X_m) \cdot Q(X_m)} \leq \sqrt{(1-t) \cdot (1-t)} = 1 - t \ .$$

Now let $dist$ be the true distance to compute and $dist^t$ be the truncated approximation. Then we get

$$|dist^t - dist| = \left|\sqrt{1-\hat{\rho}} - \sqrt{1-\hat{\rho}-error}\right|$$

$$= \sqrt{1-\hat{\rho}} - \sqrt{1-\hat{\rho}-error} \leq \sqrt{1-\hat{\rho}} - \sqrt{1-\hat{\rho}-(1-t)} \ .$$

This upper bound on the deviation of true and truncated distance is maximal for cases in which $\hat{\rho}$ is maximal. As $\hat{\rho}$ is bounded above by t, the upper bound on the deviation is:

$$|dist^t - dist| \leq \sqrt{1-\hat{\rho}} - \sqrt{1-\hat{\rho}-(1-t)} \leq \sqrt{1-t} \ .$$

How the upper bound on the deviation develops with respect to the approximated Bhattacharyya coefficient can be seen in figure 7 (c), where the example of a truncation level $t = 0.04$ is illustrated. As one can see, for the maximum Bhattacharyya coefficient $\hat{\rho} = 0.96$, the deviation is maximal. After that, it rapidly decays due to the square root used in computing the distance.

5. APPLICATION EXAMPLE

To further illustrate the definitions we have made in the previous chapter, consider the rather simple set of business processes shown in figure 8 (a). It contains the processes P3 to P8 in form of PLTSs. Processes P1 and P2 shall be the ones already known from the last chapter. They can be seen in figure 4 (a) and (b) respectively. We now want to analyze this set of processes to investigate how the similarity measures perform on this example.

First, we take a look at the processes themselves to get a broad overview of their characteristics. All of them define activity sequences over the activities A,B,C,D,E,F,G,H, and I, but not all of them include the entire set of activities. Two of the eight processes, namely P3 and P6, define infinite sequences since they contain loops. Therefore, an approximate similarity calculation will be necessary. We used a truncation level of $t = 0.99999$ for all our calculations in this section, giving a maximum deviation from the true distance equal to 0.01.

One directly sees from figure 8 (a) that some processes seem to be very similar. For instance, processes P7 and P8 look identical on the first sight. A closer look, however, reveals that the left branch is much more likely to be taken in P7 than in P8 and vice versa. Also the models P5 and P6 seem to be quite similar, as the only difference between P6 as compared to P5 are the two additional loops. Also model P4 defines behavior that is very similar to P5 and even P6. Process P3 on the other hand does not have much in common with the other processes in the set.

We now want to use the distance metrics to derive and visualize the thoughts we have just made. For the first analysis we choose the metric $dist^{fuzzy}_{complete}$ for comparison and compute a complete distance matrix for the set of business processes. This information on how close the processes are can then be used to represent this distance graphically. The result of this computation is shown in figure 8 (b). It represents the business processes as points in a two-dimensional space. All the points are fitted into this space such that distances between them reflect the distances of the processes they represent. The technique we have used to create this picture is called Multidimensional Scaling (MDS). From this picture, one can easily see groups of similar processes being close to each other. The groups we would suggest based on the picture are indicated by dashed circles around the points.

Another analysis that can be performed on the models is to cluster them with respect to their distance. For this experiment, we have chosen the metric $dist^{strict}_{overlap}$. We then applied an agglomerative clustering algorithm to the distance matrix computed from that metric to find possible clusters of processes. The dendrogram visualizing the results can be seen in figure 8 (c). When compared to the results of the MDS analysis, one can see that, although a simpler metric was used that does not identify similar activity sequences with each other, the result is rather similar. The same groups are suggested to the analyst.

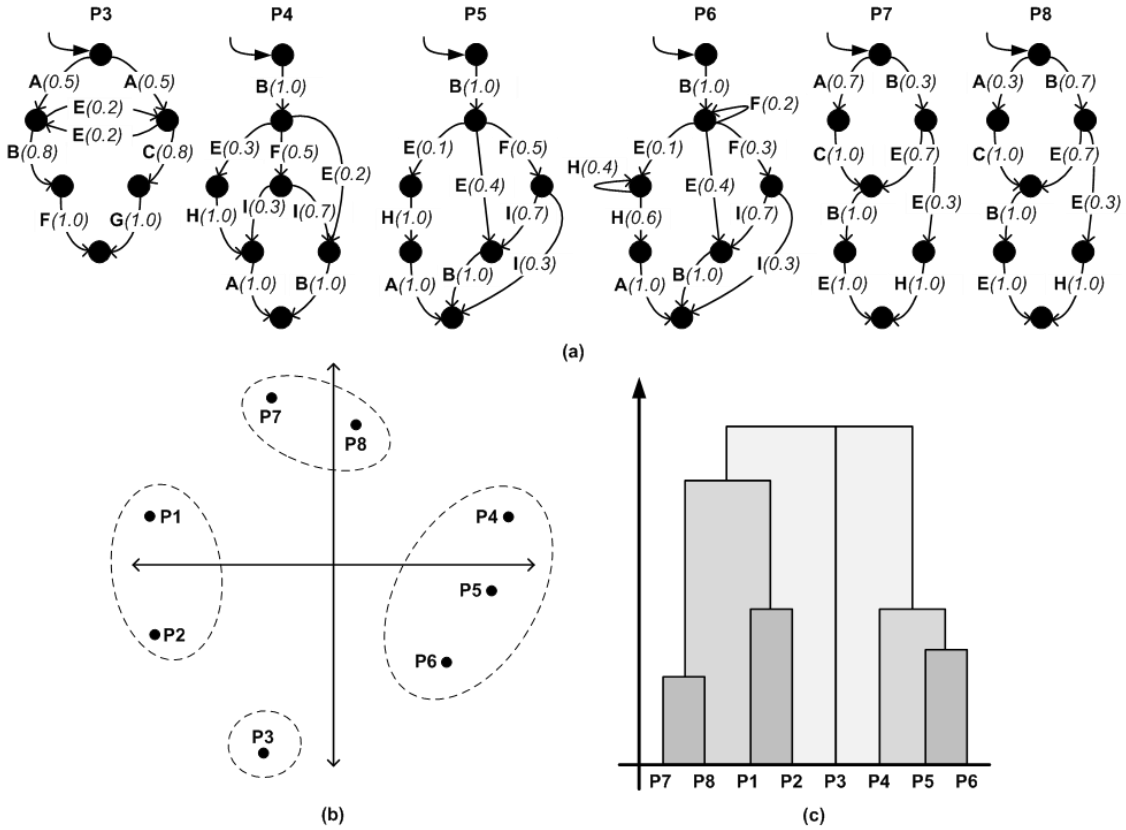

Figure 8: (a) defines some business processes as PLTSs, (b) represents the result of running MDS on these processes using fuzzy complete distance, (c) represents the dendrogram after running agglomerative clustering using strict overlapping distance. Processes P1 and P2 are shown in figure 4 (a) and (b).

6. CONCLUSION AND OUTLOOK

In this paper we presented a set of distance measures for business processes having the following distinct properties:

- *Behavioral*: The emphasis lies upon the behavior of the business process in form of the possible activity sequences the processes allow. No structural aspects are considered. The approach abstracts from the modeling language used to represent to process.
- *Probabilistic*: The measures explicitly incorporate the probabilities observing certain behavior and weights more probable behavior stronger.
- *Approximate*: The incorporation of probabilities allows approximating the distance of processes having infinitely many possible activity sequences.
- *Customizable*: Various different notions of distance measures are proposed ranging from very strong to more relaxed versions.
- *Metric*: The distance measures can be interpreted as metric distances, making them suitable for algorithms exploiting such structures.

The approach builds on the idea that business process behavior is not only defined by the possible activity sequences being compliant with a process model but also by the frequencies with which these activity sequences occur in the real world. It is, however, not restricted to cases in which explicit annotations on decision probabilities are given since it may be valid to make certain assumptions. For instance, one could assume that any branching of the control flow is equally likely, which would reduce our metrics to quantities very similar to other approaches in the literature.

Furthermore, one does not even have to assume the existence of an explicit process model to calculate distances. In many cases, process-aware information systems like ERP or CRM systems provide event logs documenting the past behavior of a possibly unknown process [35]. Clearly, such event logs can be used to approximate a distribution over activity sequences that can be used in a distance calculation.

In the future, we plan to intensively evaluate the metrics we have proposed, especially with respect to the conformance with human judgment. Several studies in literature evaluated similarity measures by experimentally counterchecking them with human opinions on similarity [16, 17, 35].

7. REFERENCES

[1] Kueng, P. 2000. Process performance measurement system: a tool to support process-based organizations. *Total Quality Management & Business Excellence* 11, 1, 67-85.

[2] van Dongen, B., Dijkman, R., Mendling, J. Year. Measuring Similarity between Business Process Models. In *Proceedings of the 20th International Conference on Advanced Information Systems Engineering, 2008*, 450-464.

[3] van der Aalst, W. M. P., Reijers, H. A., Weijters, A., van Dongen, B., de Medeiros, A. K., Song, M., Verbeek, H. M.

W. 2007. Business process mining: An industrial application. *Information Systems* 32, 5, 713-732.

[4] Namiri, K., Stojanovic, N. Year. Pattern-based design and validation of business process compliance. In *Proceedings of the 2007 OTM Confederated International Conference on the Move to Meaningful Internet Systems: CoopIS, DOA, ODBASE, GADA, and IS, 2007*, 59-76.

[5] Dumas, M., Garcıa-Banuelos, L., Dijkman, R. 2009. Similarity search of business process models. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* 32, 3, 23-28.

[6] Navarro, G. 2001. A Guided Tour to Approximate String Matching. *ACM Computing Surveys* 33, 1, 31-88.

[7] Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Massachusetts.

[8] Navigli, R. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)* 41, 2, 10.

[9] Hopcroft, J. E., Motwani, R., Ullman, J. D. 2001. *Introduction to automata theory, languages, and computation*. 2 ed., Addison-Wesley Longman, Amsterdam.

[10] Dijkman, R., Dumas, M., García-Bañuelos, L. Year. Graph matching algorithms for business process model similarity search. In *Proceedings of the 7th International Conference on Business Process Management (BPM), 2009*, 48-63.

[11] Minor, M., Tartakovski, A., Bergmann, R. Year. Representation and structure-based similarity assessment for agile workflows. In *Proceedings of the 7th International Conference on Case-Based Reasoning, ICCBR, 2007*, 224-238.

[12] Lu, R., Sadiq, S. Year. On the Discovery of Preferred Work Practice Through Business Process Variants. In *Proceedings of the 26th international conference on Conceptual modeling (ER'07), 2007*, 165-180.

[13] Madhusudan, T., Zhao, J. L., Marshall, B. 2004. A case-based reasoning framework for workflow model management. *Data & Knowledge Engineering* 50, 1, 87-115.

[14] Grigori, D., Corrales, J., Bouzeghoub, M. 2008. Behavioral matchmaking for service retrieval: Application to conversation protocols. *Information Systems* 33, 7-8, 681-698.

[15] Li, C., Reichert, M., Wombacher, A. Year. On measuring process model similarity based on high-level change operations. In *Proceedings of the 27th International Conference on Conceptual Modeling (ER'08), 2008*, 248-264.

[16] Dijkman, R., Dumas, M., Dongen, B. V. 2009. Similarity of business process models: Metrics and evaluation. *Information Systems* 36, 2, 498-516.

[17] van Dongen, B., Dijkman, R., Mendling, J. Year. Measuring Similarity between Business Process Models. In *Proceedings of the 20th International Conference on Advanced Information Systems Engineering, 2008*, 450-464.

[18] Zha, H., Wang, J., Wen, L., Wang, C., Sun, J. 2010. A workflow net similarity measure based on transition adjacency relations. *Computers in Industry* 61, 5, 463-471.

[19] Park, D. Year. Concurrency and automata on infinite sequences. In *Proceedings of the 5th GI-Conference on Theoretical Computer Science, 1981*, 167-183.

[20] Nejati, S., Sabetzadeh, M., Chechik, M. Year. Matching and merging of statecharts specifications. In *Proceedings of the 29th International Conference on Software Engineering, 2007*, 54-64.

[21] Sokolsky, O., Kannan, S., Lee, I. Year. Simulation-based graph similarity. In *Proceedings of the 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), 2006*, 426-440.

[22] Wombacher, A., Rozie, M. 2006. Evaluation of workflow similarity measures in service discovery. *Service Oriented Electronic Commerce*, 51-71.

[23] Mohri, M. 2003. Edit-distance of weighted automata: General definitions and algorithms. *International Journal of Foundations of Computer Science* 14, 6, 957-982.

[24] Mohri, M. Year. Edit-distance of weighted automata. In *Proceedings of the 7th International Conference on Implementation and Application of Automata (CIAA), 2003*, 1-23.

[25] de Medeiros, A. A., Aalst, W. V. D. 2008. Quantifying process equivalence based on observed behavior. *Data & Knowledge Engineering* 64, 1, 55-74.

[26] Winskel, G., Nielsen, M. 1995. *Models for concurrency*. Oxford University Press, Oxford, UK.

[27] Gibbs, A., Su, F. 2002. On choosing and bounding probability metrics. *International Statistical Review* 70, 3, 419-435.

[28] Bhattacharyya, A. 1943. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Sociecty* 35, 4.

[29] Thacker, N., Aherne, F., Rockett, P. 1997. The Bhattacharyya metric as an absolute similarity measure for frequency coded data. *Kybernetika* 34, 4, 363-368.

[30] Rauber, T., Conci, A., Braun, T., Berns, K. Year. Bhattacharyya Probabilistic Distance of the Dirichlet Density and its Application to Split-And-Merge Image Segmentation. In *Proceedings of the 15th International Conference on Systems, Signals and Image Processing (IWSSIP), 2008*, 145-148.

[31] Zezula, P., Amato, G., Dohnal, V., Batko, M. 2006. *Similarity search: the metric space approach*. 1st edn ed., Springer, New York.

[32] Hjaltason, G. R., Samet, H. 2003. Index-Driven Similarity Search in Metric Spaces. *ACM Transactions on Database Systems (TODS)* 28, 4, 517-580.

[33] Comaniciu, D., Ramesh, V., Meer, P. 2003. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 5, 564-575.

[34] Damerau, F. J. 1964. A technique for computer detection and correction of spelling errors. *Communications of the ACM* 7, 3, 171-176.

[35] van der Aalst, W. M. P., Weijters, A. 2004. Process Mining: A Research Agenda. *Computers in Industry* 53, 3, 231-244.