Analysing Differences Between Business Process Similarity Measures

Michael Becker[1] and Ralf Laue[2]

[1] Department of Business Information Systems, University of Leipzig, Germany
michael.becker@uni-leipzig.de
[2] Chair of Applied Telematics / e-Business, University of Leipzig, Germany
laue@ebus.informatik.uni-leipzig.de

Abstract. Nowadays, it is not uncommon that organisations maintain repositories containing hundreds or thousands of business process models. For the purpose of searching such a repository for models that are similar to a query model, many similarity measures have been suggested in the literature. Other measures have been suggested for different purposes like measuring compliance between a model and a reference model. As those similarity measures differ in many aspects, it is an interesting question how they rank "similarity" within the same set of process models. In our study, we investigated, how different kinds of changes in a process model influence the values of 22 different similarity measures that have been published in academic literature.
Furthermore, we identified eight properties that a similarity measure should have from a theoretical point of view and analysed how these properties are fulfilled by the different measures. Our results show that there are remarkable differences among existing measures. We give some recommendations which kind of measure is useful for which kind of application.

1 Introduction

Business Process Models (BPMs) are nowadays a common approach to analyse existing processes and to create new processes in a structured way. They are used for purposes like supporting the communication in organisations, documentation in projects, and training of employees [1]. This wide area of application has led to the existence of a tremendous amount of BPMs. Large scale enterprises usually own repositories consisting of hundreds or thousands of models [2], usually developed by different persons. To manage these repositories, suitable methods for searching a BPM repository are necessary. A common requirement is to search for BPMs that are similar to a given query model. For this purpose, there is a need of a similarity measure that quantifies the similarity between models.

As the similarity measures that have been suggested in the literature differ in many aspects, it is an interesting question how the different measures rank "similarity" within the same set of BPMs. In our study, we investigated, how different kinds of changes in a BPM influence the values of 22 different similarity measures that have been published in academic literature. To our best knowledge, no study has been made so far that compares such a large number of different BPM similarity measures.

After explaining some fundamental preliminaries in Sect. 2, we discuss some properties that a "good" similarity measure should have in Sect. 3. We apply different changes (described in Sect. 4) to an example model and calculate the similarity between the original model and its variants in Sect. 5 together with a discussion about the implications from our results. Finally, Sect. 6 gives a conclusion.

2 Approaches for Measuring Process Similarity

When BPMs have to be compared, the first challenge is to identify the activity nodes in one model that correspond to activity nodes in the other model. In particular, if the models have been created in different organisations or if they describe a business process on different levels of detail, this can become a non-trivial task. This first step is, however, not in the focus of our paper. We assume that a mapping between corresponding activity nodes in the BPMs to compare has been established, either by using one of the existing algorithms or based on experts' judgment. The interested reader can find a discussion of different mapping techniques in [3–5].

Once a mapping between the activities has been established, several approaches have been suggested for measuring the similarity of BPMs. Rather simple measures are related to the number of activities that two BPMs have in common [6, 5] or the percentage of nodes and arcs that can be found in both BPMs [7–10]. These measures can be considerably improved by considering as well the position that an activity has within a BPM [11].

As in the most common modelling languages, BPMs are modelled as directed attributed graphs, other researchers suggested to use graph-based approaches for comparing BPMs. A graph-based similarity measure is the edit-distance, i.e. the lowest number of elementary operations (like adding or deleting a node) that transfers one model (or graph) into another. Such measures are discussed in [12–15]. In [16], the use of a graph-edit distance based on high-level operations (containing more than one elementary operation) is discussed. Bae et al. [17] transform BPMs into trees before calculating a graph-based similarity measure between those trees.

Other authors compare the set of all possible traces (or possible sequences of activities) of a BPM [18]. As this set of traces can become very large or even infinite, it has also been suggested to compare process logs, i.e. a finite subset of the set of traces which can be obtained from simulation [19]. Another stream of research investigates relationships between activities in a BPM (like "A is always followed by B") in order to draw conclusions about their similarity [20–22, 3].

3 Desirable Properties of Distance and Similarity Measures

For introducing desirable properties for similarity measures for BPMs (or distance measures that aim to measure dissimilarity), we make use of the research results on properties of similarity measures in general [23–25].

Let $\mathbf{M}$ be the set of BPMs. A *distance measure dist* is a function $dist : \mathbf{M} \times \mathbf{M} \to \mathbb{R}^+ \cup \{0\}$. We assume that for comparing a BPM M_0 with a BPM M_1, a partial function *map* has been established that maps the nodes in M_0 to "corresponding" nodes in M_1. M_0 is said to be equal to M_1 (symbol: $M_0 = M_1$) if the set of nodes of M_0 is $n^1, n^2, \dots n^n$ while the set of nodes of M_1 is $map(n^1), map(n^2), \dots map(n^n)$, and the set of arcs of M_1 is identical to all those arcs $(map(n), map(m))$, where n and m are nodes in M_0.

A *similarity measure* is a function $sim : \mathbf{M} \times \mathbf{M} \to [0, 1]$. The formula

$$sim(x, y) = \frac{1}{1 + dist(x, y)} \tag{1}$$

can be used for a transformation between distance (i.e. dissimilarity) and similarity measures.

Santini and Jain [23] point out that a number of dissimilarity measures proposed in the literature assume that those measures are distance measures in a metric space. $(\mathbf{M}, dist)$ becomes a metric space, if the following properties hold:

Property 1 $dist(M_0, M_1) \geq 0 \quad \forall M_0, M_1 \in \mathbf{M}$ (non-negativity)
Property 2 $dist(M_0, M_1) = dist(M_1, M_0) \quad \forall M_0, M_1 \in \mathbf{M}$ (symmetry)
Property 3 $dist(M_0, M_1) = 0 \Leftrightarrow M_0 = M_1$
Property 4 $dist(M_0, M_2) \leq dist(M_0, M_1) + dist(M_1, M_2)$ (triangle inequality)

For measuring the "dissimilarity" distance between BPMs, it is reasonable to require Property 1 and Property 2. Property 3 that says that the distance between two models is 0 if and only if the models are identical is too strict for certain application areas. The same set of traces (i.e. the same set of possible executions of activities of a model M, denoted as $\Sigma(M)$) can be obtained in different ways. For example, the model shown in Fig. 2(a) (see Sect. 4) has the same set of traces as the model shown in Fig. 2(b). A distance measure that calculates the distance between both models as 0 would correctly describe the fact that both models show exactly the same business process.

A more relaxed requirement is that $dist(M_0, M_1)$ is 0 iff both models have the same set of traces. For our purposes, the sets of traces $\Sigma(M_0)$ and $\Sigma(M_1)$ are considered as being the same (symbol: $\Sigma(M_0) \equiv \Sigma(M_1)$) if $\langle s_1, s_2, \dots \rangle \in \Sigma(M_0)$ implies that $\langle map(s_1), map(s_2), \dots \rangle \in \Sigma(M_1)$ and vice versa, $\langle t_1, t_2, \dots \rangle \in \Sigma(M_1)$ implies that there is a $\langle s_1, s_2, \dots \rangle \in \Sigma(M_0)$ such that $map(s_i) = t_i \ \forall i$. With this interpretation of equality between sets of traces, Property 3 can be substituted by the less strict requirement:

Property 3a:
$dist(M_0, M_1) = 0 \Leftrightarrow \Sigma(M_0) \equiv \Sigma(M_1)$.

Property 4, the triangle inequality, is not essential for measuring the dissimilarity (distance) between BPMs (or for (dis)similarity measures in general, see [24]), therefore we will not examine the suggested measures with respect to this property. It is a useful property anyway, because a distance measure that fulfills all four properties given above allows to organize a BPM repository using data structures in which the search for similar models is very fast [26].

From an information-theoretic discussion of the concept of similarity (see [24, 25]), one more requirement for a similarity measure can be derived: Such a measure should take into consideration both the commonality between two models as the differences between them (**Property 5**). For example, we would not get a good similarity measure by just counting the number of activities that are shared among two models without relating this number to the overall number of activities in the models: If two models with 20 nodes have 15 node names in common, it would be reasonable to say that they are more similar to each other than two models with 200 nodes from which 15 node names can be found in both models.

As mentioned before, the definition of the function *map* that assigns "corresponding" nodes between two models is outside the main focus of this paper. We just assume that such a mapping has been established. The approaches that calculate *map* automatically start with a function *corr* which quantifies the similarity between single activities. It would be a desirable property of a similarity measure $sim : \mathbf{M} \times \mathbf{M} \rightarrow [0, 1]$ if the information gained from the similarity measure *corr* between *activities* would be considered in the calculation of the similarity measure *sim* between the *models as a whole* (**Property 6**). This is illustrated in Fig. 1, showing three sequential models M_0, M_1 and M_2 with four activities and the mappings between them (as dotted arrows). Assume that

$$\begin{aligned} 1 &= corr(\text{“confirm draft“}, \text{“confirm draft“}) \\ &> corr(\text{“confirm draft“}, \text{“dismiss draft“}) \end{aligned}$$

and that

$$\begin{aligned} 1 &> corr(\text{“sign draft contract“}, \text{“sign contract“}) \\ &= corr(\text{“sign draft contract“}, \text{“archive draft contract“}) \end{aligned}$$

(which could be the result *corr* defined as a simple word-by-word comparison). In such a case, it would be desirable that the result that the activities in M_2 are more similar to the activities in M_0 than those in M_1 would not "get lost" when the similarity measure *sim* is calculated, i.e. we would prefer to have $sim(M_0, M_1) < sim(M_0, M_2)$ instead of $sim(M_0, M_1) = sim(M_0, M_2)$.

Furthermore, it is reasonable to require that a distance or similarity measure can be applied for comparing arbitrary BPMs without imposing additional syntax restrictions (such as that the model must not contain loops) (**Property 7**). And last but not least, there is another requirement that is related to the computational complexity of the calculation of a distance or similarity measure.

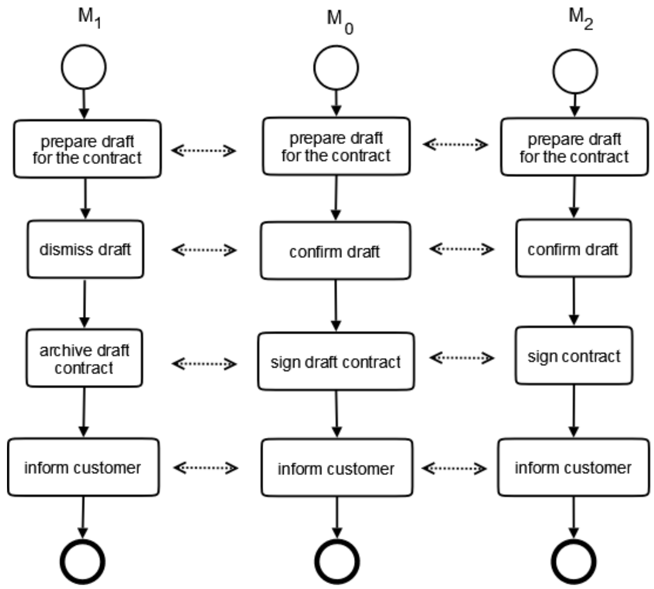


Fig. 1. Models with corresponding labels

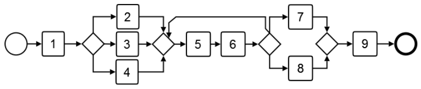
In simple terms, it should be possible to calculate the values of distance / similarity measures quickly (**Property 8**). Approaches that require the calculation of the whole set of traces of a BPM often do not fulfill this requirement.

4 Model Changes

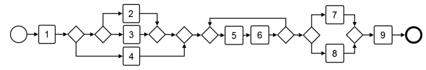
In order to analyse the values of similarity measures, we compute the similarity between some example models. For this purpose, we use BPMN models that can be built from the notational elements start event (exactly one per model), end event (exactly one per model), activity, AND connectors (i.e. split or join gateways in BPMN terminology), XOR connectors and inclusive OR connectors. Almost all similarity measure published in the literature are restricted to this subset of notational elements.

Starting from a moderately sized BPMN model, we apply different change operations as described in [27–29] and shown in Fig. 2. For the various similarity measures that have been described in the literature, we compute the similarity between the original model V_0 (Fig. 2(a)) and each of its variants $V_1, \dots V_7$. If the original authors of a measure described it as a distance measure rather than a similarity measure, we use Equation 1 for transforming the distance measure into a similarity measure.

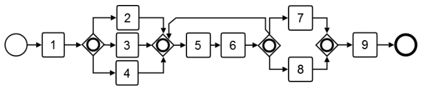
First, we modify the original model V_0 of Fig. 2(a) by splitting some XOR connectors into more than one connector (see Fig. 2(b)). Note that $\Sigma(V_1) = \Sigma(V_0)$. Next, we change the types of connectors: In model variant V_2 (Fig. 2(c)),



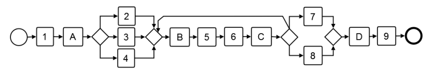
(a) V_0: original BPMN model



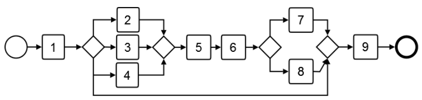
(b) V_1: model with same set of traces as V_0



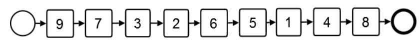
(c) V_2: model with modified connector types



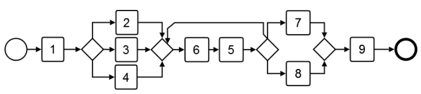
(d) V_3: model with additional activies



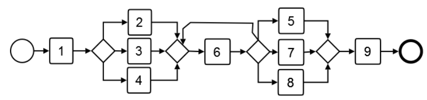
(e) V_4: model with modified control flow arcs



(f) V_5: model with modified control flow



(g) V_6: model with modified order of activities 5 and 6



(h) V_7: model with modified activity 5

Fig. 2. Initial model V_0 and variants $V_1 \ldots V_7$

all XOR connectors (◇) have been replaced by inclusive OR connectors (◈). In variant V_3 (Fig. 2(d)), four additional activities A, B, C and D have been added to the original model.

Model variant V_4 (Fig. 2(e)) has exactly the same nodes as V_0, but one arc has been added while another one has been deleted. Variant V_5 (Fig. 2(f)) does contain the same activities as V_0, but no connectors at all. The order of the activities does not correspond to the order in which the activities occur in executions of V_0. In model variant V_6 (Fig. 2(g)), the order of activities 5 and 6 has been changed. Finally, in model variant V_7, (Fig. 2(h)), activity 5 has been moved inside the second conditional control block.

5 Results and Discussion

Tab. 1 shows the support of the different measures for the properties discussed in Sect. 3. The results for property 8 are based on a somewhat subjective judgment – measures that require to calculate the set of traces of a BPM were categorised as computationally inefficient. From Tab. 1, we can see that no measure fulfills all desirable properties.

Tab. 2 shows the similarity values we have computed between our example model V_0 and its variants $V_1 \ldots V_7$. For measures that can be parametrised by attaching different weights to factors, we used the most reasonable and simple

Table 1. Comparison of similarity measures by adherence to properties in Sect. 3

	1	2	3	3a	5	6	7	8
Measures based on the correspondence of nodes and edges (not taking into account the control flow)								
Percentage of Common Activity Names [6]	yes	yes	no	no	yes	yes	yes	yes
Label Matching Similarity [3]	yes	no	no	no	yes	yes	yes	yes
Similarity of Activity Labels [5]	yes	no	no	no	yes	yes	yes	yes
Feature-Based Activity Similarity [11]	yes	yes	no	no	yes	yes	yes	yes
Percentage of Common Nodes and Edges [12]	yes	yes	no	no	yes	yes	yes	yes
Node- and Link-Based Similarity [10]	yes	yes	no	no	yes	yes	yes	yes
Measures based on graph edit distances								
Graph Edit Distance [3]	yes	yes	yes	no	yes	yes	yes	yes
Graph Edit Distance [30]	yes	yes	yes	no	no	no	yes	yes
Label Similarity and Graph Edit Distance [13]	yes	yes	no	no	yes	yes	yes	yes
Label Similarity and Graph Edit Distance [26]	yes	yes	yes	no	no	yes	yes	yes
Number of High-Level Change Operations [16]	yes	yes	no	yes	yes	no	n/a	yes
Comparing BPMs Represented as Trees [17]	yes	yes	no	no	yes	no	yes	yes
Distance Between Reduced Models [7]	yes	no	no	no	yes	no	yes	yes
Measures that analyse causal dependencies between activities								
Comparing Dependency Graphs [8, 9]	yes	yes	no	no	yes	no	yes	yes
Comparing Dependency Graphs [22]	yes	yes	no	no	yes	no	n/a	yes
Reference Similarity [21]	yes	yes	no	yes	yes	no	yes	no
TAR-Relationship [21]	yes	yes	no	no	yes	no	yes	no
Causal Behavioural Profiles [20]	yes	yes	no	no	yes	no	no	yes
Causal Footprints [3]	yes	yes	no	no	yes	no	yes	no
Set of Traces as n-grams [14]	yes	no	no	no	no	no	yes	no
Measures that compare sets of traces or logs								
Longest Common Subsequence of Traces [18]	yes	yes	no	no	yes	no	yes	no
Similarity Based on Principal Transition Sequences [31]	yes	yes	no	no	yes	no	yes	yes
Similarity Based on Traces [19]	yes	yes	no	no	yes	no	yes	yes

parameters. The table shows that a variety of measures deliver results that do not comply with the intuitive (but rather subjective) understanding of "process model similarity".

Similarity measures for BPMs have been proposed for a number of purposes. The purposes named in the literature include:

- finding "related" models in a repository [11]
- finding "similar" models in a repository for the purpose of reuse, preventing duplication and assisting process design [3, 26, 31, 21, 22, 6]
- minimise the efforts to transform one model into another one with the aim to support dynamic process changes [16, 20]
- identifying common or similar models in the context of company mergers [3, 21]
- measuring the conformance between a BPM used as system specification and a workflow model that implements the process [20]

Table 2. Similarity Measures for our Example Models

	Similarity between V_0 and ...						
	V_1	V_2	V_3	V_4	V_5	V_6	V_7
Measures based on the correspondence of nodes and edges (not taking into account the control flow)							
Percentage of Common Activity Names [6]	1.00	1.00	0.82	1.00	1.00	1.00	1.00
Label Matching Similarity [3]	1.00	1.00	0.82	1.00	1.00	1.00	1.00
Similarity of Activity Labels [5]	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Feature-Based Activity Similarity [11]	1.00	1.00	0.82	1.00	1.00	1.00	1.00
Percentage of Common Nodes and Edges [12]	1.00	1.00	0.40	0.95	0.58	0.76	0.79
Node- and Link-Based Similarity [10]	0.55	0.60	0.49	0.59	0.61	0.60	0.55
Measures based on graph edit distances							
Graph Edit Distance [3]	1.00	1.00	0.63	0.97	0.73	0.86	0.12
Graph Edit Distance [30]	0.05	0.04	0.20	0.33	0.03	0.33	0.17
Label Similarity and Graph Edit Distance [13]	0.81	1.00	0.60	0.96	0.61	0.79	0.84
Label Similarity and Graph Edit Distance [26]	0.05	0.03	0.06	0.33	0.03	0.14	0.20
Number of High-Level Change Operations [16]	1.00	0.17	0.20	0.33	0.14	0.50	0.50
Comparing BPMs Represented as Trees [17]	1.00	1.00	0.07	0.12	0.06	0.14	0.14
Comparing BPMs Represented as Trees [17]	1.00	0.07	1.00	1.00	0.00	0.80	0.81
Measures that analyse causal dependencies between activities							
Comparing Dependency Graphs [8, 9]	1.00	1.00	0.04	0.33	0.06	0.09	0.10
Comparing Dependency Graphs [22]	1.00	0.93	0.54	0.90	0.51	0.98	0.83
Reference Similarity [21]	not defined (V_0 has a loop!)						
TAR-Relationship [21]	1.00	0.57	0.04	0.85	0.11	0.41	0.47
Causal Behavioural Profiles [20]	1.00	0.93	0.63	0.93	0.22	0.98	0.89
Causal Footprints [3]	1.00	1.00	0.45	0.80	0.59	0.97	0.84
Set of Traces as n-grams [14]	1.00	0.10	0.04	0.33	0.05	0.09	0.10
Measures that compare set of traces or logs							
Longest Common Subsequence of Traces [18]	1.00	0.86	0.79	1.00	0.43	0.93	0.90
Similarity Based on Principal Transition Sequences [31]	1.00	0.83	0.61	0.84	0.20	0.85	0.83
Similarity Based on Traces [19]	1.00	0.90	0.33	0.83	0.22	0.72	0.65

– finding models or model parts that should be merged into one integrated model to improve the maintainability of the BPM repository [13]
– discovering services from a description of their behaviour [30, 19, 14]
– measuring the conformance between a BPM and a reference model [18, 20]
– comparing models that are constructed from the same template in order to manage process variants and to support flexible workflow systems [12, 7].

Our observations gives some first insights which measures are more useful than others for a given purpose. These suggestions are based on the results shown in Tab. 2. However, it must be stated that several advantages and disadvantages of the measures can only be evaluated based on specific application areas. For example, it cannot objectively be decided wheter V_3 or V_7 should be regarded as more similar to V_0.

Simple measures that just count the number of common nodes or arcs in the models are useful for finding related models from a repository (and less useful for purposes that make reference to the model behaviour). An interesting use case for such rather simple measures has been suggested by [11]: A search for related models can be used as a first step of a search in a large repository. It helps to filter out unrelated models such that the more precise (but also slower) algorithms can be applied to a small subset of the original search space.

When models are compared with the aim of discovering services or measuring conformance, approaches that consider the actual behaviour of a process execution have to be used. Preference should be given to the methods that exploit relationships between activities (such as [20–22]) instead of requiring a calculation of the whole set of traces (as [18]).

The reason is that calculating the whole set of traces of a model can demand large memory and processing resources. It has to be noted that the approach based on causal footprints described in [3] is computationally inefficient as well and cannot be recommended to be used in the context of large BPM repositories.

Processing speed can be less important if only two models have to be compared, for example to measure conformance. In such cases, using approaches that require to calculate the set of traces can be an option.

Some applications require to compare BPMs that have been designed on different levels of granularity. For example, this can be the case if the conformance between a BPM serving as a specification and the actual implementation in a workflow system have to be compared. In such cases, it is recommended to use a measure that finds a similarity even between such models. In particular, the approaches described in [7, 20, 18, 30] support the comparison of models on different abstraction levels.

Although not extensively discussed in our paper, it should be noted that the quality of the mapping between the nodes has a significant contribution to the quality of a similarity measure. In particular, regarding nodes as corresponding to each other only if they have exactly the same label is reasonable only in a few special application areas such as comparing models that have been derived from the same template.

To furthermore enhance the reproducability and significance of our findings, we developed an analysis plugin[3] for the well-known ProM - Framework for Process Mining tool [32]. The plugin takes two arbitrary types of process models as inputs and shows the similarity values for the different approaches presented in this paper. In this paper we only presented a qualitative analysis to show first insights about the various measures. Based on the plugin implementation, several process model repositories will be analysed, e.g. the SAP reference model consisting of more than 600 models [33].

6 Conclusion

In our paper, we elaborated a number of desirable properties for BPM similarity measures. We analysed 22 similarity measures that have been described

[3] available at https://sourceforge.net/projects/prom-similarity/

in the literature with respect to those properties. Also, we computed the similarity between example BPMs using the different similarity measures. While the rather small number of example models cannot show the relationships between the measures in a comprehensive manner, some first conclusions can be drawn. The results show that hardly a measure fulfills all desirable properties. Furthermore, it can be seen that different similarity measures rank the similarity between BPMs very differently. We conclude that there is not a single "perfect" similarity measure. Instead, we gave some recommendations for the selection of an appropriate similarity measure for different use cases.

References

1. Gulla, J.A., Brasethvik, T.: On the challenges of business modeling in large-scale reengineering projects. IEEE International Conference on Requirements Engineering (2000) 17
2. Dumas, M., García-Bañuelos, L., Dijkman, R.: Similarity search of business process models. IEEE Data Engineering Bulletin **32** (2009) 23–28
3. Dijkman, R., Dumas, M., van Dongen, B., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. Inf. Syst. **36** (2011) 498–516
4. Weidlich, M., Dijkman, R.M., Mendling, J.: The ICoP framework: Identification of correspondences between process models. In: Advanced Information Systems Engineering, 22nd International Conference, CAiSE 2010, Hammamet, Tunisia. Volume 6051 of LNCS., Springer (2010) 483–498
5. Ehrig, M., Koschmider, A., Oberweis, A.: Measuring similarity between semantic business process models. In: Fourth Asia-Pacific conference on Comceptual modelling - Vol. 67. (2007) 71–80
6. Akkiraju, R., Ivan, A.: Discovering business process similarities: An empirical study with SAP best practice business processes. In: Service-Oriented Computing - 8th International Conference. Volume 6470 of LNCS. (2010) 515–526
7. Lu, R., Sadiq, S.: On managing process variants as an information resource. Technical report, The University of Queensland, School of Information Technology and Electrical Engineering (2006)
8. Bae, J., Liu, L., Caverlee, J., Rouse, W.B.: Process mining, discovery, and integration using distance measures. IEEE International Conference on Web Services (2006) 479–488
9. Bae, J., Liu, L., Caverlee, J., Zhang, L.J., Bae, H.: Development of distance measures for process mining, discovery and integration. Int. J. Web Service Res. **4** (2007) 1–17
10. Huang, K., Zhou, Z., Han, Y., Li, G., Wang, J.: An algorithm for calculating process similarity to cluster open-source process designs. In: Grid and Cooperative Computing 2004 Workshops, Springer (2004) 107–114
11. Yan, Z., Dijkman, R.M., Grefen, P.: Fast business process similarity search with feature-based similarity estimation. In: On the Move to Meaningful Internet Systems - Confederated International Conferences, Proceedings 2010, Part I. Volume 6426 of LNCS., Springer (2010) 60–77
12. Minor, M., Tartakovski, A., Bergmann, R.: Representation and structure-based similarity assessment for agile workflows. In: 7th International Conference on Case-Based Reasoning, Springer (2007) 224–238

13. Rosa, M.L., Dumas, M., Uba, R., Dijkman, R.M.: Merging business process models. In: On the Move to Meaningful Internet Systems - Confederated International Conferences 2010, Proceedings, Part I. Volume 6426 of LNCS., Springer (2010) 96–113
14. Wombacher, A., Rozie, M.: Evaluation of workflow similarity measures in service discovery. In: Service Oriented Electronic Commerce: Proceedings zur Konferenz im Rahmen der Multikonferenz Wirtschaftsinformatik. Volume 80 of LNI., GI (2006) 51–71
15. Dijkman, R., Dumas, M., García-Bañuelos, L.: Graph matching algorithms for business process model similarity search. In: Business Process Management. Volume 5701 of LNCS. Springer (2009) 48–63
16. Li, C., Reichert, M., Wombacher, A.: On measuring process model similarity based on high-level change operations. In: 27th International Conference on Conceptual Modeling, Springer (2008) 248–264
17. Bae, J., Caverlee, J., Liu, L., Yan, H.: Process mining by measuring process block similarity. In: Business Process Management Workshops 2006. Volume 4103 of LNCS., Springer (2006) 141–152
18. Gerke, K., Cardoso, J., Claus, A.: Measuring the compliance of processes with reference models. In: On the Move to Meaningful Internet Systems - Confederated International Conferences 2009, Proceedings, Part I, Springer (2009) 76–93
19. Alves de Medeiros, A.K., van der Aalst, W.M.P., Weijters, A.J.M.M.: Quantifying process equivalence based on observed behavior. Data Knowl. Eng. **64** (2008) 55–74
20. Weidlich, M., Mendling, J., Weske, M.: Efficient consistency measurement based on behavioural profiles of process models. IEEE Transactions on Software Engineering **99** (2010)
21. Zha, H., Wang, J., Wen, L., Wang, C., Sun, J.: A workflow net similarity measure based on transition adjacency relations. Comput. Ind. **61** (2010) 463–471
22. Jung, J.Y., Bae, J., Liu, L.: Hierarchical clustering of business process models. International Journal of Innovative Computing, Information and Control **5** (2009) 1–11
23. Santini, S., Jain, R.: Similarity measures. IEEE Trans. Pattern Anal. Mach. Intell. **21** (1999) 871–883
24. Lin, D.: An information-theoretic definition of similarity. In: Proceedings 15th International Conference on Machine Learning. (1998) 296–304
25. Tversky, A.: Features of similarity. Psychological Review **84** (1977) 327–352
26. Kunze, M., Weske, M.: Metric trees for efficient similarity search in process model repositories. In: Proceedings of the 1st International Workshop Process in the Large (IW-PL '10). Hoboken, NJ, September 2010. (2010)
27. Dijkman, R.: A classification of differences between similar business processes. In: Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference, Washington, DC, USA, IEEE Computer Society (2007) 37–50
28. Weber, B., Reichert, M., Rinderle-Ma, S.: Change patterns and change support features - enhancing flexibility in process-aware information systems. Data Knowl. Eng. **66** (2008) 438–466
29. Weidlich, M., Barros, A.P., Mendling, J., Weske, M.: Vertical alignment of process models - how can we get there? In: Enterprise, Business-Process and Information Systems Modeling, 10th International Workshop. Volume 29 of LNBIP., Springer (2009) 71–84
30. Grigori, D., Corrales, J.C., Bouzeghoub, M., Gater, A.: Ranking BPEL processes for service discovery. IEEE Transactions on Services Computing **3** (2010) 178–192

31. Wang, J., He, T., Wen, L., Wu, N., ter Hofstede, A.H.M., Su, J.: A behavioral similarity measure between labeled petri nets based on principal transition sequences - (short paper). In: On the Move to Meaningful Internet Systems - Confederated International Conferences 2010, Proceedings, Part I. Volume 6426 of LNCS., Springer (2010) 394–401
32. van Dongen, B., de Medeiros, A., Verbeek, H., Weijters, A., van der Aalst, W.: The prom framework: A new era in process mining tool support. In Ciardo, G., Darondeau, P., eds.: Applications and Theory of Petri Nets 2005. Volume 3536 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2005) 1105–1116
33. Curran, T., Keller, G., Ladd, A.: SAP R/3 business blueprint: understanding the business process reference model. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1998)