Evaluating Social Tagging for Business Process Models

Ralf Laue[1] and Michael Becker[2]

[1] University of Applied Sciences of Zwickau, Germany
ralf.laue@fh-zwickau.de
[2] Department of Business Information Systems, University of Leipzig, Germany
michael.becker@uni-leipzig.de

Summary. Finding business process models in a model repository is a challenge that needs to be tackled for efficient business process management. Existing process model similarity measures compare models based on named elements and model structure. Social tagging enriches models with so-called tags – words or short phrases describing the content of the model. The tags given to models offer another possibility to judge about the similarity between models. In this paper we compare both approaches based on a study conducted with students. We discuss first insights and perspectives for tag-based search for process models.

Key words: New opportunities provided by social software for BPM; process model, similarity, tagging

1 Introduction

Due to the vast amount of existing process models in organisations, it is necessary to support users in developing, finding, and reusing models [1]. One approach for finding models in a repository is using an existing process model similarity measure. In [2] we presented a selection and comparison of different such measures.

One of the main success factors of social software, the *wisdom of the crowds*, can be seen as an inspiration for an alternative way to judge about the similarity between models. So-called tags - words or short phrases that describe the content of the model - can be assigned to the models in a repository by its users. Models are regarded as similar to each other if similar tags have been assigned to them. Social tagging for the purpose of finding models in a repository has been suggested by various authors such as Nigel et al. [3], Prilla [4] and Vanderhaeghen et al. [5].

Tools such as *Oryx* [6] and process model repositories[1] already offer possibilities for tagging and tag-based searching. However, we are not aware of any study that researches the suitability of social tagging for business process models. With our paper, we want to compare a similarity measure based on social tagging (obtained from an experiment) with eight existing approaches for calculating the

[1] see for example www.businessprocessincubator.com

similarity between business process models. In doing so, a first evaluation of the suitability can be established and used as groundwork for further research.

The remainder of this paper is structured as follows. Sect. 2 presents the experimental setup for assigning tags to models. Following, in Sect. 3 we describe the models used for comparison. Sect. 4 and 5 present the similarity of models based on similarity measures and on tag similarity. In Sect. 6 we compare the two approaches for calculating similarity. Finally, Sect. 7 presents conclusions and further research directions.

2 Experimental Setup

After giving a lecture on BPMN to a group of 23 bachelor students of information science at the University of Applied Sciences of Zwickau, we presented 15 BPMN models on paper to the students. We asked them to add tags describing the models such that those tags can be used by someone who searches for a model in a repository. Deliberately, the students did not get training in selecting "good" tags. The tasks had to be carried out as a pen-and-paper exercise, and no time limit was given. The labels were in German language which was the native language of all participating students.

Two of the students clearly misunderstood the nature of the tagging exercise and described the model in whole sentences. Their answers had to be disregarded, i.e. we collected the tags given by 21 students.

When compiling the machine-readable tag lists from the filled papers, we applied some simple sanity checks. For example, we corrected some spelling errors and used the same form for tags that appeared in different versions (for example both "check in" and "checkin" were transformed to "check-in"). All those sanity checks could easily be made by a computer system.

3 Models

To compare the different approaches for calculating similarity between process models, we selected 15 BPMN models. As booking of a flight or travel is a frequently used example in the BPMN literature, we selected several such models from different sources. This way, we tried to emulate a situation typically occurring in an organisation: Different modellers create models for the same kind of process with different granularity, different vocabulary and emphasis on different aspects.

For the purpose of this paper we named the models such that the model name clearly shows which activity is described by the model. The suffix `ONLINE` shows that the modelled process is performed using a web portal. Due to space limitations we only give a brief overview about the models. However, the entire collection of models is publicly available[2].

[2] `http://bis.informatik.uni-leipzig.de/MichaelBecker/files?get=bpms2.zip`

Table 1 shows the models we use in this paper. For every model, a unique name and a short description of the model is shown as well as the source. Many models marked as "created for the purpose of this paper" are adapted examples taken from [7]. For each model, we provide the number of nodes (activities and gateways) and edges.

4 Calculating Model Similarity

Several approaches for calculating similarity between process models have been introduced in the academic literature. In [2] we analysed a variety of these approaches and compared them according to several properties. We have identified three distinct types of measures: The first group (group A) contains measures based on activity labels only, not taking into account the control flow in a process model. The second group (group B) contains measures based on graph edit distance, i.e. the amount of steps necessary to transform one process model represented as a direct graph into another one. The third group (group C) considers causal dependencies between activities. These similarity measures analyse predecessor and successor relations between activities or compare sets of execution logs with each other.

To automate similarity calculation we have developed a plug-in for the process mining framework *ProM* [12]. This plug-in is publicly available as open source[3] and can be used as a standalone application, too. We used the *ProM* plug-in and an adaption of the measures as described in [2] for calculating several similarity measures for the models described in Sect. 3. Using the tool *Gephi* [13], the results of the calculations have been visualised as graphs in Fig. 1. The models are depicted as nodes. Similarity between nodes that exceeds a certain limit is depicted as an edge. The thickness of the vertices corresponds to the calculated similarity, i.e. the nodes depicting models that have been ranked as very similar to each other are connected by a thick edge. In addition, the Force Atlas algorithm built in in *Gephi* was used to visualise the similarity between models by the position of their corresponding nodes in the graph. It uses the principle of force-based graph layout algorithms [14]: similar nodes attract each other while non-similar ones are pushed apart.

The following measures were used to calculate similarity between process models:

(A1) *Similarity score based on common activity names*: Akkiraju and Ivan [15] propose a similarity measure based on the number of identically labelled activities. In order to calculate similarity, they use Dice's coefficient [16] for activity names.

(A2) *Label matching similarity*: The approach proposed by Dijkman et al. [1] is similar to measure A1 described above. However, Dijkman et al. introduce a threshold, i.e. the similarity of two activities is set to 0 if it is below a specific threshold.

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

Table 1. Models for similarity calculation

`BOOKING-FLIGHT1` (7 activities, 4 gateways, 12 edges, source: [8]) Passengers check and reserve seats and book flights. If booking takes too long, there is a timeout.
`BOOKING-FLIGHT2` (4 activities, 8 gateways, 15 edges, source: [9]) Passengers catch information on flight plans and delays, choose on-board catering, and process travel card.
`BOOKING-FLIGHT3` (3 activities, 4 gateways, 8 edges, source: [9]) Passengers catch information on flight plans and delays and book a flight.
`BOOKING-TRAVEL1` (source: [10]) This model of a travel booking process involves more than one swimlane and messages between actors. See Sect. 6 for details.
`BOOKING-FLIGHT-ONLINE1` (12 activities, 4 gateways, 19 edges, created for the purpose of this paper) Passengers log in to a website and book a flight.
`BOOKING-TRAVEL-ONLINE` (16 activities, 7 gateways, 27 edges, created for the purpose of this paper) Passengers log in to a website and book a flight. Furthermore, they can book a hotel, reserve a rental car, and buy travel cancellation insurance.
`BOOKING-TRAVEL2` (8 activities, 6 gateways, 17 edges, source: [11]) Passengers book a flight at a travel agency including hotel and rental car reservation.
`BOARDING1` (4 activities, 2 gateways, created for the purpose of this paper) The boarding pass of travelers is checked during boarding.
`BOARDING2` (4 activities, 2 gateways, 6 edges, created for the purpose of this paper) Essentially the same model as `BOARDING1` with renamed activities.
`MILES-ONLINE` (7 activities, 4 gateways, 12 edges, created for the purpose of this paper) Passengers log in to a website and can spend their frequent flyer miles by ordering products.
`UPGRADE-ONLINE` (10 activities, 8 gateways, 21 edges, created for the purpose of this paper) Passengers log in to a website and select an upgrade for their flight.
`CANCEL-ONLINE` (6 activities, 2 gateways, 9 edges, source: created for the purpose of this paper) Passengers log in to a website and cancel their booking by entering a booking code.
`BAGGAGE` (7 activities, 4 gateways, 12 edges, source: created for the purpose of this paper) Luggage is weighted. If it is too heavy or too big, the passengers have to pay fees for excess luggage. Luggage is registered and a luggage security control is conducted.
`CHECK-IN` (7 activities, 4 gateways, 13 edges, source: created for the purpose of this paper) Passengers need to proof their identity using a pass, a frequent flyer card, or an ID card. Following, their ticket is scanned, a seat is selected, and the boarding pass is printed.
`SHOP-ONLINE` (10 activities, 8 gateways, 22 edges, source: created for the purpose of this paper) Process for an online bookshop: Customers log in or register, browse book offers, and order books. Payment can be done using a credit card, direct debit, or a by using their frequent flyer card.

(A3) *Feature-based similarity estimation*: Yan et al. [17] compare models by taking into account the similarity of activity names in conjunction with similarity based of the amount of incoming and outgoing edges of an activity node.

(A4) *Percentage of common nodes and edges in the graph*: Minor et al. [18] transform process models into their graph representation and calculate similarity according to the amount of identically labelled activity nodes and coinciding edges between these nodes.

(A5) *Node- and link-based similarity*: In [19] Huang et al. present a similarity measure that works on common nodes and edges, too. However, they weight edges according to their relevance in the process model, e.g. outgoing edges of an XOR-split with n exits get a weight of $\frac{1}{n}$.

(B1) *Graph edit distance similarity*: Dijkman et al. [1] use a graph edit distance consisting of the amount of necessary elementary operations to transform one process model graph into another one. Examples for such elementary operations are adding and deleting nodes and edges.

(C1) *Dependency graph comparison by hierarchical clustering*: Jung et al. [20] analyse similarity based on dependencies between activities introduced due to control flow connections. Furthermore, they assign so-called execution probabilities to activities. The execution probability is governed by preceding control flow splits, e.g. exclusive choices or parallelisation. Similarity is then calculated by comparing dependencies and execution probabilities of two process models.

(C2) *TAR-similarity*: Similar to the dependency graph established by Jung et al., Zha et al. [21] introduce a TAR set containing activities that are in direct predecessor-successor-relation. The similarity of two process models is established by comparing their TAR sets with each other.

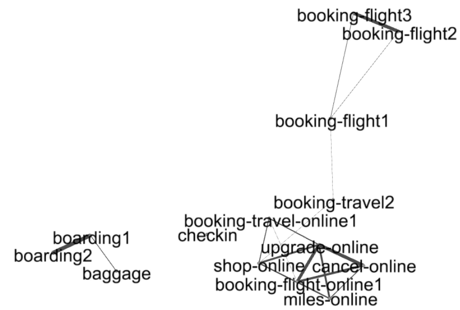
Before it is possible to calculate similarity between process models, a mapping between elements of these models is necessary. This means that for each node (in particular for each activity node) in a model it has to be analysed whether there is a node in the other model that corresponds to this node. Approaches for finding such a mapping are discussed in [1, 22]. For the purpose of our paper, we established a mapping manually (which is most likely superior to computer-based algorithms). Due to the nature of the models every node in one model corresponds to exactly one node in the model to compare with. This is especially worth mentioning when a similarity measure takes the edges of a graph into account - which is done by most similarity measures used in this paper, with A1 and A2 being the only exceptions.

5 Calculating Tag Similarity

In our experiment, a total of 1637 tags were given to the models, detailed statistics are shown in Tab. 2.

We regarded the tags given to a model M as the elements of a multiset T (where an element can be contained more than once). For calculating the

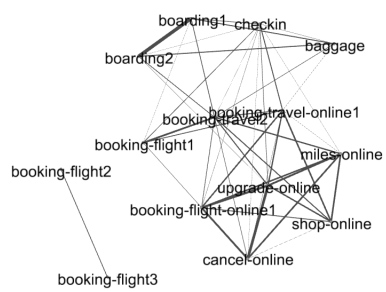
(a) A1: Common activity names



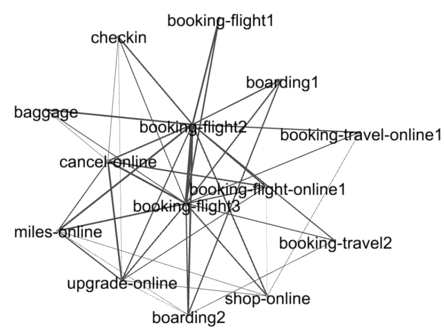
(b) A2: Label matching

(c) A3: Feature-based similarity

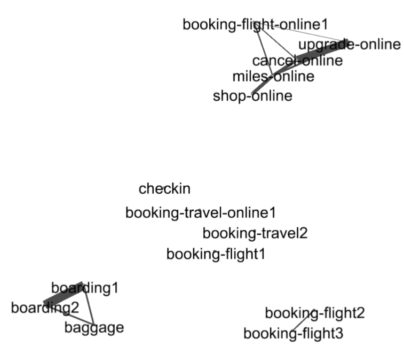
(d) A4: Common nodes and edges



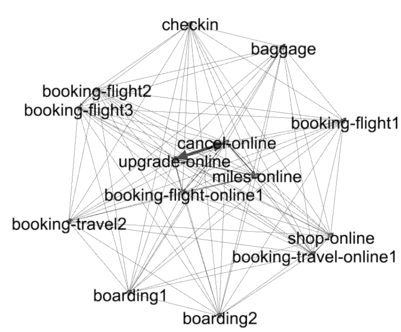
(e) A5: Node-and Link based



(f) B1: Graph edit distance



(g) C1: Dependency graph



(h) C2: TAR sets

Fig. 1. Visualisation of the similarity, calculated by different measures

Table 2. Number of tags collected from the 21 participants

Model	Tags	Different tags	Tags given by at least 2 persons
BAGGAGE	117	60	22
BOARDING1	82	45	17
BOARDING2	78	39	17
BOOKING-FLIGHT-ONLINE1	153	66	32
BOOKING-FLIGHT1	92	35	14
BOOKING-FLIGHT2	81	37	10
BOOKING-FLIGHT3	68	30	10
BOOKING-TRAVEL-ONLINE1	184	88	38
BOOKING-TRAVEL1	145	60	21
BOOKING-TRAVEL2	92	38	14
CANCEL-ONLINE	94	41	17
CHECK-IN	96	44	17
MILES-ONLINE	94	50	14
SHOP-ONLINE	156	68	27
UPGRADE-ONLINE	105	50	20

similarity between a model M_1 with the multiset of tags T_1 and a model M_2 with the multiset of tags T_2, we used Dice's coefficient [16], which is defined as $similarity(M_1, M_2) = \frac{2|T_1 \cap T_2|}{|T_1|+|T_2|}$, where $|T_i|$ denotes the number of elements in the tag list T_i (if the multiset contains an element more than once, each occurrence is counted.) This resulted in the similarity measures given in Table 3.

Fig. 2. Visualisation of Tag-based similarity

Table 3. Tag similarity (percentage of tags occurring in both models)

Model	BAGGAGE	BOARDING1	BOARDING2	BOOKING-FLIGHT-ONLINE1	BOOKING-FLIGHT1	BOOKING-FLIGHT2	BOOKING-FLIGHT3	BOOKING-TRAVEL-ONLINE-1	BOOKING-TRAVEL1	BOOKING-TRAVEL2	CANCEL-ONLINE	CHECK-IN	MILES-ONLINE	SHOP-ONLINE	UPGRADE-ONLINE
BAGGAGE	100	17	10	7	2	3	3	5	3	1	3	11	2	4	3
BOARDING1	17	100	30	5	10	4	5	3	4	5	3	21	6	1	3
BOARDING2	10	30	100	3	3	2	2	2	2	3	2	11	2	2	2
BOOKING-FLIGHT-ONLINE1	5	5	3	100	15	2	16	33	15	8	36	6	29	27	43
BOOKING-FLIGHT1	2	10	3	15	100	5	21	10	21	16	21	12	9	1	9
BOOKING-FLIGHT2	3	4	2	2	5	100	52	3	4	1	3	5	4	0	3
BOOKING-FLIGHT3	3	5	2	16	21	52	100	10	18	3	11	6	6	0	12
BOOKING-TRAVEL-ONLINE1	5	3	2	33	10	3	10	100	17	18	16	3	11	25	26
BOOKING-TRAVEL1	3	4	2	15	21	4	18	17	100	15	10	4	15	11	16
BOOKING-TRAVEL2	1	5	3	8	16	1	3	18	15	100	2	2	3	12	11
CANCEL-ONLINE	3	3	2	36	21	3	11	16	10	2	100	4	27	14	42
CHECK-IN	11	21	11	6	12	5	6	3	4	2	4	100	5	2	4
MILES-ONLINE	2	6	2	29	9	4	6	11	15	3	27	5	100	29	33
SHOP-ONLINE	4	1	2	27	1	0	0	25	11	12	14	2	29	100	29
UPGRADE-ONLINE	3	3	2	43	9	3	12	26	16	11	42	4	33	29	100

6 Comparing Similarity based on Tags to Other Approaches

The similarity measure calculated from the tag lists turned out to have some desirable properties: The largest similarity value (0.52) has been given to the two models `BOOKING-FLIGHT2` and `BOOKING-FLIGHT3`, the models that have been published as examples for variants of the same base model in [9]. The second largest similarity value (0.43) has been calculated between `UPGRADE-ONLINE` and `BOOKING-FLIGHT-ONLINE` which also seems to be quite reasonable.

The models `BOARDING1` and `BOARDING2` (in fact identical models, due to activity renaming) have 30% of the tags in common. On the other hand, the model `SHOP-ONLINE` (for which the only link to the airline domain is that it is possible to pay using frequent flyer miles) received the lowest similarity scores when being compared with non-online processes. The most similar model to `SHOP-ONLINE` is `MILES-ONLINE` - in fact both represent a possibility to spend frequent flyer miles using a web site.

If we compare Fig. 2 with the graphs in Fig. 1, we can see that the tag-based similarity measure has some commonalities with the results obtained by measures A1, A2, and A3. Measures that calculate similarity based on direct predecessor-successor-relations like B1 and C2 cannot find similarities between models that do not share edges. For example, models `BOOKING-FLIGHT-ONLINE1` and `BOOKING-TRAVEL-ONLINE1` share several activities that occur in a sequence. However, due to different ordering of sequences, measure B1 results in a large number of edit operations necessary to modify the edges. Measure C2 on the other hand does not find common sequences of activities. Both facts result in a rather low similarity score in comparison to tag similarity for these two models. Therefore, we conclude that similarity search based social tagging is at least a useful alternative to the established algorithms for detecting similar models.

An interesting observation was that some tags were given not based on the information directly available in the model but based on the domain knowledge of the users. Examples for such tags were "terminal" (where baggage check and check-in take place) and "complete vacation package" (for booking hotel, flight and rental car together). Finding such connections is really hard to achieve by computer-based similarity detection - even by using ontology-based approaches. This is the major difference between a tag-based similarity measure and measures that compare activity names only (such as measure A1 and A2). However, as Koschmider et al. [23] pointed out, it is reasonable to combine both approaches by using activity names as candidates for tags.

Two other advantages of tag-based search are worth mentioning as well. First, all currently known algorithms for calculating model similarity (see [2]) only consider the most basic notational elements of a modelling language such as BPMN. None of these algorithms take model elements like data-flow, swimlanes, exceptions, etc. into account. This is the reason why the model `BOOKING-TRAVEL1` which makes use of swimlanes and data-flow arcs could not be compared to the other models using the traditional algorithms. Second, tag-based search can be used in a more flexible way than similarity-based search. While the latter requires constructing a model before searching for similar ones, it would be sufficient to type in a few tags in order to make use of tag-based search in a model repository.

Besides differences in the results, there are also some technical differences between tag similarity and existing similarity measures. A great advantage of tag based similarity is that it is not necessary to establish a mapping between elements of the process models. Furthermore, the similarity between tags can be calculated relatively efficiently. Both facts reduce the complexity and increase the efficiency of calculating tag similarity in comparison to the other approaches. However, it is necessary to note that tag bases similarity cannot take the structure of models into account, i.e. structural differences between models are not identified if the models are tagged with similar words. Homonyms (words which have more than one meaning) add to the relevance of this problem.

7 Conclusion

In this paper we presented two different approaches for calculating similarity between process models. First, similarity can be calculated using an established measure from literature based on process models as direct input. Second, we presented a measure calculating the similarity between tags that have been manually assigned to the models. We compared the results of these measures and pointed out some observations worth mentioning.

For comparing tags we used a rather simple similarity measure based on Dice's coefficient. In real-world applications, better results can be expected when this measure is tuned for dealing with flawed tags and imprecise use of language. For example, using synonyms and preprocessing homonyms as shown in [24] is a reasonable approach to increase the precision of tag similarity. Furthermore, users of tagging systems might be supported by spell checkers and corporate taxonomies.

The study described in this paper was accomplished using tags given by bachelor students. The students are process model novices and we did not restrict their creativity during tagging. This might seem as a limitation of the experiment. Anyway, we think that we were able to obtain promising results (based on the comparison to model similarity) in this experimental setting. These findings should encourage future research to repeat the experiment with both students trained in process modelling as well as practitioners having better domain knowledge.

The results of this paper can be seen as a first evaluation of searching process models based on tag similarity. Using this evaluation as a basis, it was possible to identify advantages and disadvantages of this approach. However, due to our experimental setup, we cannot state whether the results of a tag-based search for similar models meet the expectations of users. In future research, it is necessary to empirically evaluate the results. This can be achieved by comparing the tag similarity with a manually established ranking of models similar to the evaluation in [1].

8 Acknowledgement

The German Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung, BMBF) partially supported this work in the context of the DLR project "Concept and Implementation of an Information Production System (IPS) for Precision Farming" ("Konzept und Implementierung eines Informations-Produktionssystems (IPS) im Precision Farming", support code 01IS12013B). Additional information on this project can be found online at `http://ips.uni-leipzig.de`.

References

1. Dijkman, R., Dumas, M., van Dongen, B., Kaarik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. Information Systems **36**(2) (2011) 498 – 516
2. Becker, M., Laue, R.: A comparative survey of business process similarity measures. Computers in Industry **63**(2) (2012) 148–167
3. Fengel, J., Rebstock, M., Nüttgens, M.: Modell-Tagging zur semantischen Verlinkung heterogener Modelle. In: EMISA 2008, Sankt Augustin, Germany, Gesellschaft für Informatik (September 2008) 53–69
4. Prilla, M.: Models, Social Tagging and Knowledge Management - A fruitful Combination for Process Improvement. In: Business Process Management Workshops. Volume 43 of LNBIP. Springer (2010) 266–277
5. Vanderhaeghen, D., Fettke, P., Loos, P.: Organizational and technological options for business process management from the perspective of web 2.0. Business & Information Systems Engineering **2**(1) (2010) 15–28
6. Decker, G., Overdick, H., Weske, M.: Oryx — an open modeling platform for the BPM community. In: BPM '08: Proceedings of the 6th International Conference on Business Process Management. Volume 5240 of LNCS., Springer (2008) 382–385
7. Störrle, H.: UML 2 erfolgreich einsetzen. Addison-Wesley, Munich (2005)
8. Wong, P.Y.H., Gibbons, J.: A process semantics for BPMN. In: 10th International Conference on Formal Engineering Methods, ICFEM 2008. Volume 5256 of LNCS., Springer (2008) 355–374
9. Montero, I., na, J.P., Ruiz-Cortés, A.: ATL transformation: Feature models for representing runtime variability in BIS to business process model notation. (2008) `http://www.isa.us.es/uploads/tools/fm2bpmn/doc/draft.pdf`.
10. Gravesande, K., Kim, S., Mowla, R., Resptrepo, A., Thomas, C., Weaver, S.: The essentials of BPEL. (2008) `http://csis.pace.edu/~ctappert/dps/d861-08/team1.pdf`.
11. White, S.A.: Using BPMN to model a BPEL process. BP Trends (March 2005)
12. 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: Applications and Theory of Petri Nets 2005. Volume 3536 of LNCS. Springer (2005) 1105–1116
13. Bastian, M., Heymann, S., Jacomy, M.: Gephi: An open source software for exploring and manipulating networks. In Adar, E., Hurst, M., Finin, T., Glance, N.S., Nicolov, N., Tseng, B.L., eds.: ICWSM, The AAAI Press (2009)
14. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. Software: Practice and Experience **21**(11) (1991) 1129–1164
15. Akkiraju, R., Ivan, A.: Discovering business process similarities: An empirical study with SAP best practice business processes. In: Service-Oriented Computing. Volume 6470 of LNCS. Springer (2010) 515–526
16. Dice, L.R.: Measures of the amount of ecologic association between species. Ecology **26**(3) (1945) 297–302
17. Yan, Z., Dijkman, R., Grefen, P.: Fast business process similarity search with feature-based similarity estimation. In: On the Move to Meaningful Internet Systems: OTM 2010. Volume 6426 of LNCS. Springer (2010) 60–77
18. Minor, M., Tartakovski, A., Bergmann, R.: Representation and structure-based similarity assessment for agile workflows. In: Case-Based Reasoning Research and Development. Volume 4626 of LNCS. Springer (2007) 224–238

19. 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. Volume 3252 of LNCS. Springer (2004) 107–114
20. Jung, J.Y., Bae, J., Liu, L.: Hierarchical clustering of business process models. International Journal of Innovative Computing, Information and Control **5**(12) (2009) 1349–4198
21. Zha, H., Wang, J., Wen, L., Wang, C., Sun, J.: A workflow net similarity measure based on transition adjacency relations. Computers in Industry **61**(5) (2010) 463 – 471
22. Weidlich, M., Dijkman, R., Mendling, J.: The ICoP Framework: Identification of Correspondences between Process Models. In: Advanced Information Systems Engineering. Volume 6051 of LNCS. Springer (2010) 483–498
23. Koschmider, A., Hornung, T., Oberweis, A.: Recommendation-based editor for business process modeling. Data & Knowledge Engineering **70**(6) (2011) 483 – 503
24. Ehrig, M., Koschmider, A., Oberweis, A.: Measuring similarity between semantic business process models. In: APCCM '07: Proceedings of the fourth Asia-Pacific conference on Comceptual modelling, Darlinghurst, Australia, Australian Computer Society, Inc. (2007) 71–80