Comparing Complex Business Process Models

Philip Weber[1*], Paul Taylor[2], Basim Majeed[3], Behzad Bordbar[1]
[1]School of Computer Science, University of Birmingham, B15 2TT, UK, ({p.weber,b.bordbar}@cs.bham.ac.uk.)
[2]BT Innovate & Design, Adastral Park, Martlesham Heath, Ipswich, UK, (paul.n.taylor@bt.com).
[3]Etisalat BT Innovation Centre (EBTIC), Khalifa University, Abu Dhabi UAE, (basim.majeed@bt.com).

***Abstract*—Business process models extracted from real event data via process mining are often visually complex and hard to interpret, cannot easily display all of the relevant process data in one view, and are of unknown quality. It is therefore difficult to present a meaningful comparison of two mined process models, for example to confirm business changes have had a significant effect. We propose a framework for comparing process models, which integrates metrics, graph partitioning, notions of statistical significance, and visualisation techniques. This framework enables well-founded and intuitive methods for understanding and exploring differences between processes.**

***Keywords*—Business process mining, graph partitioning, visualisation, probability distributions.**

I. Introduction

Business process mining is the learning and analysis of business process models, from event logs produced by information systems. Fig.1 shows a simplified example of a process, using an informal graphical representation. Nodes represent tasks and arcs the causal relations between them, labelled with frequencies of occurrence of the task or of following the arc, in the data from which this model was mined.

Many mined processes are not so structured, due to flexible work, process activity recorded at different levels of abstraction (e.g. [1]), or multiple processes recorded together (e.g. [2]). Traditional process mining algorithms then produce visually complex 'spaghetti' [3] models. Methods to reduce the complexity of models include clustering or aggregating process instances [2], [4], [5] or nodes and arcs [6]–[8]. This is not always desirable, e.g. if the most important questions are what are are the 'outlying' or rare process flows or tasks.

Recently a cartographic metaphor has been proposed [3], [6] for visualising processes, inspired by the use of line weight, colour, etc. in maps. Node sizes are varied to indicate relative significance (of tasks) or how many subnodes they aggregate, arcs are weighted or coloured to indicate significance of the relationships represented, and strength of correlation between the tasks they connect. This visual metaphor is helpful in allowing increased information to be conveyed in a manageable way, but has not been applied to comparing process models.

Since businesses use processes to ensure efficient operation or adherence to rules, it is critical to be able to compare mined processes to detect unexpected change, or to measure the effect of planned process changes. For complex processes, it is especially important to be able to pinpoint which parts of the model have changed. To this end, many metrics have been proposed to measure the difference between process models, such as by counting nodes and arcs [9], quantifying behavioural aspects in a particular representation (e.g. [10]), or comparing with 'reference' log files [11].

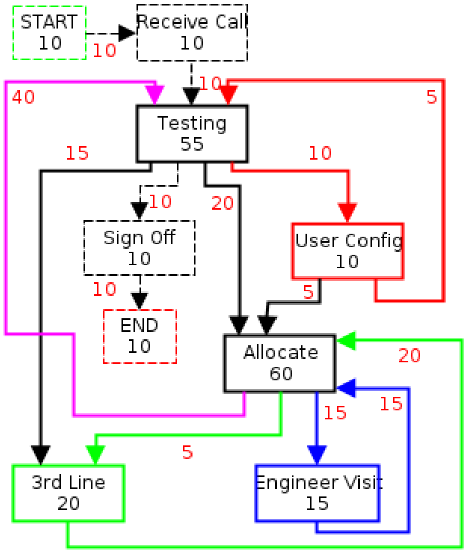


Fig. 1. Example process to handle a customer call for product support. Initial testing is followed by user configuration or allocation to engineer or 3^{rd} line support. Several iterations can occur before the call is complete.

Many metrics are specific to one representation such as Petri nets, or are difficult to interpret, especially for complex models: what is the meaning of a single number representing the difference? What if parts of the model are similar while other parts differ? In addition, metrics are usually limited to the model structure and behaviour, and do not take account of other process and task attributes which may be relevant.

Existing metrics also do not consider the significance of differences. The underlying process is stochastic: sequences of tasks (traces) occur with specific probabilities, so there will be random variation in the frequencies of traces observed in different logs produced by the process. This variation may cause differences between models mined from different event logs from the same process. The values calculated for measures of difference between models therefore depend on factors such as characteristics of the models, underlying event probabilities, how much data was used and the learning

[*] P. Weber is supported by a Doctoral Training Grant funded by EPSRC and the School of Computer Science, Birmingham, UK. This work was carried out under the International Fellowship program at the Etisalat BT Innovation Centre (EBTIC), Abu Dhabi, United Arab Emirates.

behaviour of the mining algorithm (see e.g. [12]).

We propose an interactive and flexible framework for comparing business process models, bringing together

1) *metrics* for comparing whole or part process models,
2) *graph partitioning* to compare the component parts of process models at different levels of abstraction,
3) *visualisation* techniques to intuitively present the differences between complex models, and
4) *statistical significance* tests to give meaning to metrics and quantify the confidence which can be placed in them.

Business process mining is an inherently practical activity, needing flexible solutions which enable exploratory analysis. Metrics and statistical analysis provide a rigorous foundation, combined with visualisation techniques to allow differences between models to be located, quantified and explored.

A framework allows choice of the best components for any situation, and the use of improved methods in the future. Different metrics may be appropriate to different process representations or applications, and the choice of graph partitioning algorithm may depend on the type of graph representation or how structured the process models are. Many approaches to visualisation could be applied. The discussion of statistical significance depends on what is known about the provenance of the process data, behaviour of the mining algorithm, and how faithfully the mined model represents probabilities in the underlying model. All of these areas should be considered together when comparing process models.

In this paper we describe and present

- our implementation of the components of this framework, and initial results,
- extensions to the difference measures in [13] and [14] to include processes with frequency information and multiple data attributes, and
- extending the cartographic metaphor for understanding complex process models, to visualisation of the differences between processes.

We briefly describe our event data and process model definition, before describing our framework in section III.

II. Preliminaries

A. Event Data

Assume a business process which records the execution of process instances, i.e. sequences of tasks with a defined start and end task. Task start and end times are recorded, together with names and values for other task and process attributes.

B. Process Model

We exemplify our framework using a process definition from a particular industrial context, presented in [15]. A process is a directed graph $G = (V, E, T, l, f, g, D_V, D_G)$:

- V is a set of nodes representing tasks. T is a set of task names. Nodes $v^{(i)} \in V$ are labelled with names from T by function $l : V \rightarrow T$.
- $E \subset V \times V$ are edges describing relations between tasks. We write $v^{(i)} \rightarrow v^{(j)}$ for arc from node $v^{(i)}$ to $v^{(j)}$.
- functions $f : V \rightarrow \mathbb{N}$ and $g : E \rightarrow \mathbb{N}$ assign frequencies of use ('weights') to nodes and arcs. We write $f(i)$ for weight of node $v^{(i)}$, $g(ij)$ for weight of arc $v^{(i)} \rightarrow v^{(j)}$.
- $D_G \subset (A_G \times d_G), D_V \subset (V \times A_V \times d_V)$ are sets of attribute names A_G, A_V and values d_G, d_V, associated with process instances and tasks respectively.

The mining algorithm naïvely constructs a process model from instances in such a way that the type (exclusive choice or parallel split) of splits and joins cannot be determined. Hence our models are not formally parsable, and we treat them as simple directed graphs. Fig.1 shows a simple example process.

III. An Implementation of the Framework

Next we describe and evaluate an implementation of the framework, applied to processes as defined in section II.

A. Metrics for Comparison by Node and Arc Frequencies

We build on the adjacency matrix comparison method of [13] to calculate a weighted average of the difference between the graphs' (weighted) arcs and (weighted) nodes. These are comparable to the metrics of [14].

We consider two process models $G_1 = (V_1, E_1, l_1, f_1, g_1)$, $G_2 = (V_2, E_2, l_2, f_2, g_2)$[1], with $N_1 = |V_1|, N_2 = |V_2|$ nodes respectively, N unique nodes in total which may be in one or both models, i.e. $|V_1 \cap V_2| \geq 0$, $N = |V_1 \cup V_2| \leq N_1 + N_2$.

The simplest comparison is of weights of nodes and arcs in the two models. Assume that both models were generated from the same number of process instances, or that frequencies have been scaled as though this was the case.

Following [13] we define $N \times N$ 'normalised' adjacency matrices A_1 and A_2 for for G_1, G_2. The matrix elements $a_{k\in\{1,2\}}^{(ij)}$ are set to the proportion of process instances from the arc's source node $v_k^{(i)}$ that follow the arc to node $v_k^{(j)}$. If node $v_k^{(i)}$ is labelled with frequency $f_k(i)$, and the arc from $v_k^{(i)}$ to $v_k^{(j)}$ with frequency $g_k(ij)$, then

$$a_k^{(ij)} = \frac{g_k(ij)}{f_k(i)} \in [0, 1]. \tag{1}$$

$a_k^{(ij)} = 0$ if there is no arc between the nodes.

We also define a $N \times N$ node *similarity* matrix S. Nodes are matched on task name, or using other methods, e.g. from [14], to allow for duplicately-labelled tasks or to match nodes by their context (e.g. neighbourhood) in the graph. Let

$$d(i, j) = \frac{f_1(i) - f_2(j)}{f_1(i) + f_2(j)} \in [-1, 1] \tag{2}$$

be the signed 'distance' between matched nodes $v_1^{(i)} \in V_1, v_2^{(j)} \in V_2$, labelled with frequencies $f_1(i), f_2(j)$ respectively. The elements $s^{(ij)}$ of the node similarity matrix S are set to $sign\big(d(i,j)\big) - d(i,j)$ if the nodes are matched, else 0. (The $sign$ function returns 1 for a positive or zero argument, -1 for a negative argument). When nodes in G_2 match uniquely with nodes in G_1 and *vice versa*, N is a diagonal matrix, the identity matrix if all nodes match exactly.

[1] For simplicity we omit notation not relevant to the discussion here.

We now define a Graph Edit Distance

$$g = w \cdot \Delta_A + (1-w) \cdot \Delta_N, w \in [0,1], \tag{3}$$

a weighted combination of arc difference Δ_A and node difference Δ_N between G_1, G_2, for weighted nodes and arcs:

Δ_A is a sum-of-squares measure of arcs differences

$$\Delta_A = \frac{1}{|E_1| + |E_2|} tr\big((A_1 - A_2)(A_1 - A_2)^T\big), \tag{4}$$

where A^T is the transpose of matrix A, $tr(A)$ the trace of A, the sum of squares of the elements on the leading diagonal. $\Delta_A \in [0,1]$ since the maximum difference is when none of the arcs have an equivalent in the other model. Then the trace is $(|E_1| + |E_2|) \cdot (1-0)^2$, which we use to scale the measure. The minimum is zero when all arcs are matched.

Δ_N is a sum-of-squares measure of nodes differences

$$\Delta_N = 1 - \frac{1}{N} tr(S \times S^T). \tag{5}$$

$\Delta_N \in [0,1]$ with maximum 1 when all N nodes correspond.

B. Comparison by Data Attributes

The previous methods compare process models using the visible information from the graphical representation, but other data attributes underlying two similarly-structured process models may be relevant when comparing the real processes.

While a mapping can be constructed between the nodes in two graphs (approximately if duplicate labels are allowed), many values for each task attribute may be associated with each node, as attributes take different values in each process instance. Comparison of nodes $v_1^{(i)}, v_2^{(j)}$ by attribute $C \in \{A_G \cup A_V\}$ is now between vectors representing samples from two underlying distributions $Pr_1^{(i)}, Pr_2^{(j)}$ over possible values of C for the two nodes. $Pr_{k \in 1,2}^{(i)}$ is the distribution of values that C can take for the task, in the process underlying G_k.

To compare nodes on a categorical attribute C_{cat} (e.g. department or user IDs), we compare distributions over the frequencies of values that C_{cat} takes using a metric such as the *Euclidean Distance*

$$d_2(Pr_1^{(i)}, Pr_2^{(j)}) = \sqrt{\sum_x \big(Pr_1^{(i)}(x) - Pr_2^{(j)}(x)\big)^2}, \tag{6}$$

where x is defined over the union of the possible values of attribute C_{cat} for nodes $v_1^{(i)}, v_2^{(j)}$. $Pr_k^{(i)}(x)$ is the sample probability $Pr_k^{(i)}(C_{cat} = x)$, estimated as the fraction of process instances passing through the node, for which C_{cat} took the value x, i.e.

$$Pr_k^{(i)}(x) = \frac{|x|}{f_k(i)}. \tag{7}$$

Now we set the elements of the similarity matrix

$$s_{cat}^{(ij)} = 1 - d_2(Pr_1^{(i)}, Pr_2^{(j)}). \tag{8}$$

We compare a continuous attribute C_{cont} (such as task duration) by calculating the mean values $\overline{c}^{(i)}, \overline{c}^{(j)}$ of the attribute for tasks $v_1^{(i)}, v_2^{(j)}$ and compare in the same way as the frequencies (section III-A). Let $c(v_k^{(i)}, m)$ be the value taken by attribute C_{cont} in the mth process instance associated with node $v_k^{(i)}$ in model G_k, $k \in \{1,2\}$. Then

$$\overline{c}^{(i)} = \frac{1}{f_k(i)} \sum_{1 \le m \le f_k(i)} c(v_k^{(i)}, m). \tag{9}$$

Now we set the elements of the similarity matrix S

$$s_{cont}^{(ij)} = 1 - \frac{\overline{c}^{(i)} - \overline{c}^{(j)}}{\overline{c}^{(i)} + \overline{c}^{(j)}}. \tag{10}$$

We use a weighted combination of these attribute similarity measures to obtain a total attribute similarity between two nodes, based on multiple attributes:

$$s^{(ij)} = \sum_{1 \le p \le q} w_p \cdot s_p^{(ij)}, \tag{11}$$

for q attributes, $0 \le w_p \le 1, \sum_{1 \le p \le q} w_p = 1$. $s_p^{(ij)}$ is the similarity of nodes $v_1^{(i)}, v_2^{(j)}$ based on the pth attribute.

C. Graph Clustering and Partitioning

We use graph partitioning to recursively split process models G_1, G_2 into logical subgraphs, calculate the metrics between pairs of subgraphs as though between full process models, and use to match pairs with the lowest distance. We recursively repeat this process on the matched subgraphs until no further partitioning is possible. In this way ‘areas’ of the models which differ can be identified, iteratively focussing to specific differences, facilitating comparison of complex process models. We also use the partitioning results to visualise differences at different levels of abstraction (section III-D).

There are many methods for graph partitioning. We used Edge Betweenness (see e.g. [16]) and Spectral Partitioning (see e.g. [17]), but since our models are relatively unstructured, characterised by nodes with high centrality, more process-specific methods may be more possible.

D. Visualisation

We visualise differences between processes using colour schemes inspired by relief maps, which use colour to depict altitude. We colour nodes, arcs and subgraphs using a palette from blue to red, set as shown in Fig.2. Colour indicates how frequencies of arcs, nodes, or parts of one model, compare to their equivalents in the other. Elements or areas in one model only, or with much higher frequency in that model, are coloured at the red end of the spectrum. Corresponding elements in the other model, towards the blue end of the spectrum. The resulting pairs of coloured models are the inverse of each other and can be considered as ‘relief’ maps visualising the frequencies. Each model shows frequencies ‘above’ and ‘below’ the corresponding parts of the other, which is metaphorically at ‘sea level’ (Fig.3).

Colouring is achieved by parsing the normalised adjacency difference $A_1 - A_2$ and node similarity S matrices, whose elements are signed. Controlling opacity and line weights enables clearer highlighting of the bigger differences. Visualisation should be user-controllable to allow exploration of the models.

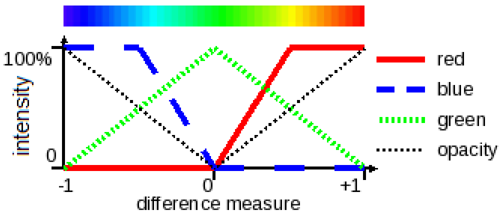


Fig. 2. Colour settings for visualisation, using colour spectrum from blue (relative low frequency of arc or node usage) through green, to red (relative high frequency). The graphs indicate the setting of the R,G,B colour components from the frequency differences between the two process models.

E. Significance and Confidence

Business processes are stochastic in that tasks and paths through the process will occur with particular probabilities. Different logs used for mining the same process are therefore random samples from an unknown underlying probability distribution over process instances, and the structure and frequencies in the mined model may vary according to the distribution of process instances in the sample log.

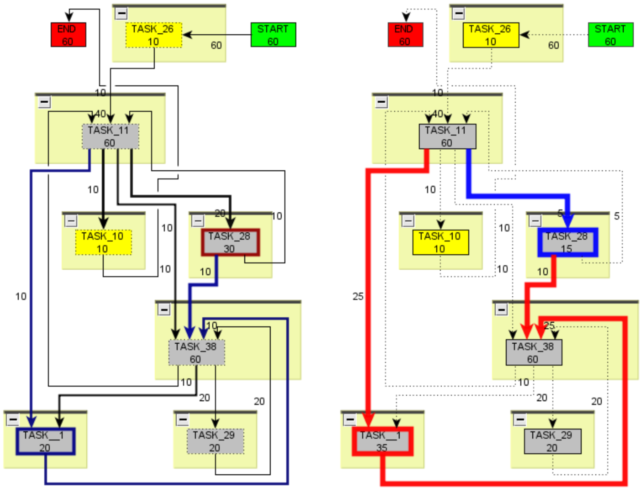
If differences are found between process models, we need to know whether the underlying processes are truly different, or whether the processes are the same and the differences in the models due to sampling variation. Since in this paper we do not know the underlying process distributions, and the modelling formalism is not probabilistic, we cannot say anything about the significance of the overall difference between the processes, beyond setting arbitrary thresholds on the metrics.

Instead, we compare individual arc and node frequencies. Assume we are comparing G_1 and G_2, mined from n_1, n_2 process instances respectively. The frequencies $g_1(ij)$ on arcs $a_1^{(ij)}$ leaving node $v_1^{(i)}$ sum[2] to the source node frequency $f_1(i)$. From these frequencies we estimate the conditional probability $p_1^{(ij)} = \frac{g_1(ij)}{f_1(i)}$ of following arc $a_1^{(ij)}$, i.e. the probability that after task $v_1^{(i)}$, the next task will be $v_1^{(j)}$.

To simplify, we take G_1 as the ground truth (i.e. as representing the true process model)[3]. Then we model the number of times the arc is followed, by a random variable π which follows a Binomial distribution with probability parameter $p_1^{(ij)}$, i.e. $\pi \sim Bin(p_1^{(ij)}, n_1)$. Then the probability ϵ under this distribution of seeing the arc frequency $g_2(ij)$ observed in the second model is

$$\epsilon = \int_{\pi=-\infty}^{\pi=g_2(ij)} Bin(p_1^{(ij)}, n_2)d\pi, \text{ if } g_2(ij) < g_1(ij),$$
$$\epsilon = \int_{\pi=g_2(ij)}^{\pi=\infty} Bin(p_1^{(ij)}, n_2)d\pi. \text{ otherwise.} \qquad (12)$$

We reject the null hypothesis h_0 that the probability of the arc in the underlying distribution for each model is unchanged, if $\epsilon < \alpha$ for some chosen threshold $0 < \alpha \ll 1$.



(a) (b)

Fig. 3. Comparison of processes using node and arc frequencies. Arcs and nodes (a) weighted and coloured to show relative frequency of use, (b) coloured red/blue to indicate significant differences at chosen significance threshold (e.g. 95%), with insignificant arcs faded out.

We treat nodes similarly, modelling the frequency on the node using a Binomial distribution, assuming that the task either occurs in a process instance or does not[4].

IV. Experimentation and Discussion

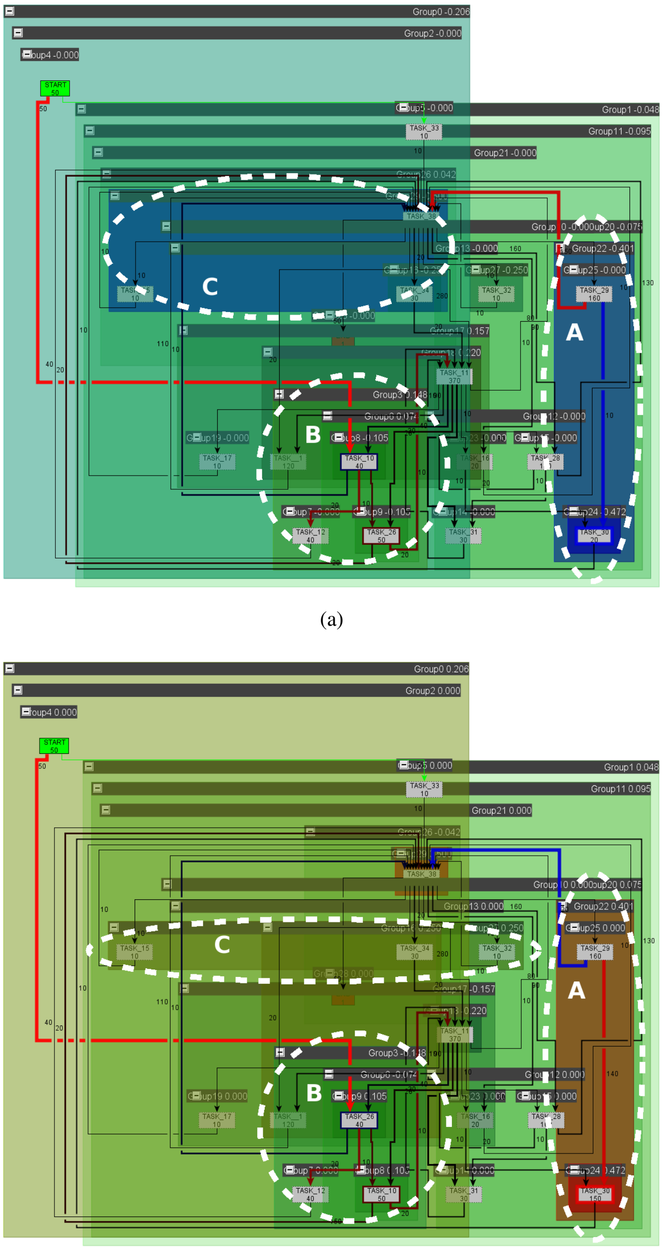
We illustrate the results produced by our method. Fig.3 shows the visualisation of frequency differences (section III-A) between two simple processes with the same structure but different frequencies (tasks 10 and 28). We show one of each pair only, since the paired models differ only in the inversion of colours. Fig.3(a) uses colouring and line width to show all differences in frequencies, e.g. for further investigation. Fig.3(b) focusses on just those differences which are unlikely to be the result of random variation, by highlighting only differences significant at the 95% threshold.

Fig.4 illustrates graph partitioning and colouring (section III-C) to 'map' the differences between two more complex processes. These are produced from artificial data, with two differences introduced, annotated as A and B. The frequency of the path between two tasks has been changed (A), and two tasks swapped (B). The colours of these areas are "more red" in one model, "more blue" in the other, along with corresponding colouring and width of the arcs and nodes involved. Unfortunately the graph partitioning algorithm has been influenced by differences in the graphs to partition them differently, leading to the spurious colouring at C. In future work we plan to develop more process-appropriate clustering and partitioning methods, for example using process patterns or centrality of nodes and directedness of these graphs.

[2] Here we assume all splits represent exclusive choice, no parallelism in the model, which would make estimation of arc probabilities more difficult.

[3] This is approximate. In reality G_1 and G_2 are both built from samples from underlying distributions, so we should calculate the joint probability of $f(a_1^{(ij)})$ and $f(a_2^{(ij)})$ being drawn from the same Binomial distribution, integrated over the distributions for all possible values of parameter $p_1^{(ij)}$.

[4] This is not so satisfactory, since the task frequency is dependent on its location on the model, and therefore may be affected by changes to probability elsewhere in the model.



(a)

(b)

Fig. 4. Partitioning and visualisation of differences between two more complex, structurally similar process models. Main areas of difference highlighted A) frequencies, B) swap of tasks 10 and 26, C) spurious (artefact of partitioning, see text).

TABLE I
NESTED GRAPH PARTITIONS AND DIFFERENCES BETWEEN MATCHED SUBGRAPHS, FOCUSSING ON THE TWO MAIN CHANGES IN FIG.4.

Swapped Tasks	**Changed Frequency**
Group 0 (difference 0.206) → 3 (0.148) → 6 (0.074) → 9 *(task 26) &* 8 *(task* 10)	Group 1 (0.048) → 11 (0.095) → 20 (0.075) → 22 (0.401) → 25 *(task 29) &* 24 *(task* 30)

Table I shows the metrics between the matched subgraphs focussing in to differences A and B. Here there is no clear relationship between the metric and the level of abstraction. Further work could also explore this relationship and develop difference measures which are more intuitive in this context.

V. Conclusion

We presented a framework for comparing complex process models, combining process comparison metrics, statistical methods, graph clustering and visualisation. We showed that this allows well-founded but flexible and intuitive comparison of process models, to understand significant differences. A fuller description and analysis will be published elsewhere.

References

[1] R. P. J. C. Bose, H. M. W. E. Verbeek, and W. M. P. van der Aalst, "Discovering hierarchical process models using prom," in *CAiSE Forum*, ser. CEUR Workshop Proceedings, S. Nurcan, Ed., vol. 734. CEUR-WS.org, 2011, pp. 33–40.

[2] G. Greco, A. Guzzo, and L. Pontieri, "Discovering expressive process models by clustering log traces," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 8, pp. 1010–1027, 2006.

[3] W. M. P. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Heidelberg Dordrecht London New York: Springer, 2011.

[4] R. P. J. C. Bose and W. M. P. van der Aalst, "Context aware trace clustering: Towards improving process mining results," in *SDM*. SIAM, 2009, pp. 401–412.

[5] M. Song, C. W. Günther, and W. M. P. van der Aalst, "Trace clustering in process mining," in *BPM Workshops*, ser. LNBIP, D. Ardagna, M. Mecella, and J. Yang, Eds., vol. 17. Springer, 2008, pp. 109–120.

[6] C. W. Günther and W. M. P. van der Aalst, "Fuzzy mining - adaptive process simplification based on multi-perspective metrics," in *BPM*, ser. LNCS, G. Alonso, P. Dadam, and M. Rosemann, Eds., vol. 4714. Springer, 2007, pp. 328–343.

[7] C. W. Günther, A. Rozinat, and W. M. P. van der Aalst, "Activity mining by global trace segmentation," in *BPM Workshops*, ser. LNBIP, S. Rinderle-Ma, S. W. Sadiq, and F. Leymann, Eds., vol. 43. Springer, 2009, pp. 128–139.

[8] B. F. van Dongen and A. Adriansyah, "Process mining: Fuzzy clustering and performance visualization," in *BPM Workshops*, ser. LNBIP, S. Rinderle-Ma, S. W. Sadiq, and F. Leymann, Eds., vol. 43. Springer, 2009, pp. 158–169.

[9] S. Pinter and M. Golani, "Discovering workflow models from activities' lifespans," *Computers in Industry*, vol. 53, no. 3, pp. 283–96, 2004.

[10] A. Rozinat and W. M. P. van der Aalst, "Conformance checking of processes based on monitoring real behavior," *Information Systems*, vol. 33, no. 1, pp. 64–95, 2008.

[11] A. K. Alves de Medeiros, W. M. P. van der Aalst, and A. J. M. M. Weijters, "Quantifying process equivalence based on observed behavior," *Data Knowl. Eng.*, vol. 64, no. 1, pp. 55–74, 2008.

[12] P. Weber, B. Bordbar, and P. Tino, "A principled approach to the analysis of process mining algorithms," in *IDEAL*, ser. LNCS, H. Yin, W. Wang, and V. J. Rayward-Smith, Eds., vol. 6936. Springer, 2011, pp. 474–481.

[13] J. Bae, L. Liu, J. Caverlee, L.-J. Zhang, and H. Bae, "Development of distance measures for process mining, discovery, and integration," *Int'l Journal of Web Serv. Res.*, vol. 4, no. 4, pp. 1–17, 2007.

[14] R. M. Dijkman, M. Dumas, B. F. van Dongen, R. Käärik, and J. Mendling, "Similarity of business process models: Metrics and evaluation," *Inf. Syst.*, vol. 36, no. 2, pp. 498–516, 2011.

[15] P. Taylor, M. Leida, and B. A. Majeed, "Case study in process mining in a multinational enterprise," *(To Appear), LNBIP*, 2012.

[16] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, p. 026113, 2004. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevE.69.026113

[17] U. von Luxburg, "A tutorial on spectral clustering," *CoRR*, vol. abs/0711.0189, 2007.

[18] S. Rinderle-Ma, S. W. Sadiq, and F. Leymann, Eds., *BPM Workshops, BPM 2009 Int'l Workshops, Ulm, Germany, 2009. Revised Papers*, ser. LNBIP, vol. 43. Springer, 2010.