A Visualization Approach for Difference Analysis of Process Models and Instance Traffic

Simone Kriglstein[1], Günter Wallner[2], and Stefanie Rinderle-Ma[3]

[1] SBA Research, Vienna, Austria
skriglstein@sba-research.at
[2] University of Applied Arts, Institute of Art and Technology, Vienna, Austria
guenter.wallner@uni-ak.ac.at
[3] University of Vienna, Faculty of Computer Science, Vienna, Austria
stefanie.rinderle-ma@univie.ac.at

Abstract. Organizations are often confronted with the task to identify differences and commonalities between process models but also between the instance traffic that presents how instances have progressed through the model. The use cases range from comparison of process variants in order to identify redundancies and inconsistencies between them to the analysis of instance traffic for the (re)design of models. Visualizations can support users in their analysis tasks, e.g., to see if and how the models and their instance traffic have changed. In this paper we present a visualization approach to highlight the differences and commonalities between two models and – if available – their instance traffic.

Keywords: Visualization, Control Flow Analysis.

1 Introduction

In the last years, the interest to develop approaches in order to support users in analyzing process models and their instances with regard to their differences and commonalities has increased. This could be partly caused by the need to manage the increasing number of process models and their instances that can accumulate in organizations over the years (cf. [12]). Therefore different ways to compare process models have been developed. One way is the use of similarity checks (see, e.g., [8,17]) for checking commonalities. Another way is to check explicitly for the differences between the process models (e.g., [7]).

According to van der Aalst [2], there exist only a few techniques (e.g., [1,9,11]) for detecting differences in process models, but because of the importance it needs more attention. For example, in addition to the analysis of process models, it is also of interest to find techniques to analyze the instance traffic (based on the executable logs or simulation data) that reflects how instances have progressed through the model in order to, e.g., see the distribution of instances over the different paths through the process model. Understanding the flow of instances helps to distinguish well-designed models from models that require modifications (e.g., to identify redundant paths because of changed conditions). Furthermore,

F. Daniel, J. Wang, and B. Weber (Eds.): BPM 2013, LNCS 8094, pp. 219–226, 2013.

the analysis of the differences between the instance traffic helps to identify trends across multiple process instances or time periods. Moreover, it allows to see if and how the traffic has changed, e.g., to detect more or less visited paths or to follow the consequences of changes in process models with respect to the execution of instances.

In this paper we present a visualization approach to highlight the difference information between process models and – if available – between the instance traffic in a single graph. The differences between the two input models and their instance traffic are visualized as difference model that merges the two input models in such a way that it allows users to visually see differences and commonalities between the two models. With the presented visualization approach, users have the possibility to analyze two models and their instance traffic to support the following tasks: 1) comparison of two process models, 2) comparison of instance traffic between two process models and 3) comparison of instance traffic of one model at different points in time.

2 Related Work

Comparing artifacts in order to detect their differences plays an important role in many application domains. For example, detecting differences in models is an essential operation in software development including version and change management, software evolution etc. in order to find problems or to detect discrepancies between the models (see, e.g., [3,13,14,16]). Especially for business processes, delta analysis is used in order to compare the differences between two models. For example, various approaches were developed that use delta analysis to compare predefined process models with discovered models derived from event logs (e.g., [1,9,11]).

For the representation of differences, color-coding is often used to highlight which nodes and edges were added or removed from a graph (see, e.g., [5,6,10,13,14]). In contrast, Andrews et al. [4] use color-coding in such a way that each of the two input models is associated with a single color. The difference model is a superposition of the two input models in order to highlight differences and commonalities between both models. The coloring of the nodes in the calculated difference model depicts in which input model the node is present. If a node is present in both input models the node is two colored. In contrast to our approach, their approach requires to specify node similarities a priori before the difference model can be calculated.

3 Basic Concepts

In this paper, we focus on a visualization concept for directed connected graphs in order to provide a basis for existing business process modeling and execution notations such as Event-driven Process Chains (EPC), UML Activity Diagrams, and the Business Process Modeling Notation (BPMN). For special concepts of certain languages corresponding extensions might become necessary.

At this point we should also emphasize that we are looking at the processes from the control flow perspective.

3.1 Difference Model

We define a process model as a directed connected graph $PM = (N, E \subseteq N \times N)$, where N is a set of nodes and E is a set of directed control edges. Each node $n \in N$ is described by a 3-tuple (id, l, t) where id is a unique identifier, l is the label and t is the type of the node. Different business modeling languages like BPMN or EPC distinguish between different types of nodes. For example, BPMN differentiates between activity nodes, event nodes, and gateway nodes for the control flow graph. For the sake of simplicity, we restrict the following discussion to activity and gateway nodes.

A process model contains one start node and one end node. Nodes are connected in such a way that each node is on a path from the start point to the end point. The start node has no incoming edge and the end node has no outgoing edge. Let $PM_1 = (N_1, E_1)$ and $PM_2 = (N_2, E_2)$ be the two process model to be compared, then the *difference model*[1] can be defined as

$$\begin{aligned} DM = PM_2 - PM_1 &:= (N_d, E_d, M_{N_d}, M_{E_d}) = \\ &= (N_2 \cup N_1, E_2 \cup E_1, M_{N_d}, M_{E_d}) \end{aligned} \tag{1}$$

where M_{N_d} and M_{E_d} describe the node and edge markings of DM with $M_{N_d} : N_d \mapsto \{-1, 0, 1\}$ and $M_{E_d} : E_d \mapsto \{-1, 0, 1\}$. For a node $n \in N_d$ its marking is determined as follows:

$$m_n = \begin{cases} 0, & \text{if } n \in N_1 \wedge n \in N_2. \\ 1, & \text{if } n \in N_2 \wedge n \notin N_1. \\ -1, & \text{otherwise.} \end{cases} \tag{2}$$

The same applies for the marking of a control edge $e \in E_d$:

$$m_e = \begin{cases} 0, & \text{if } e \in E_1 \wedge e \in E_2. \\ 1, & \text{if } e \in E_2 \wedge e \notin E_1. \\ -1, & \text{otherwise.} \end{cases} \tag{3}$$

The markings are used to distinguish between *add* and *delete* change operations. The markings also indirectly cover some other change operations like *move* (moving a node will remove it from the old location and add it at a new location in the model), but currently we do not account for them explicitly in the visualization.

3.2 Instance Traffic

If the control flow of process instances – either on the same process model or two different process models – should be compared the above concept can be extended by considering how often control edges have been executed by individual

[1] Mathematically, this can be considered as merging PM_1 and PM_2. However, with the term *difference model* we want to emphasize that the merged model reflects the *differences* between PM_1 and PM_2.

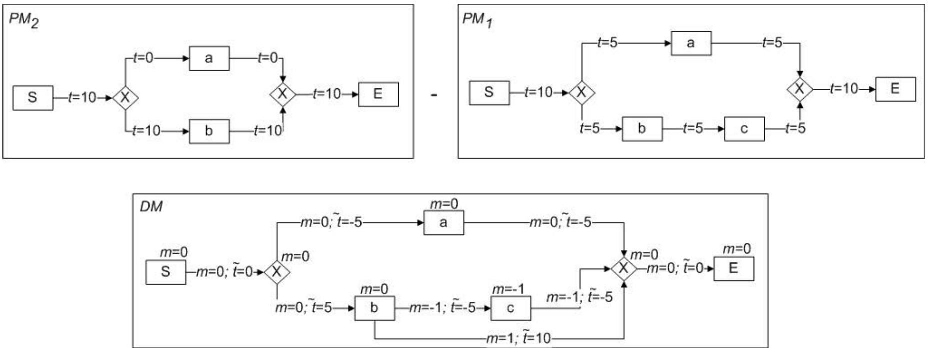

Fig. 1. An example to illustrate the instance traffic for PM_1 with $k_1 = 10$ and PM_2 with $k_2 = 10$ and the relative traffic for DM with its corresponding marking information

instances (in the following referred to as *instance traffic*) to observe the distribution of instances over the different paths through the process model. For a given process model $PM = (N, E)$ with a set of instances $\mathcal{I}$, $|\mathcal{I}| = k$ executed on PM, the instance traffic $t(e)^{PM}$ for a control edge $e \in E$ is given by the number of instances which passed e during the execution. Please note that $t(e)^{PM}$ can be greater than k if there are loops in PM which may cause an instance to pass e several times.

However, when comparing two sets of instances $\mathcal{I}_1$ executed on PM_1 and $\mathcal{I}_2$ executed on PM_2 usually $|\mathcal{I}_1| \neq |\mathcal{I}_2|$ will be different. Calculating the difference directly from the traffic values of the individual process models would therefore skew the result in favor of one of the two process models being compared. We therefore equalize the traffic by weighting the traffic of PM_2 with $\delta = k_1/k_2$. The *relative* traffic $\tilde{t}(e)^{DM}$ of $e \in E_d$ in the difference model DM is then given by

$$\tilde{t}(e)^{DM} = \delta t(e)^{PM_2} - t(e)^{PM_1} \quad (4)$$

where $t(e)^{PM_1} = 0$ if $e \notin E_1$, $t(e)^{PM_2} = 0$ if $e \notin E_2$.

To illustrate these concepts the example in Figure 1 shows the instance traffic for two input models and the relative traffic in the corresponding difference model ($k = 10$ for both input models) with the corresponding marking information. The node S is the start point and the node E is the end point. In this example, the difference between PM_1 and PM_2 is the node c and the edges from b to c, c to gateway node x, and b to x. The instance traffic in PM_1 shows that the instances split into two halves after the XOR split (presented by the gateway node x), but in PM_2 all instances go across b. The difference model DM allows to observe these changes of the instance traffic between PM_1 and PM_2 in a single graph. For instance, the relative traffic in the difference model shows that the traffic has decreased for the path via a and increased for the path x→b→x.

Table 1. Description of the visual elements which can occur in the difference model

	Meaning	Short Description
Representation of Activities/Gateways/Edges		
	No Change	Black is used to highlight all nodes $n \in N_d$ where the marking $m_n = 0$. The same applies for an edge $e \in E_d$ with $m_e = 0$.
	Only in PM_2	Green is used to present all $n \in N_d$ where $m_n = 1$. The same applies for an edge $e \in E_d$ with $m_e = 1$.
	Only in PM_1	Gray is used to visualize all $n \in N_d$ where $m_n = -1$. The same applies for an edge $e \in E_d$ with $m_e = -1$.
Representation of Instance Traffic between Activities/Gateways		
	No Change	Blue is used to present the instance traffic between nodes if the traffic $t(e)^{PM_1} = \delta t(e)^{PM_2}$ of $e \in E_d$.
	Increased Traffic	Green is used if the traffic $\delta t(e)^{PM_2} > t(e)^{PM_1}$ of $e \in E_d$.
	Increased Traffic *(New Edge)*	Light green is used if the traffic $\delta t(e)^{PM_2} > 0 \wedge e \in E_2 \wedge e \notin E_1$ to highlight that the instance traffic increased due to the addition of e.
	Decreased Traffic	Red is used to highlight the instance traffic if the traffic $\delta t(e)^{PM_2} < t(e)^{PM_1}$ of $e \in E_d$.
	Decreased Traffic *(Removed Edge)*	Orange is used if the traffic $t(e)^{PM_1} > 0 \wedge e \in E_1 \wedge e \notin E_2$ to highlight that instance traffic decreased due to the removal of e.

4 Visualization Design and Implementation

For the visualization of the two input models and the difference model, we use a node-link representation. For the left-to-right arrangement of the nodes a Sugiyama-style layouter [15] is used. Activities are displayed as rectangular nodes and gateways as diamond shaped nodes. Color-coding is used to highlight the changes between the process models and between their instance traffic (cf. Table 1).

The thickness d of an arrow depicting the instance traffic in an input model along edge e is given by $\Delta d \cdot t(e)^{PM}$ where Δd is a user-changeable parameter. However, in case of the difference model we have to take care of the special case that $\tilde{t}(e)^{DM} = 0$ due to the adjusted instance traffic being equal in both input models. To be able to distinguish this case (and to highlight this fact in the visualization) from the case where the difference is zero because no traffic occurred along e in PM_1 as well as PM_2, the thickness in the difference model is given by

$$d = \begin{cases} d_{min} & \text{if } t(e)^{PM_1} = \delta t(e)^{PM_2} \\ \Delta d \cdot \tilde{t}(e)^{DM} & \text{otherwise.} \end{cases} \qquad (5)$$

As proof of concept, we implemented a C# prototype (see Figure 2) which allows the user to load two process models that should be compared in .xml format.

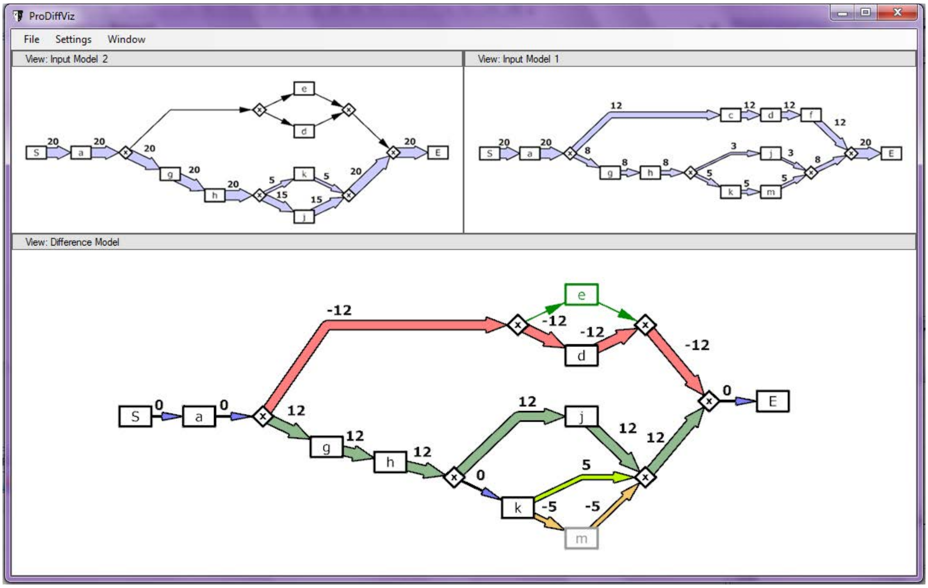

Fig. 2. Interface of the prototype. The two input models are shown on the top and the difference model at the bottom.

Multiple views are used to present the two input models and their difference model at the same time. Furthermore, users have the possibility to simulate the execution of a certain number of instances on a process model. If instances are simulated, this information is automatically considered when calculating the difference model. Options for filtering allow the user to hide or show the different types of edges (e.g., only showing edges which have been added or removed). This can be useful to reduce visual clutter especially for larger process models.

5 Use Case

Organizations are often confronted with the need to adapt their business process to react to new or changed environmental conditions (e.g., requirements of customers changed). The comparison of the different process versions and their instance traffic helps to analyze the impact of such changes. For example, the process model in Figure 3 on the right side shows that only the lower path of the process was executed whereas the upper path was never executed. A simulation of the changed process model shows the new distribution of instances over the different paths (cf. Figure 3, left side). The difference model highlights how the instance traffic changed between the two process versions. For example, the instance traffic increased for the upper path but decreased for the lower one. A reason is that the path with the increased instance traffic, has never been executed in the old version. The decreased instance traffic was caused by a different distribution of instances. A reason could be that some of the activities

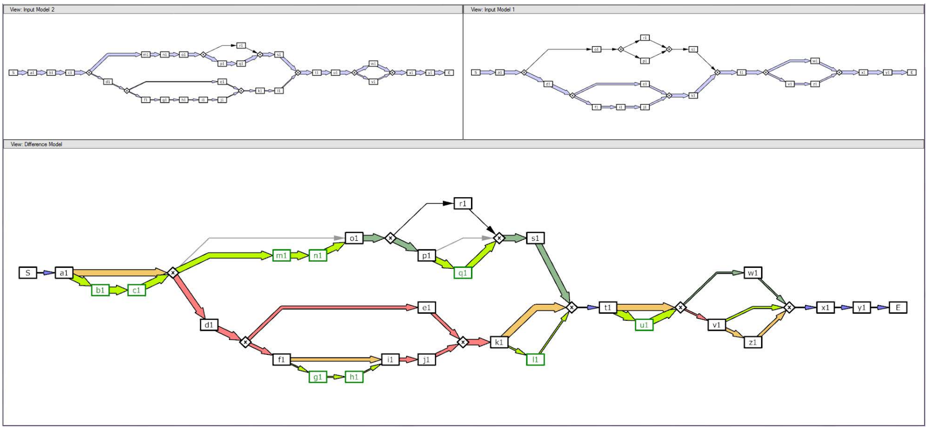

Fig. 3. Use Case: Comparison of instance traffic between two process versions

may not be well-suited for the adapted process model anymore. The difference visualization of instance traffic makes the effects of the changes visible and can support users in their design decisions (e.g., if a redesign of the process model is necessary or not).

6 Conclusion

The interest to develop approaches for the identification of differences and commonalities between process models and instances has increased in the last years. In this paper we presented a visualization approach with the goal to highlight differences but also commonalities to support the following tasks: comparison of two process models, the comparison of instance traffic between two process models and the comparison of instance traffic of one model at different points in time.

Acknowledgements. The research was funded by COMET K1, FFG - Austrian Research Promotion Agency.

References

1. van der Aalst, W.M.P.: Business alignment: Using process mining as a tool for delta analysis and conformance testing. Requir. Eng. 10(3), 198–211 (2005)
2. van der Aalst, W.M.P., Weijters, A.J.M.M.: Process mining: a research agenda. Comput. Ind. 53(3), 231–244 (2004)
3. Alanen, M., Porres, I.: Difference and union of models. In: Stevens, P., Whittle, J., Booch, G. (eds.) UML 2003. LNCS, vol. 2863, pp. 2–17. Springer, Heidelberg (2003)
4. Andrews, K., Wohlfahrt, M., Wurzinger, G.: Visual graph comparison. In: Proc. of the 13th Int. Conf. Information Visualisation, pp. 62–67. IEEE (2009)

5. Archambault, D.: Structural differences between two graphs through hierarchies. In: Proc. of Graphics Interface, pp. 87–94. GI, Canadian Information Processing Society (2009)
6. van den Brand, M., Protić, Z., Verhoeff, T.: Generic tool for visualization of model differences. In: Proc. of the 1st Int. Workshop on Model Comparison in Practice, pp. 66–75. ACM Press (2010)
7. Delugach, A., de Moor, H.: Difference graphs. In: Meersman, R., Tari, Z. (eds.) Proc. of Common Semantics for Sharing Knowledge: Contributions to the 13th Int. Conf. on Conceptual Structures, pp. 41–53. ICCS (2005)
8. Dijkman, R., Dumas, M., van Dongen, B., Käärik, R.R., Mendling, J.: Similarity of business process models: metrics and evaluation. Information Systems 36(2), 498–516 (2011)
9. Esgin, E., Senkul, P.: Delta analysis: A hybrid quantitative approach for measuring discrepancies between business process models. In: Corchado, E., Kurzyński, M., Woźniak, M. (eds.) HAIS 2011, Part I. LNCS, vol. 6678, pp. 296–304. Springer, Heidelberg (2011)
10. Geyer, M., Kaufmann, M., Krug, R.: Visualizing differences between two large graphs. In: Brandes, U., Cornelsen, S. (eds.) GD 2010. LNCS, vol. 6502, pp. 393–394. Springer, Heidelberg (2011)
11. Kleiner, N.: Delta analysis with workflow logs: aligning business process prescriptions and their reality. Requirements Engineering 10, 212–222 (2005)
12. La Rosa, M., Dumas, M., Uba, R., Dijkman, R.: Merging business process models. In: Meersman, R., Dillon, T.S., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6426, pp. 96–113. Springer, Heidelberg (2010)
13. de Moor, H., Delugach, A.: Software process validation: comparing process and practice models. In: Proc. of the 11th Int. Workshop on Exploring Modeling Methods for Systems Analysis and Design (EMMSAD 2006) held in conjunction with the 18th Conf. on Advanced Information Systems (CAiSE 2006), pp. 533–540 (2006)
14. Ohst, D., Welle, M., Kelter, U.: Differences between versions of UML diagrams. In: Proc. of the 9th European Software Engineering Conf. held jointly with 11th ACM SIGSOFT Int. Symposium on Foundations of Software Engineering, ESEC/FSE, pp. 227–236. ACM Press (2003)
15. Sugiyama, K., Tagawa, S., Toda, M.: Methods for visual understanding of hierarchical system structures. IEEE Trans. Systems, Man and Cybernetics 11(2), 109–125 (1981)
16. Treude, C., Berlik, S., Wenzel, S., Kelter, U.: Difference computation of large models. In: Proc. of the the 6th Joint meeting of the European Software Engineering Conf. and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC-FSE, pp. 295–304. ACM Press (2007)
17. Yan, Z., Dijkman, R., Grefen, P.: Fast business process similarity search. Distributed and Parallel Databases 30, 105–144 (2012)