Detecting Semantic Business Process Model Clones

Thomas S. Heinze[1], Wolfram Amme[2], and André Schäfer[2]

[1] German Aerospace Center (DLR)
thomas.heinze@dlr.de
[2] Friedrich Schiller University Jena
[wolfram.amme,andre.schaefer]@uni-jena.de

Abstract. Process modeling with languages like BPMN allows process designers to create the same business process model in various ways. Detecting model clones, i.e., pairs of business process models sharing a certain degree of similarity can be difficult. In this paper, we propose an approach to process model clone detection based upon dominator trees and targeted at detecting semantically though not necessarily syntactically similar process models of business processes.

1 Introduction

Duplicated process models is a common issue in business process management and modeling and in particular relevant when process models are organized in model repositories [11]. Matching business process models and estimating their similarity has thus been an important research topic and has many applications, ranging from analyzing conformance to reference models [3], tracing and identification of process variants [2] to process model search and clone detection [1]. In this paper, we address the latter problem of finding process models which share a certain degree of similarity, which is known as model clone detection in the literature [11,12]. Note that such model clones can origin from homologous development [14], where a process designer reuses and modifies an existing model to generate a new process model. As a result, the original and new model usually share a high lexical similarity. In contrast, in heterologous development [14], process models are created independently of each other but implementing similar functionalities. Thus, such models are not necessarily syntactically similar, i.e., can differ in the number of activities and gateways or in their structure. Finding such semantic clones is in particular hard for business process models due to the large number of modeling purposes and practices, e.g., block- and graph-oriented process modeling styles. While differing modeling styles can be found in different business process modeling languages, multiple paradigms can even exist in the very same language. A well-known representative for such a language is BPMN.

As an example, consider the two BPMN process models shown on the left-hand side of Fig. 1. Apparently, both process models implement a similar business process. In particular, in terms of possible execution traces, the lower model's

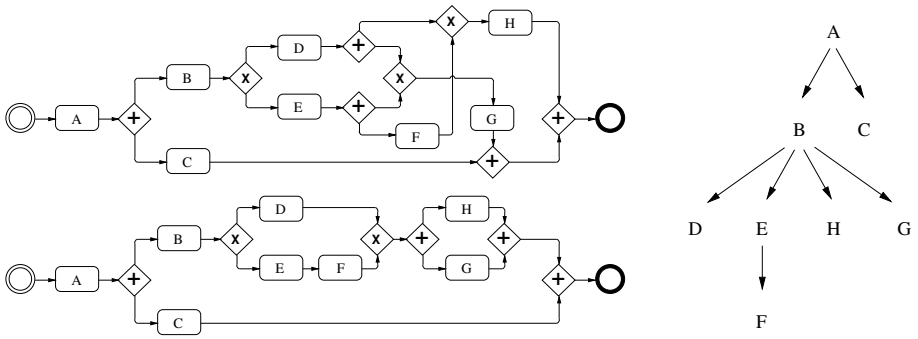

Fig. 1. Business process models (left) and dominator tree (right)

behavior is a subset of the upper model's behavior (activity G can precede activity F in the upper model in contrast to the lower model). Estimating the models' similarity and therefore identifying them as model clones is though not straightforward due to the more graph-oriented modeling style of the upper model and the more block-oriented style of the lower model. Such a pair of process models is known as a semantic model clone, i.e., two process models which have similar behavior but do not necessarily share the same syntax and structure [12].

2 Process Model Clone Detection

In order to identify semantic model clones, we propose a model clone detection method based on dominator trees [10]. Dominator trees provide for an abstraction of process control flow and therefore encode the dominance relation of flow graphs. A node n in a flow graph is said to dominate another node m, iff n comes before m on every path from the flow graph's start node to m. Note that this relation can be efficiently computed for flow graphs using data flow analysis, also for workflow graphs [4,8]. Each node and its immediate, i.e., closest dominating node then results in an edge in the dominator tree. The same dominator tree represents both the example models and is shown on the right-hand side of Fig. 1. As can be seen, the dominator tree comprises guaranteed happens-before relations between the models' activities, while abstracting from the process models' structure.

In order to allow for an efficient comparison of process models, dominator trees are encoded into sets of integer sequences. In this way, standard metrics, e.g., Hamming or edit distance, can be used for their comparison. If the thus computed difference between two process models does not exceed a certain threshold value, the models are considered to be a model clone. For the encoding, each path of a dominator tree, starting at its root node and ending at a leaf node, is mapped to a sequence of integers, yielding a set of integer sequences. For the dominator tree in Fig. 1, this would mean to encode the five paths $[A, B, D]$, $[A, B, E, F]$, $[A, B, H]$, $[A, B, G]$, and $[A, C]$. As the two example process models share the same dominator tree shown in Fig. 1, they are identified as model clone. Also

considering modifications like adding, modifying, or removing a single or a small number of nodes in one of the models, as in case of homologous development, would only slightly affect their encoding and the result of their comparison.

Control flow is an integral part of a business process model, but its other aspects have to be considered as well. While the encoding of dominator trees introduced above does not cover the analysis of node labels by itself, stemming, bag-of-words, word embeddings, and other related methods [6,11] can be integrated into our approach. In this way, the encoding becomes more permissive in the presence of differing node labels, as is common in business process models. Furthermore, the pre-processing of node labels by means of Static Single Assignment Form [4,8] helps in including aspects of process data in the encoding [4]. The thorough study of an optimal encoding is subject to our future work.

3 Related Work

Business process model similarity estimation and matching has been a frequent topic in research. Due to space constraints, we refer to the survey in [11] for a comprehensive overview. Respective approaches can be categorized into: syntactical, structural, behavioral methods and approaches based upon human judgement. Syntactical methods compare process models based on the therein used labels, e.g., as in [6]. Structural methods use process models as graphs for estimating their similarity, e.g., calculating the graph edit distance or isomorphisms [1]. Behavioral methods instead use process execution traces or logs to compare process models, e.g., measuring the longest common subseqence [3]. Note that our proposed method aligns between both, as it does not rely on the actual process modeling structure. In this way, we avoid the usually costly analysis of process behavior but are still able to abstract from modeling styles and practices. The comparison with behavioral methods and analysis of the tradeoff between precision and scalability to large process model repositories is subject to future work. In a way, our method reminds of causal footprints [2] or behavioral profiles and footprints [7,13]. Remember that dominator trees encode the guaranteed happens-before relation of activities. Behavioral profiles and footprints are based on execution traces and the direct or eventual successor relation of activities therein, respectively. Measuring similarity is then conducted using the relations' Jaccard index for two process models, or an execution log and a process model. In general, the major part of research on clone detection addresses source code. However, some approaches have been considering clone detection for model-based languages besides business process modeling languages, e.g., UML [12].

4 Next Steps

We are interested in implementing the proposed method in a system for detecting semantic BPMN model clones, e.g., starting with the existing control flow analyses available in mojo [9]. The implementation would allow us for experimenting with various optimizations and parameters, including differing metrics and encodings.

Furthermore, the thorough evaluation and comparison with state-of-the-art approaches to process model clone detection and similarity estimation, in particular the trace-based approaches mentioned above, is another item for future work. Such an evaluation requires though a dataset with ground truth, i.e., known process model clones. As such datasets are unfortunately scarce, we plan to resort to human judges for generating missing labels in an existing dataset, using for example the set of mined process models in [5]. Alternatively, applying small mutations on seed models can be used to generate a dataset, similar to [7].

References

1. Dijkman, R.M., Dumas, M., van Dongen, B.F., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. Inf. Syst. **36**(2), 498–516 (2011)
2. van Dongen, B., Dijkman, R., Mendling, J.: Measuring Similarity between Business Process Models. In: CAiSE 2008. LNCS, vol. 5074, pp. 450–464. Springer (2008)
3. Gerke, K., Cardoso, J.S., Claus, A.: Measuring the Compliance of Processes with Reference Models. In: OTM 2009. LNCS, vol. 5870, pp. 76–93. Springer (2009)
4. Heinze, T.S., Amme, W., Moser, S.: Static analysis and process model transformation for an advanced business process to Petri net mapping. Softw. Pract. Exp. **48**(1), 161–195 (2018)
5. Heinze, T.S., Stefanko, V., Amme, W.: Mining BPMN Processes on GitHub for Tool Validation and Development. In: BPMDS/EMMSAD@CAiSE 2020. LNBIP, vol. 387, pp. 193–208. Springer (2020)
6. Klinkmüller, C., Weber, I., Mendling, J., Leopold, H., Ludwig, A.: Increasing Recall of Process Model Matching by Improved Activity Label Matching. In: BPM 2013. LNCS, vol. 8094, pp. 211–218. Springer (2013)
7. Kunze, M., Weidlich, M., Weske, M.: Behavioral Similarity – A Proper Metric. In: BPM 2011. LNCS, vol. 6896, pp. 166–181. Springer (2011)
8. Lee, J., Midkiff, S.P., Padua, D.A.: Concurrent Static Single Assignment Form and Constant Propagation for Explicitly Parallel Programs. In: LCPC 1997. LNCS, vol. 1366, pp. 114–130. Springer (1997)
9. Prinz, T.M., Spieß, N., Amme, W.: A First Step towards a Compiler for Business Processes. In: CC 2014. LNCS, vol. 8409, pp. 238–243. Springer (2014)
10. Schäfer, A., Amme, W., Heinze, T.S.: Detection of Similar Functions Through the Use of Dominator Information. In: ACSOS 2020 Comp. pp. 206–211. IEEE (2020)
11. Schoknecht, A., Thaler, T., Fettke, P., Oberweis, A., Laue, R.: Similarity of Business Process Models – A State-of-the-Art Analysis. ACM Comput. Surv. **50**(4), 52:1–52:33 (2017)
12. Störrle, H.: Towards Clone Detection in UML Domain Models. Softw. Syst. Model. **12**(2), 307–329 (2013)
13. Weidlich, M., van der Werf, J.M.: On Profiles and Footprints – Relational Semantics for Petri Nets. In: Petri Nets 2012. LNCS, vol. 7347, pp. 148–167. Springer (2012)
14. Wu, M., Wang, P., Yin, K., Cheng, H., Xu, Y., Roy, C.K.: LVMapper: A Large-Variance Clone Detector Using Sequencing Alignment Approach. IEEE Access **8**, 27986–27997 (2020)