Detecting Approximate Clones in Process Model Repositories with Apromore

Chathura C. Ekanayake[1], Felix Mannhardt[2⋆], Luciano García-Bañuelos[3], Marcello La Rosa[1], Marlon Dumas[3], and Arthur H.M. ter Hofstede[1,4]

[1] Queensland University of Technology, Australia
[2] Bonn-Rhein-Sieg University of Applied Sciences, Germany
[3] University of Tartu, Estonia
[4] Eindhoven University of Technology, The Netherlands

Abstract. Approximate clone detection is the process of identifying similar process fragments in business process model collections. The tool presented in this paper can efficiently cluster approximate clones in large process model repositories. Once a repository is clustered, users can filter and browse the clusters using different filtering parameters. Our tool can also visualize clusters in the 2D space, allowing a better understanding of clusters and their member fragments. This demonstration will be useful for researchers and practitioners working on large process model repositories, where process standardization is a critical task for increasing the consistency and reducing the complexity of the repository.

1 Overview of the tool

Identification and analysis of similar process fragments, aka *approximate clones*, is a major step in business process standardization initiatives, where similar process fragments can be replaced with standardized fragments to reduce differences across different organizational units, products or brands. In order to offer concrete support to such process standardization initiatives, we developed a tool that allows analysts to identify, cluster, analyze and visualize approximate clones.

The tool is part of the Apromore advanced process model repository [5, 3]. The purpose of Apromore goes beyond that of simple model storage. Apromore aims to provide a one-stop place for the research community to expose algorithms and techniques that operate over (large) process model collections. Examples of techniques that have already been implemented are process similarity search [1] and process merging [4]. An advantage of being integrated into Apromore, is that the tool exploits Apromore's *canonical process format*, an independent format used for internal process representation. All process models imported into Apromore are converted into this internal format. Doing so, approximate clones can be detected in process models defined in different modeling languages such as BPMN, EPC, PNML, etc.

Apromore is a SaaS reachable via the Web. The functions offered by the approximate clone detection tool are available through Apromore's Web interface (the Apromore portal), as well as via Web service operations. The Apromore portal consumes

⋆ Work done while visiting Queensland University of Technology, Australia

these operations itself, but they can also be consumed by external applications (e.g. the WoPeD tool[5] – a Petri net editor – can connect to Apromore).

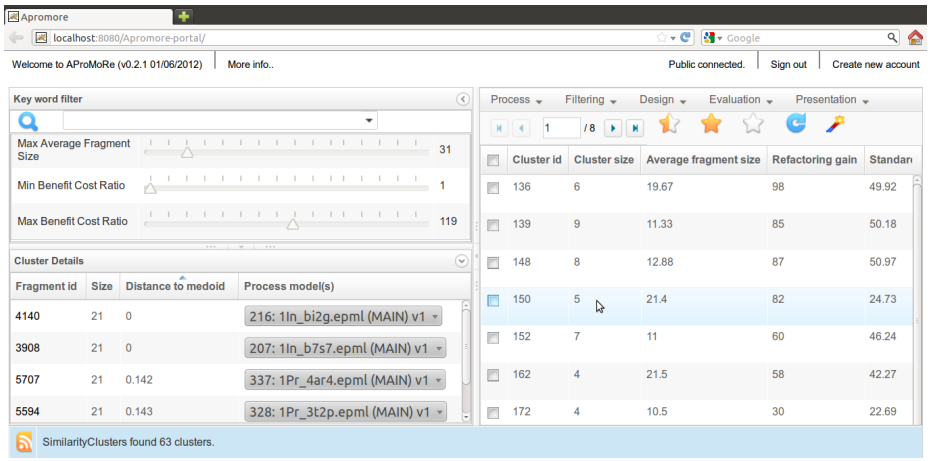

Fig. 1. Web interface of the approximate clone detection tool in Apromore

The Web interface of the approximate clone detection tool (shown in Fig. 1) provides features for creating, browsing and visualizing fragment clusters. Users can select one or more process models, specify the clustering parameters (such as the preferred clustering algorithm), and kick off the clustering. Once the fragments included in the selected process models have been clustered, users can apply different filtering criteria (e.g. on the size of the clusters) and browse the resulting clusters in a detailed list view. Another useful feature is the visualization of clusters in the 2D space. The visualization component (shown in Fig. 2) displays each fragment in a cluster as a point in the space and positions fragments within a cluster according to their distances to the medoid (distances being represented as edges between the points). It also positions the clusters in the space according to the GEDs among their medoids. One can also click on the point corresponding to a process fragment and visualize its corresponding model using any process modeling language supported by Apromore (e.g. EPCs, BPMN).

Under the hoods, the approximate clone detection tool relies on three techniques that have also been integrated into Apromore: i) RPST, ii) RPSDAG and iii) graph-edit distance. The RPST algorithm [6] is used to decompose each process model into a set of Single-Entry Single-Exit (SESE) process fragments. Such decomposed process fragments and their parent-child relationships are stored in the RPSDAG [7], an indexing structure which captures the union of the RPSTs of all process models by identifying cloned process fragments. This information about fragments and their parent-child relationships is used by a clustering algorithm to identify meaningful clusters.

[5] www.woped.org

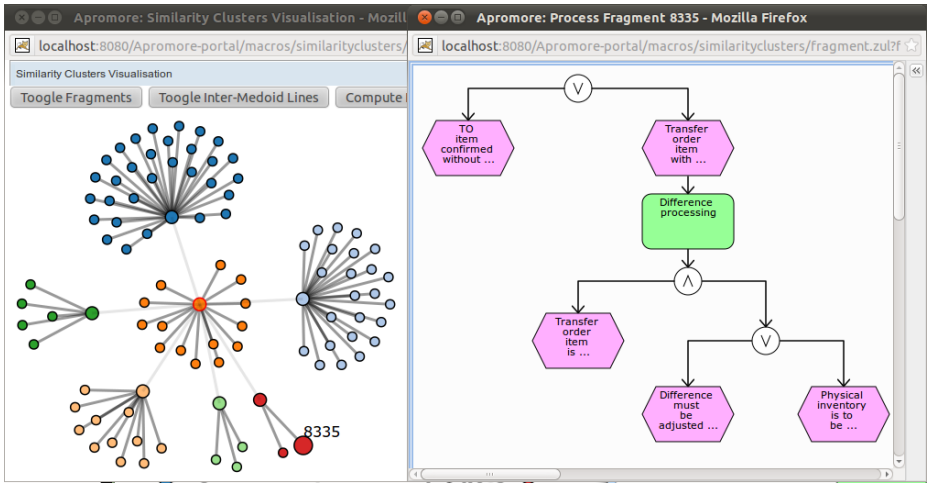

Fig. 2. Cluster visualization component of the approximate tool detection tool

Clustering algorithms need a distance measure between data objects (i.e. process fragments) in order to identify clusters. Our tool uses the graph edit distance (GED) defined in [1] as the distance measure between process fragments: this metric measures the distance between two process models (or fragments) based on a combination of structural and node labels similarity. All pairwise GEDs need be computed before a clustering algorithm can be invoked. As a process model repository can contain a large number of fragment pairs the GED calculation can be expensive. To overcome this problem, we employ several optimizations that speed up the computation. One such optimization is to exploit the RPSDAG structure to avoid the calculation of GEDs between fragments in the same hierarchy, as we do not want to have two fragments in a cluster if one fragment contains the other fragment. Once GED values are calculated, these are stored in Apromore so that users can efficiently test various clustering options using different combinations of parameters.

Clustering is performed by using data clustering algorithms. The tool features a modified version of the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm and the Hierarchical Agglomerative Clustering (HAC) algorithm. However, any suitable clustering algorithm can be integrated with the tool. Once the repository has been clustered, clusters are analyzed by computing their medoid (the fragment which is the closest to the cluster center) if this is not yet available, average fragment sizes, benefit-cost ratios in terms of standardization, etc.

2 Maturity and Significance

Our tool contributes a novel method for identifying and analyzing similar fragments of possibly different process models. The approach is innovative as the application of

clustering algorithms for approximate clone detection in process models has not been studied yet. The main features of the tool are:

- Identify similar fragment groups using the HAC algorithm and a modified DBSCAN algorithm;
- Perform clustering / re-clustering operations very efficiently, in order to facilitate fine-tuning of clustering parameters;
- Analyze identified clusters to generate information useful for determining costs and benefits of standardization;
- Support filtering and browsing of clusters and their member fragments;
- Visualize clusters in the 2D space based on the GEDs between medoids and fragments;
- Visualize single fragments in various process modeling formats (e.g. BPMN, EPCs) for in-depth review.

The tool has been used to identify approximate fragment clones in two industrial process model collections: the SAP R/3 collection and a process model collection of a large insurance company under condition of anonymity. The SAP dataset contains 595 process models with sizes ranging from 5 to 119 nodes, and including 2,348 non-trivial fragments; the insurance company dataset contains 363 process models ranging from 4 to 461 nodes and including 2,037 non-trivial fragments. Both collections were clustered using our tool by using both the supported clustering algorithms and trying various clustering parameters. Once GED values were computed, the clustering phase was executed in a very short time (less than 4 seconds). Working on such short times, it was possible to experiment with different configuration parameters. The tool identified a large number of clusters (ranging from 243 to 364 clusters) from both collections using the two supported clustering algorithms. Some identified clusters contained fragments with minor differences (e.g. spelling mistakes in task labels), while some clusters contained similar fragments with more interesting differences (e.g. additional tasks, substitution of a task with a different one, additional branches, etc.). These clusters, and especially those from the latter group, could be useful for standardizing similar business processes, e.g. those that originate from copy/pasting followed by independent modifications to the copied fragments. The details of this study are available in [2].

In addition to the above studies, our tool was extensively evaluated with artificially generated datasets, to determine the accuracy of the tool in terms of precision, recall and weighted average FScore (the latter is a measure of the quality of a clustering algorithm). For this purpose, we built an evaluation framework that generates groups of similar process fragments (i.e. fragment clusters) taking content from the two industrial datasets, and integrate these fragments into separately generated process models (again, taking content from the two industrial datasets). Then those process models were given as input to our tool, which computed the clusters of approximate fragment clones and compared these with the artificially generated clusters. Average recall and precision were high, ranging from 0.71 to 0.82 (recall) and 0.84 to 0.89 (precision). The weighted average FScore was also high ranging from 0.73 to 0.77. Details of these experiments are documented in [2].

The Apromore repository can be accessed from the Apromore Web-site.[6] The source code of the approximate clone detection tool is distributed under the LGPL license along with the Apromore source code.[7]

A screencast of the tool, showcasing its main features, is available at `http://www.screenr.com/ZTn8` and `http://www.screenr.com/VTn8`.

Acknowledgments We thank all contributors to the Apromore initiative, in particular Marie-Christine Fauvet and Cameron James for their work on the implementation. This research was carried out as part of the activities of, and funded by, the Smart Services Cooperative Research Centre (CRC) through the Australian Government's CRC Programme (Department of Innovation, Industry, Science and Research).

References

1. R.M. Dijkman, M. Dumas, and L. García-Bañuelos. Graph matching algorithms for business process model similarity search. In *BPM*, volume 5701 of *LNCS*. Springer, 2009.
2. C.C. Ekanayake, M. Dumas, L. García-Bañuelos, M. La Rosa, and A.H.M. ter Hofstede. Approximate clone detection in repositories of business process models. In *BPM*. Springer, 2012.
3. M.C. Fauvet, M. La Rosa, M. Sadegh, A. Alshareef, R.M. Dijkman, L. Garcia-Banuelos H.A. Reijers, W.M.P. van der Aalst, M. Dumas, and J. Mendling. Managing Process Model Collections with APROMORE. In *Proceedings of Service-Oriented Computing (ICSOC 2010)*, volume 6470, 2010.
4. M. La Rosa, M. Dumas, R. Uba, and R.M. Dijkman. Business Process Model Merging: An Approach to Business process Consolidation. *ACM Transactions on Software Engineering and Methodology*, 2012 (to appear).
5. M. La Rosa, H.A. Reijers, W.M.P. van der Aalst, R.M. Dijkman, J. Mendling, M. Dumas, and L. García-Bañuelos. APROMORE: An Advanced Process Model Repository. *Expert Systems With Applications*, 38(6), 2011.
6. A. Polyvyanyy, J. Vanhatalo, and H. Völzer. Simplified Computation and Generalization of the Refined Process Structure Tree. In *WSFM*, 2010.
7. R. Uba, M. Dumas, L. García-Bañuelos, and M. La Rosa. Clone detection in repositories of business process models. In *BPM*, pages 248–264, 2011.

[6] `http://apromore.org`
[7] `http://code.google.com/p/apromore`