

A framework for multi-level semantic trace abstraction

Manuel Striani¹

¹ PhD Candidate

Department of Computer Science, University of Torino
Corso Svizzera 185, 10149 Torino, Italy
striani@di.unito.it

Keywords: Abstraction, Trace comparison, Process discovery, Stroke management

Extended abstract

1 Introduction

This PhD thesis describes the development of a framework able to compare process traces, represented at different levels of abstraction. The framework is interfaced to other trace analysis and process analysis tools, including process mining. The framework is currently being applied to the domain of stroke patient management.

2 Domain knowledge and framework description

With the help of an expert physician, medical domain knowledge has been formalized in an ontology (which has been organized by goals) by using the Protégè ontology editor [1]. Actions in traces can be mapped to the ground terms of the ontology, so, by navigating the ontology, it is possible to abstract actions by goals at different levels of detail.



Fig 1. (left): An excerpt from the stroke domain ontology **(right):** Forward chaining on the rules allows the determination of the correct ancestor for CAT

In order to identify which of the possibly multiple ancestors (i.e. goals) of an action in the ontology should be considered for abstracting the action itself, a rule base has been formalized. Contextual information (i.e., the actions that have already been executed on the patient concerned in the trace, and/or her/his specific clinical conditions) is used to activate the correct rules.

Figure 1 shows, as an example, the multiple ancestors of the CAT (computerized axial tomography) action. The CAT action has two ancestors (monitoring and timing). Forward chaining on the rules allows to determine the correct ancestor of CAT depending on the context. The ontology has been integrated with the SNOMED-CT [3] code, which is an attribute of CAT action. According to the architecture described in Figure 2, in the framework trace abstraction is provided as an input to

- Trace comparison and clustering
- Process mining

Trace comparison resorts to a metric [4] that has been properly extended to deal with abstracted traces as well. Process mining resorts to classical process model discovery algorithms embedded in the open-source framework ProM [2].

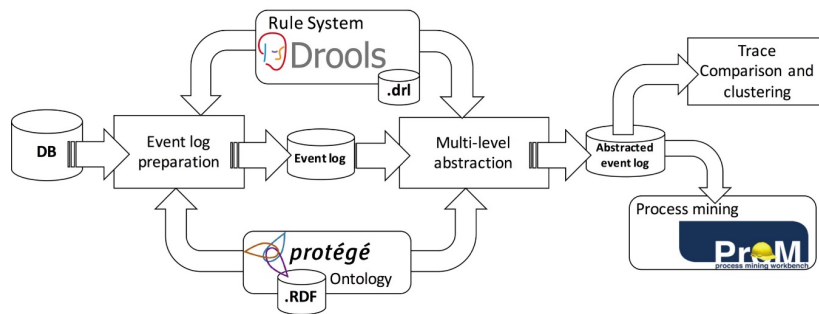


Fig 2. Framework architecture and data flow

The impact of the abstraction mechanism has been tested both on process mining and on trace comparison.

The available event log for the experiments is composed of more than 15000 traces, collected at the 40 Stroke Unit Network (SUN) collaborating centers of the Lombardia region, Italy. Traces are composed of 13 actions on average. The 40 Stroke Units (SUs) are not all equipped with the same human and instrumental resources: in particular, according to resource availability, they can be divided into 3 classes. Class-3 SUs are top class centers, able to deal with particularly complex stroke cases; class-1 SUs, on the contrary, are the more general centers, where only standard cases can be managed.

3 Experimental results

3.1 Process mining

First, a test was performed to see whether the capability to abstract the event log traces on the basis of their semantic goals allows process models to be obtained where unnecessary details are hidden, but key behaviors are clear. Indeed, if this hypothesis holds, in the stroke application domain it becomes easier to compare process models of different SUs, highlighting the presence/absence of common paths. This happens regardless of minor action changes (e.g., different ground actions that share the same goal) or irrelevant different action ordering or interleaving (e.g., sets of ground actions, all sharing a common goal, that could be executed in any order). Figure 3 compares the process models of two different SUs (SU-A and SU-B), mined by resorting to ProM's Heuristic Miner [2], operating on ground traces. Figure 4, on the other hand, compares the process models of the same SUs as Figure 3, again mined by resorting to Heuristic Miner, but operating on traces abstracted according to the ontology. Generally speaking, a visual inspection of the two graphs in Figure 3 is very difficult. Indeed, these two ground processes are “spaghetti-like” and the extremely large number of nodes and edges makes it hard to identify commonalities in the two models. The abstract models in Figure 4, on the other hand, are much more compact, and it is possible for a medical expert to analyze them. In particular, the two graphs in Figure 4 are not identical, but in both of them it is easy to identify the abstracted actions which correspond to the treatment of a typical stroke patient. However, the model for SU-A exhibits a more complex control flow (with the presence of loops), and shows three additional actions with respect to the model of SU-B. This finding can be explained, since SU-B is a class-1 SU, where SU-A is a class-2 SU, where different kinds of patients, including some atypical/more critical ones, can be managed, thanks to the availability of different skills and instrumental resources.

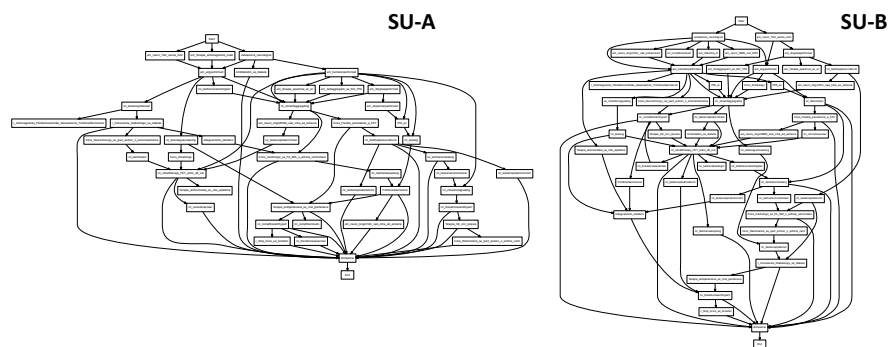


Fig 3. Comparison between two process models, mined by resorting to Heuristic Miner, Comparison between two process models, mined by resorting to Heuristic Miner, operating on ground traces. The Figure is not intended to be readable, but only to give an idea of how complex the models can be

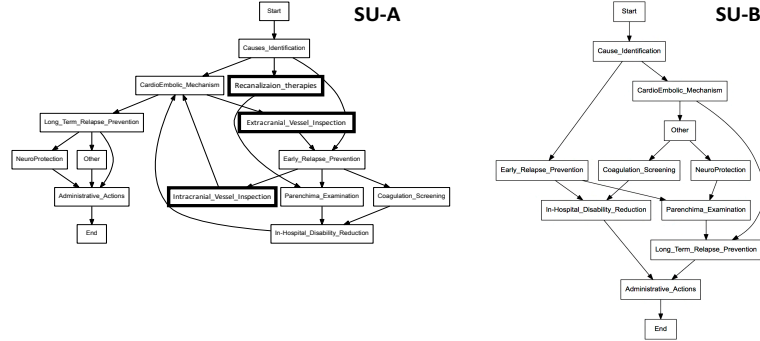


Fig 4. Comparison between the two process models of the same SUs as Figure 4, mined by resorting to Heuristic Miner, but operating on abstracted traces

3.2 Clustering

In a second experimental work the impact of the abstraction mechanism on trace comparison has been analyzed on the quality of trace clustering.

The technique that has been used is a hierarchical clustering technique, known as Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [5]. UPGMA operates in a bottom-up fashion.

In the experiments the hypothesis that needed to be tested was the following: the application of the abstraction mechanism allows one to obtain more homogeneous and compact clusters (i.e., able to aggregate closer examples - homogeneity is a widely used measure of the quality of the output of a clustering method). However, outliers (i.e., in our application domain, traces that could correspond to the treatment of atypical patients suffering from several inter-current complications such as diabetes, hypertension, ventricular arrhythmia, or venous thrombosis, who required many extra tests and many specialist counseling sessions) are still clearly identifiable, and isolated in the cluster hierarchy.

The average of cluster homogeneity values was computed level by level in the hierarchies. Figure 5 and 6 report, as an example, the results on a specific SU. The obtained cluster hierarchy height was 19 when working on ground traces, and 21 when working on abstracted ones. As can be observed in Figure 5, homogeneity on abstracted traces is always higher than the one calculated on ground traces.

As for the management of outliers, they are still clearly isolated when we work on abstracted traces and merged very late in the cluster hierarchy. For example, in Figure 6, trace 113 was merged only at level 0, both on ground and on abstracted traces.

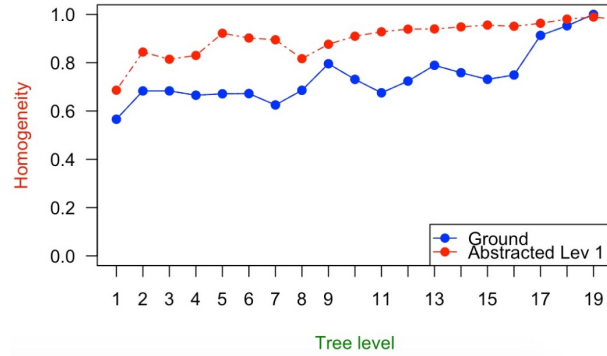


Fig 5. Comparison between average homogeneity values, computed level by level in the two cluster trees obtained by UPGMA on ground traces and on abstracted traces

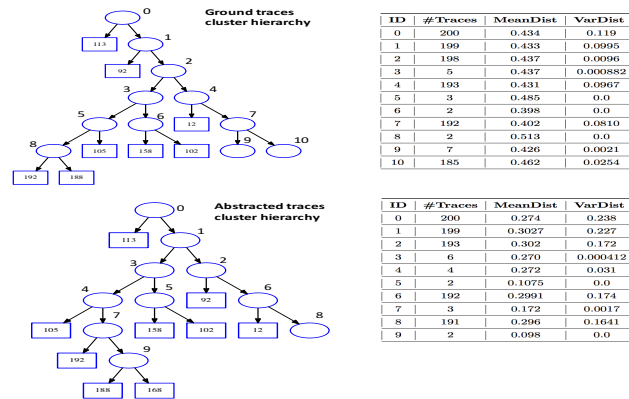


Fig 6. Identification of outliers (in rectangles) in cluster hierarchies (only the upper hierarchy levels are shown) and trace comparison statistics in clusters.

4 Future work

In the future it will be necessary to plan to extend the approach, in particular as regards the expressiveness of the rule base. Specifically, by introducing temporal constraints in rule antecedents, to better define the context of execution of an action, and thus to find its correct ancestor in the case of multiple inheritance. For instance, the execution of a specific action before the action to be abstracted could trigger a rule only if it took place within 24 hours. Moreover, rules could be made “fuzzy”: referring to the previous example, a time delay of 25 hours may be accepted as well.

Finally, as already observed, the actions exploited to implement medical goals (ground concepts in our ontology) are mapped to SNOMED [3] concepts. On the other hand, since not all the medical goals needed in our application are reported in SNOMED [3] (and the “has-intent” SNOMED [3] relation covers only partially the needs), it was not possible to map all terms at higher ontology levels to SNOMED [3] codes. Therefore, the goal/subgoal/goal-implementation

relations often had to be defined from scratch. Possible analyses for a tighter integration will be conducted in the future. Moreover, it will be necessary to investigate if the various attributes reported in SNOMED [3] (e.g., the type of substances administered to a patient, or the morphological part of the body involved) may be used to abstract actions along different dimensions (other than the goal), which could be of interest in this medical application. The already existing connection to the SNOMED [3] vocabulary will make this step relatively easy.

Moreover, it will be possible to extend the existing metric for process model comparison [6], in order to properly manage the information collected during the abstraction phase. Last but not least, further experiments will be conducted.

References

1. The Protege Team. The Protege website: [urlhttp://protege.stanford.edu/](http://protege.stanford.edu/), 2013.
2. A. K. Alves de Medeiros, W. M. P. van der Aalst, and C. Pedrinaci. Semantic process mining tools: Core building blocks. In W. Golden, T. Acton, K. Conboy, H. van der Heijden, and V. K. Tuunainen, editors, 16th European Conference on Information Systems, ECIS 2008, Galway, Ireland, 2008, pages 1953-1964, 2008.
3. SNOMED-CT website: <http://browser.ihtsdotools.org/>?
4. S. Montani, G. Leonardi, M. Striani, S. Quaglini, A. Cavallini, *Multi-level abstraction for trace comparison and process discovery*, Expert Systems with Applications 81 (2017) 398-409 - DOI:<http://doi.org/10.1016/j.eswa.2017.03.063>
5. Sokal, R. , & Michener, C. (1958). A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38 , 1409–1438 .
6. Stefania Montani, Giorgio Leonardi, Silvana Quaglini, Anna Cavallini, Giuseppe Micieli, *A knowledge-intensive approach to process similarity calculation*, Expert Systems with Applications, Volume 42, Issue 9, 2015, Pages 4207-4215, ISSN 0957-4174, <http://dx.doi.org/10.1016/j.eswa.2015.01.027>.