Declarative Process Mining: Reducing Discovered Models Complexity by Pre-Processing Event Logs

Pedro H. Piccoli Richetti, Fernanda Araujo Baião*, Flávia Maria Santoro*

Department of Applied Informatics - Federal University of the State of Rio de Janeiro, Rio de Janeiro, Brazil
{pedro.richetti, fernanda.baiao, flavia.santoro}@uniriotec.br

Abstract. The discovery of declarative process models by mining event logs aims to represent flexible or unstructured processes, making them visible to business and improving their manageability. Although promising, the declarative perspective may still produce models that are hard to understand, both due to their size and to the high number of restrictions of the process activities. This work presents an approach to reduce declarative model complexity by aggregating activities according to inclusion and hierarchy semantic relations. The approach was evaluated through a case study with an artificial event log and its results showed complexity reduction on the resulting hierarchical model.

Keywords: process mining · declarative modeling

1 Introduction

Process mining techniques allow knowledge extraction from events stored by information systems. They are also an important connection between data mining and business process management. The interest on this topic has grown due to the advancement on computers technology and processes management, so even more events can be registered and more details about business process are available, and to the need for improving and supporting business processes in a competitive and rapidly changing environment [11].

Despite its benefits, process mining has some disadvantages. One of them is that discovered models tend to be large and complex, especially on flexible scenarios where process execution involves multiple alternatives. Because traditional techniques used on discovery try to model every possible process behavior, they result in a spaghetti-like model with an information overload that reduces model comprehensibility. Traditional imperative models are appropriate to represent well-structured models, because they provide better support for analysis and execution direction. On the other side of the continuum are the unstructured processes, where flexibility is needed to drive changes or deviations on the activities flow. van der Aalst et al. [12]

* Fernanda Araujo Baião and Flávia Maria Santoro are partially funded by the CNPq brazilian research council, respectively under the projects 309069/2013-0 and 307377/2011-3.

show how a declarative approach enables a better balance between flexibility and support. However, declarative process mining techniques may produce models with a high quantity of constraints, which may be incomprehensible for humans, as showed by Bose et al. [2].

In this work, we address the problem of high complexity of declarative models generated by automatic process mining. Our proposed approach reduces the model complexity by automatically generating process hierarchies in pre-processing time, in which proposed subprocesses aggregate activities according to semantic relations.

The rest of this work is structured as follows. Section 2 presents theoretical background and related work about declarative process modeling and mining, and about complexity reduction through activities abstraction. Section 3 explains the method to abstract activities through semantic relations. Section 4 presents the first ideas towards the proposal for preprocessing and mining event logs applying activity abstraction. Section 5 describes the case study, and its results are discussed in Section 6. Section 7 concludes the paper and points to future work.

2 Background and Related Work

A declarative approach focuses on the logic that governs interactions between the actions of a process, describing what can be done, restricting only the undesired behavior [14]. An example of declarative modeling language is Declare [12], which is grounded on constraint templates modelled in linear temporal logic (LTL). A set of Declare constraints is presented in [8]. An implementation for declarative process mining is the DeclareMiner [8], available as a ProM[1] plugin.

Haisjackl et al. [4] showed that the combination of constraints in a process model might generate new hidden dependencies, which are complex and difficult to be identified by humans. Reijers et al. [9] said that the increasing number of restrictions negatively impacts on the model quality.

Abstraction is seen as an effective approach to represent readable models, showing aggregated activities and hiding irrelevant details [10]. While on imperative models every process fragment ranging from a single entry and a single exit (SESE) can be grouped as a subprocess [13], on declarative models this structure is not informative enough, because the activities' sequence is not rigid. Hierarchies may be used to perform aggregation, thus reducing the mental effort to understand a model [14].

Zugal et al. [14] examined the effects of hierarchy on declarative models. As a result, they confirmed that structural grouping of activities is inadequate and, for declarative models, it should consider a common objective of the grouped activities. The transformation of hierarchical structures back to flat models is not always possible without changing the process structure and, possibly, its semantics. This possible loss can be compensated by the expressiveness enhancement [14].

Li et al. [7] proposed an approach to search for sequential patterns on event logs and replace them with abstract activities. For declarative models, sequential patterns

[1] The tool is available at http://www.processmining.org.

identification is not enough to infer groups of activities. Baier et al. [1] presented a method to construct abstraction layers in process models by matching events and activities. Their clustering schema is based on timestamps to calculate minimal distances. On a declarative perspective, this approach is not very adequate because there are constraints that cannot be identified by looking for minimal temporal distances.

Bose et al. [16] demonstrated how to discover hierarchical process models based on pattern abstractions by preprocessing an event log and applying Fuzzy Miner to discover maps that represent process models with abstractions. They defined a taxonomy for abstractions that considers loops and conserved regions relative to sequences in event log traces, but no semantic concerns are considered to build hierarchies.

None of the above-mentioned approaches addresses abstraction techniques on declarative process models to reduce their complexity. Thus, the contribution of this paper is showing how to automatically generate subprocesses by looking for semantic relations from activities labels of an unstructured business process. The generated subprocesses are incorporated into the event log prior to the process mining phase. The expected result is to produce a less complex declarative model.

3 A Method to Abstract Activities Through Semantic Relations

Inspired by the semantic approach of Leopold et al. [6] to name imperative process models and fragments, our approach applies natural language processing to identify common objectives between activity labels, and then abstracts these activities into hierarchies. Wordnet[2] was chosen to search for the hypernyms and holonyms semantic relations between the words in activity labels; differently from [6], we aim to search for common objectives that can be used to gather activities in a subprocess.

Algorithm 1 groups activities that have actions and objects related to abstract common senses. We keep track of how strongly a word is semantically related to its abstract concept according to the Lin metric, since its results are similar to human judgment [3]. The next step is to define how to adequately group activities into a subprocess. Algorithm 2 proposes a strategy for grouping based on a graph representation. A prototype for executing Algorithms 1 and 2 was implemented in Java language. Auxiliary Python NLTK3.0[3] scripts were used for the part-of-speech tagging step. PERL WordNet:SenseRelate::WordToSet[4] scripts were used to get the most adequate sense from a list of words to be disambiguated in a given context and WordNet:Similarity[5] scripts provided the semantic similarity relatedness calculus.

---

[2] WordNet is available at http://wordnet.princeton.edu/.

[3] The toolkit is available at http://www.nltk.org/.

[4] Refer to http://search.cpan.org/~tpederse/WordNet-SenseRelate-WordToSet-0.04/.

[5] Refer to http://search.cpan.org/~tpederse/WordNet-Similarity-2.05/.

4 Preprocessing and Mining Event Logs with Activity Abstractions

Bose et al. [15] stated that "Spaghettiness" of process models can be reduced by first mining common constructs or functionalities, abstract them and then discovering process models on the abstracted log. Given a list of activity groups found by Algorithm 2, each group may be represented by a complex activity that substitutes all occurrences of its grouped activities in an event log. For example, given the following trace from a flat model: *{a,b,c,d,c,a,d,b}*. Suppose we identify a subprocess *e* grouping the activities *a* and *c*. Substituting the activities by their subprocess, the modified trace will be: *{e,b,e,d,e,e,d,b}*. This preprocessed log can be used as input to existing declarative process mining algorithms.

After preprocessing, the declarative mining algorithm will be able to identify interactions only in the top-level process. To discover the constraints within a subprocess, the activities belonging to it should be filtered from the original event log and presented to the declarative mining algorithm. Removing all the other activities will imply in analyzing only the behavior of the subprocess activities.

Currently, our implemented approach is able to deal with only one layer of subprocesses, but this can be extended to handle deeper levels. However, the growth of level numbers may increase the fragmentation of the model and consequently increase the model complexity [14].

5 Case Study

The main objective of this case study was to observe if a declarative process model, discovered after replacing activities by subprocess directly on the event log, is less complex. The declarative process model "How to prepare oneself and materials for teaching pupils" was chosen from literature [4]. It has a flat and a hierarchical version, both manually designed.

The process was modeled and simulated in CPNTools[6], generating 5,000 traces. Using this event log, a list of unique activity labels of the process was used as input for Algorithm 1. After executing the first algorithm, a set of activity pairs with their respective average semantic similarity value was produced. Together with the previous output set, a semantic similarity threshold was defined to run Algorithm 2. This threshold is used to filter out pairs with low similarity values. In a user guided fashion, a 0.40 threshold value was chosen. The remaining pairs of activities are candidates to generate the subprocesses through Algorithm 2 execution.

Algorithm 2 provided two subprocesses as output: "*Prepare and give lessons*" containing the activities "Prepare lesson in detail", "Give lessons" and "Read about topic in more detail"; and "*Decide and prepare teaching*" containing: "Prepare teaching sequence" and "Decide on teaching method". The event log was modified by substituting every occurrence of an activity by its complex activity representing each sug-

[6] The tool is available at http://cpntools.org/.

gested subprocess. Then, the preprocessed event log was imported into ProM and the DeclareMiner plugin was used to discover a hierarchical declarative process model (Fig. 1b). The plugin parameters were set to "Min. Support" = 50 and "alpha" = 50, no additional filters were applied after the discovery. To compare the results, the unmodified event log was also mined to discover a flat process model (Fig. 1a). In order to mine each subprocess behavior, the original event log was preprocessed once more to extract only the subprocess activities. The preprocessing and mining steps should be carried out for each subprocess. All plugin settings were the same used for the hierarchical model. Table 1 summarizes the results for these mined process models.

6 Evaluation

To evaluate the results from both flat and hierarchical models, some metrics related to model complexity applicable to declarative models were calculated based on La Rosa et al. [5]. In addition, the number of constraints was used as a metric because it influences the complexity of declarative models, as stated in [9].

Algorithm 1: Identify semantic related activities

Input: List of unique activity labels A, number of levels to search in Wordnet's hypernymy and holonymy tree k
Output: Set of activity pairs with their respective average similarity measure R

Initialize R with $\varnothing$
foreach *activity label a in A* **do**
  Apply part-of-speech tagging to identify all verbs V and all nouns N in a
  foreach *verb v in V* **do**
    Identify all hypernyms for v until reach the kth level starting from v
  foreach *noun n in N* **do**
    Identify all hypernyms and holonyms for n until reach the kth level starting from n
Generate a set P_a with pairs of activities p_a(activity label *a1*, activity label *a2)* from the combination $\binom{A}{2}$
foreach *activity label pair* p_a *in* P_a **do**
  Generate a set $V_{1,2}$ with pairs of verbs $p_v(v_1,v_2)$ from the combination of each verb v_1 in V_1 from a_1 and each verb v_2 in V_2 from a_2
  foreach *pair* p_v *in* $V_{1,2}$ **do**
    Match all common hypernyms H_v between v_1 and v_2
    Invoke WordNet::SenseRelate::WordToSet algorithm to define the most adequate hypernymy h_v from H_v, using A as context
    Calculate Lin's semantic relatedness metric between v_1 and h_v and v_2 and h_v
  Generate a set $N_{1,2}$ with pairs of nouns $p_n(n_1,n_2)$ from the combination of each noun n_1 in N_1 from a_1 and each noun n_2 in N_2 from a_2
  foreach *pair* p_n *in* $N_{1,2}$ **do**
    Match all common hypernyms and holonyms H_n between n_1 and n_2
    Invoke WordNet::SenseRelate::WordToSet algorithm to define the most adequate hypernymy or holonymy h_n from H_n, using A as context
    Calculate Lin's semantic relatedness metric between n_1 and h_n and n_2 and h_n
  Calculate average semantic relatedness value s considering all nouns in N_1, N_2 and verbs in V_1, V_2 to their most adequate hypernymy or holonymy
  Add p_a and its s value to R
return R

Algorithm 2: Group semantic related activity labels

Input: List of unique activity labels *A*, Set of activity pairs with their respective average similarity measure *R*, semantic similarity threshold *t*
Output: Set of activity labels groups *S*

Initialize *S* with ∅
Remove all activity pairs from *R* with average similarity measure below *t*
Create a undirected weighted graph *G(V,E)* where each vertex *v* is an activity label from *A* and each edge *e* relates to a pair from *R* whose weight is the average similarity measure of the pair
while *G has edges* **do**
    Generate all possible vertex groups *P* where in a group each vertex relates to each other
    foreach *group p in P* **do**
        Sum the weight of all edges of *p*
    Identify the vertex group *h* with the highest weight sum
    Add *h* to *S*
    Remove all vertex in *h* from *G*
return *S*

Considering that the input event log was the same, the reduction on the total number of activities (8 in flat model to 5 in hierarchical model), together with the lower number of constraints on the second model, positively contribute for reducing the overall complexity and make it easier to understand the process with abstractions. When looking at the subprocesses, the fewer number of activities tends to make them easier to understand when compared to the full flat model. Even merging the metrics for the hierarchical model and its subprocesses, the constraint/activity ratio remained lower than in the flat model.

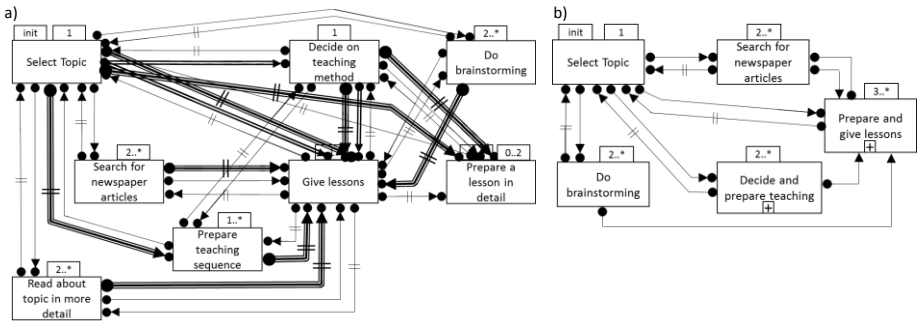

Fig. 1. Mined Declare models from the (a) flat and the (b) preprocessed event logs.

We are aware that natural language processing may introduce some bias on identifying the grammatical types of words. Not always an activity label is written as a complete sentence, which may reduce the POS tagging accuracy. The predefined search level in the hypernymy and holonymy tree results in not considering common concepts that are beyond this limit. However, choosing a broader limit may bring uninteresting or too vague common synsets that will not help to increase semantic relatedness.

The resulting mined model could be compared to the a priori theoretical model presented in [4]. The hierarchical a priori model has only one subprocess, called "*Prepare lessons*", with three activities. Our automatic proposal discovered the subprocess "*Decide and prepare teaching*", that contains two common activities with the

manually identified "*Prepare lessons*" subprocess ("Prepare teaching sequence" and "Decide on teaching method"). The "*Prepare and give lessons*" subprocess, which did not exist in the theoretical model, was found in our approach due to the affinity between its activities names ("Prepare lesson in detail", "Read about topic in more detail" and "Give lessons"). On manual modeling, other reasons besides the semantics can lead to activity aggregation, such as the execution sequence, or a deliberate decision based on personal judgment of the process modeler. The proposed automated method was able to produce less complex and easier to understand models.

Table 1. Complexity related metrics from the discovered process models.

	Flat	Hierarchical only	Subprocess "Decide and Prepare Teaching"	Subprocess "Prepare and Give Lessons"	Hierarchical + subprocesses
No. of Activities	8	5	2	3	10
No. of Constraints	45	18	5	9	32
No. of Different Constraints	9	8	5	8	10
No. of Subprocessess	0	2	0	0	2
Contraint/Activity Ratio	5.63	3.60	2.50	3.00	3.20

7 Conclusion and Future Work

Although there may be some semantic modifications relating to model constraints when using hierarchy, it is expected that the complexity reduction benefits may compensate this loss of information. The case study firstly evaluated the proposed method and evidenced its feasibility and promising results when inferring relationships between activities by looking its semantics. Further experiments will be conducted on more process models of different domains with diverse labeling quality, as well as on real life event logs, to assess its success and limitations on other scenarios.

Further improvements will consider the evaluation of quality dimensions [11] on the resulting hierarchical model, because the simplification may diminish quality, e.g., reduce precision or fitness of a model. Complimentary semantic relations such as Least Common Subsumer are also being evaluated.

This work has the purpose to help domain non experts and beginner practitioners to better understand declarative process models by automatically suggesting subprocesses to make the models less complex and more legible. When there is no previous knowledge about the model to be discovered, the proposed method may show important views of a process model that can be comprehended and then revised or applied in process improvement.

References

1. Baier, T., Mendling, J.: Bridging abstraction layers in process mining by automated matching of events and activities. In: Daniel, F., Wang, J., Weber, B. (eds.) Business Process Management, LNCS, vol. 8094, pp. 17-32. Springer Berlin Heidelberg (2013)
2. Bose, R. P. J. C., Maggi, F. M., van der Aalst, W. M. P.: Enhancing Declare Maps Based on Event Correlations. In: Daniel, F., Wang, J., Weber, B. (eds.) Business Process Management, LNCS, vol. 8094, pp. 97-112. Springer Berlin Heidelberg (2013)

3. Lin, D.: An information-theoretic definition of similarity. In : ICML '98 Proceedings of the Fifteenth International Conference on Machine Learning, vol. 98, pp. 296-304 (1998)
4. Haisjackl, C., Zugal, S., Soffer, P., Hadar, I., Reichert, M., Pinggera, J., Weber, B.: Making Sense of Declarative Process Models: Common Strategies and Typical Pitfalls. In: Nurcan, S., Proper, H., Soffer, P., Krogstie, J., Schmidt, R., Halpin, T., Bider, I. (eds.) Enterprise, Business-Process and Information Systems Modeling, LNBIP, vol. 147, pp. 2-17. Springer Berlin Heidelberg (2013)
5. La Rosa, M., Wohed, P., Mendling, J., ter Hofstede, A. H. M., Reijers, H. A., Van der Aalst, W. M. P.: Managing Process Model Complexity Via Abstract Syntax Modifications. Industrial Informatics, IEEE Transactions 7, 614-629 (2011)
6. Leopold, H., Mendling, J., Reijers, H., Rosa, M.: Simplifying process model abstraction: Techniques for generating model names. Information Systems 39, 134-151 (2014)
7. Li, J., Bose, R. P. J. C., van der Aalst, W. M. P.: Mining Context-Dependent and Interactive Business Process Maps Using Execution Patterns. In: zur Muehlen, M., Su, J. (eds.) Business Process Management Workshops, LNBIP, vol. 66, pp. 109-121. Springer Berlin Heidelberg (2011)
8. Maggi, F. M., Mooij, A. J., van der Aalst, W. M. P.: User-Guided Discovery of Declarative Process Models. In: IEEE Symposium on Computational Intelligence and Data Mining. pp. 192-199. IEEE Computer Society (2011)
9. Reijers, H. A., Slaats, T., Stahl, C.: Declarative Modeling - An Academic Dream or the Future for BPM? In: Daniel, F., Wang, J., Weber, B. (eds.) Business Process Management, LNCS, vol. 8094, pp. 307-322. Springer Berlin Heidelberg (2013)
10. Smirnov, S., Reijers, H. A., Weske, M.: A Semantic Approach for Business Process Model Abstraction. In: Mouratidis, H., Rolland, C. (eds.) Advanced Information Systems Engineering, LNCS, vol. 6741, pp. 497-511. Springer Berlin Heidelberg (2011)
11. IEEE Task Force on Process Mining: Process Mining Manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops, LNBIP, vol. 99, pp. 169-194. Springer Berlin Heidelberg (2012)
12. van der Aalst, W. M. P., Pesic, M., Schonenberg, H.: Declarative workflows: Balancing between flexibility and support. Computer Science - Research and Development 23, 99-113 (2009)
13. Weber, B., Reichert, M., Mendling, J., Reijers, H. A.: Refactoring large process model repositories. Computers in Industry 62, 467-486 (2011)
14. Zugal, S., Soffer, P., Haisjackl, C., Pinggera, J., Reichert, M., Weber, B.: Investigating expressiveness and understandability of hierarchy in declarative business process models. Software & Systems Modeling, 1-23 (2013)
15. Bose, R. P. J. C., van der Aalst, Wil M. P.: Abstractions in Process Mining: A Taxonomy of Patterns. In: Dayal, U., Eder, J., Koehler, J., Reijers, H. (eds.) Business Process Management, LNCS, vol. 5701, pp. 159-175. Springer Berlin Heidelberg (2009)
16. Bose, R. P. J. C., Verbeek, E. H. M. W., van der Aalst, W. M. P.: Discovering Hierarchical Process Models Using ProM. In: Nurcan, S. (eds.) IS Olympics: Information Systems in a Diverse World, LNBIP, vol. 107, pp. 33-48. Springer Berlin Heidelberg (2012)