

Data Science Methods for Declarative Process Mining

Nico Grohmann,¹ Gottfried Vossen²

Abstract: Process mining has recently drawn the attention of a large audience from both research and industry for the analysis of business processes based on execution data. Also, declarative approaches for process modeling and mining which are especially suitable for complex and loosely-structured processes have become of interest in research in the last years. Several tools and approaches allow the generation of declarative process models (most commonly in form of the DECLARE language) from event data. This research proposal focuses on the discovery step for declarative processes and suggests to supplement or enrich it with commonly known techniques from data science. The assumption is that data science methods like association rule mining or sequential pattern mining can also discover certain types of declarative constraints and additionally generate activation, time, and correlation conditions. These conditions can support the practical understanding of declarative process models and deliver additional insights into the analyzed processes.

Keywords: Digitalization; Declarative Process Mining; Data Science Methods

1 Introduction

Business Process Management (BPM) has a long tradition as the field of research that deals with all aspects of processes in organizations. It focuses on the understanding and modeling of existing processes and tries to identify deviations from desired process executions. With current technological advances on business and information systems, it is now possible to not only model processes by hand, but use the traces of process executions in an IT system to discover the real-world processes. Approaches like this are subsumed under the term *process mining* and are a relatively new discipline of research. Process mining connects the disciplines of process (BPM) and data science by exploiting data sources for an analysis of processes. The Process Mining Manifesto [VdAAdM12] has outlined the foundations for this technology and set standards in terminology and research foci, and was continued with the work of van der Aalst [Va16]. Process Mining and BPM, in general, is very much focused on the imperative modeling paradigm that creates process models as diagrams in the form of, e.g., Petri nets or the Business Process Model and Notation (BPMN).

However, researchers have investigated a different style of process modeling, namely the *declarative* modeling paradigm. With this paradigm, process models are no longer depicted

¹ University of Muenster, Department of Information Systems, Leonardo-Campus 3, 48149 Muenster, Germany
nico.grohmann@wi.uni-muenster.de

² University of Muenster, Department of Information Systems, Leonardo-Campus 3, 48149 Muenster, Germany
gottfried.vossen@wi.uni-muenster.de

as graphs but rather constraints and conditions that restrict the model at hand. Any behavior that is not captured by these constraints is allowed, whereas imperative modeling explicitly states the flows allowed. Researchers have also started to use the declarative style for process mining, e.g., Alman et al. [Al20] developed a tool that covers several aspects of declarative process mining like discovery, conformance checking and a multi-perspective extension that allows the introduction of additional data constraints. The declarative modeling paradigm is especially useful for complex, loosely-structured processes that would result in a “spaghetti” diagram when modeling all possible flows. As process mining bridges the gap between process and data science, it seems natural to use data science techniques to support process mining, here specifically the process discovery for the declarative paradigm. Samples for such techniques include rule mining or sequential pattern mining. In this paper, we investigate how data science techniques can support the process discovery step for declarative process models. Besides the application of data mining techniques on event data we also suggest to evaluate existing declarative process mining algorithms and to hint about how these algorithms and data science techniques could complement each other for declarative process discovery.

The remainder of this paper is organized as follows: Section 2 introduces the fundamental terms and concepts related to the research field and focus of this paper. Section 3 encompasses the actual research proposal and planned contribution. Furthermore, it shows the current status and preliminary results of our research. Finally, Section 4 concludes the paper and offers future research opportunities.

2 Background

Process mining exploits event data generated by the execution of real business processes for process analysis. This data is left as traces like logs, database entries or message interchanges in information systems that support a process. To use it for process mining, the event data has to be transformed and integrated into an event log that can be imported into process mining tools. There is a variety of commercial tool vendors like Celonis or Signavio or applications originating from research like Disco or ProM, which offer process mining functionality to different extents. The most prominent one is process discovery. Process discovery denotes the type of process mining that generates process models from event data. Thereby, performance indicators for all real executions can be created. Conformance checking denotes the activity of checking whether a given (possibly hand-made) process model matches its actual execution and, if not, where the deviations lie. Process discovery is also possible when using the declarative modeling paradigm.

One prominent notation form for declarative process models is DECLARE [PSVdA07]. DECLARE includes a set of constraint templates for describing and restricting a certain process. Maggi et al. [MMVdA11] present an approach to automatically discover DECLARE models from event data. Their algorithm takes a set of candidate constraints and then checks whether they are satisfied in a given event log. Tools like RuM [Al20] can generate declarative

process models from event log input and express them using a set of DECLARE constraints. Figure 1 shows a declarative process model for a sepsis treatment process in a hospital.

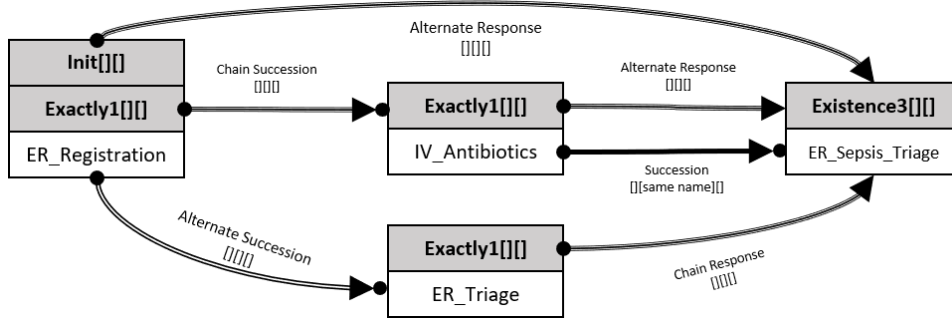


Fig. 1: Declarative Model of a Sepsis Treatment Process [A120].

It shows four instances of unary and six instances of binary DECLARE constraints. Unary constraints are applied to a single activity of the process. Binary constraints define the relationships between the activities. A user can specify attributes for each of the activities. Besides the DECLARE constraints, RuM allows users to specify activation, correlation, and time conditions. Activation conditions define conditions that have to hold to make the constraint active. Correlation conditions define the relationship between the attributes, e.g., two activities have to have the same name assigned; they are only possible on binary constraints. The same holds for the time conditions that can specify in which time window the second activity is executed after the first, e.g., between 2 and 5 hours after an activity.

Our goal is to include data science methods in the discovery process. The first two methods we have used are association rule mining and sequential pattern mining. Association rule mining generates if/then-statements that uncover existence relationships between items. Sequential pattern mining additionally takes the sequential order of events into account and generates rules that allow the deduction of temporal orders. We select these methods because we think that the set (or a subset) of DECLARE constraints is easily transferable to outputs of rule learning algorithms and vice versa. Additionally, rules and sequential patterns as an alternative representation of DECLARE constraints can contribute to the understanding of declarative process models. The following section introduces the concrete research question and outlines the contribution of our research.

3 Research Proposal

We claim that it is possible to generate certain DECLARE constraints and conditions from event logs using association rule mining and sequential pattern mining if the event log is in the usual format for process mining including a case id, an activity name and timestamp information for each line. After transforming the event data in a way that every activity belonging to one case is in one line of the data set, frequent itemset mining calculates

itemsets of activities that frequently occur in instances of the process. Then, they serve as input for the creation of the association rules. The result is a set of association rules that have one or more activities as the premises and one or more activities as the conclusion. All in all, the output of frequent itemset, association rule and sequential pattern mining is combined to generate DECLARE models as well relations and conditions on additional attributes of an event log corresponding to the multi-perspective extension of DECLARE.

Considering the process depicted in Figure 1, for instance, there could be an association rule that has *ER_Registration* as the premise and *IV_Antibiotics* as the conclusion with a support (proportion of transactions that contain an itemset) of 1 and confidence (conditional probability of B given A) of 0.5. We demonstrate our deliberations on an imaginary sample scenario, the process model does not indicate the used values. One could derive from that rule that *ER_Registration* and *IV_Antibiotics* are part of the process in a hundred percent of the cases. This could result in the introduction of an **EXISTENCE**-constraint in the DECLARE notation for both activities.

Furthermore, we can introduce a **COEXISTENCE**-constraint for the activities *ER_Registration* and *IV_Antibiotics*. One point worthy to note, however, is that we cannot conclude something about the temporal orders of the activities. To this end, we use sequential pattern mining, more concretely the Generalized Sequential Pattern (GSP) algorithm [SA96]. The GSP algorithm is capable of generating sequence patterns that take the timestamp information of an activity into account which is highly relevant for considerations with processes. Applying the GSP algorithm on an event log results in a set of sequential patterns whereby each of them consists of a transaction which in this case is an activity of the process. If we assume that the GSP algorithm discovers a sequential pattern for *IV_Antibiotics* and *ER_Sepsis_Triage* with a support of 1 we can conclude that whenever the former activity is executed the latter one eventually follows. Thus, we can introduce a DECLARE **RESPONSE**-constraint for these two activities meaning that when the former activity is executed the later one has to follow eventually. The generation of **SUCCESSION** constraints is also possible. We see that in this way association rule and sequential pattern mining can perform an important intermediate step for declarative process mining from which the way to explicit declarative constraints in the DECLARE notation is not very far.

Sequential pattern mining is capable of considering additional attributes of the event log, e.g., date and time, cost or resource information. When including them into the GSP algorithm, it can discover patterns for activity and attribute sequences. For instance, a sequential pattern of *IV_Antibiotics/name=Max* and *name=Max* could be interpreted as a correlation condition **same name** for the **SUCCESSION** constraint for *IV_Antibiotics* and *ER_Sepsis_Triage* given that this pattern also appears for other names. Similar deliberations are possible for timestamp information when trying to construct the time conditions in the multi-perspective DECLARE approach. Without the use of MP-Declare mining these conditions have to be added manually after discovering the process model. Further investigations, also regarding activation conditions, have to follow here. One idea is that an implementation of our approach can generate recommendations for activation conditions. For instance, a pattern of

ER_Registration/name=Sara with a support of 1 could lead to the proposal to introduce **name=Sara** as an activation condition for the **INIT** constraint of *ER_Registration*. Human users can then manually decide whether this should be a valid activation condition or the pattern found is just a statistical accumulation that does not represent any condition on the process.

A sequential pattern of *IV_Antibiotics/name=Max* and *ER_Triage* could also be expressed in form of an *Event-Condition-Action* (ECA) rule. ECA rules originate from active databases where they are used, for example, to describe triggers. When an event happens (E), a condition is checked (C). A true condition then triggers the execution of an action (A). For the example depicted in Figure 1, the event could be the execution of activity *IV_Antibiotics*, the condition the check whether attribute name has value 'Max' and the action the execution of activity *ER_Triage* afterwards. ECA rules could in this way represent an alternative and additional notation for declarative constraints on a process that is possibly easier to understand for practitioners or process owners that are not familiar with the DECLARE notation or (declarative) process models in general. One can choose different support and confidence settings for frequent itemsets, association rule and sequential pattern mining. Support and confidence measures are comparable to the constraint support parameters in the RuM tool. Higher values generate more precise process models but can lead to missing out on special cases and rarer process executions.

4 Conclusion and Outlook

Our research proposal focuses on the study and application of data science methods for declarative process mining and how they can play a role in discovering patterns leading to constraints on the process. We have shown how such patterns could look like using association rule and sequential pattern mining. Next, we have to further investigate how these discovered patterns can lead to constraints, especially to the constraints available in the DECLARE language for declarative process models. In this context, we require a comparative evaluation of declarative mining algorithms and our approach. Besides the discovery of declarative constraints, data science methods can also help to automatically discover activation, time, and correlation conditions for the multi-perspective DECLARE approach or ECA rules instead of manually adding them after the discovery process. A positive side effect of using data science methods for declarative process discovery is that it could improve transparency on the discovered constraints in comparison to the outcome of a traditional mining algorithm, especially for people with a non-technical or data science background. We will continue by applying the proposed methods on other event logs to further validate our ideas.

References

- [Al20] Alman, Anti; Di Ciccio, Claudio; Haas, Dominik; Maggi, Fabrizio Maria; Nolte, Alexander: Rule Mining with RuM. In (van Dongen, Boudewijn F.; Montali, Marco; Wynn, Moe Thandar, eds): ICPM. IEEE, 2020.
- [MMVdA11] Maggi, Fabrizio M; Mooij, Arjan J; Van der Aalst, Wil MP: User-guided discovery of declarative process models. In: 2011 IEEE symposium on computational intelligence and data mining (CIDM). IEEE, pp. 192–199, 2011.
- [PSVdA07] Pestic, Maja; Schonenberg, Helen; Van der Aalst, Wil MP: Declare: Full support for loosely-structured processes. In: 11th IEEE international enterprise distributed object computing conference (EDOC 2007). IEEE, pp. 287–287, 2007.
- [SA96] Srikant, Ramakrishnan; Agrawal, Rakesh: Mining sequential patterns: Generalizations and performance improvements. In: International conference on extending database technology. Springer, pp. 1–17, 1996.
- [Va16] Van der Aalst, Wil: Process Mining - Data Science in Action. Springer-Verlag Berlin Heidelberg, 2016.
- [VdAAdM12] Van der Aalst, Wil; Adriansyah, Arya; de Medeiros, Ana Karla Alves et al.: Process Mining Manifesto. In: Business Process Management Workshops, volume 99, pp. 169–194. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.