Probabilistic Declarative Process Mining

Michela Vespa[1]

[1]Department of Engineering, University of Ferrara, Italy

Abstract

This doctoral research integrates Probabilistic Logic Programming (PLP) with Declarative Process Mining (DPM) to address uncertainty in business process management. The research will have to address assumptions that are traditionally made in binary DPM but do not always hold in real cases, or are ignored in real world logs: (i) a process model able to perfectly distinguish between positive and negative examples; (ii) noise in the log, at the event level, or at the trace level; (iii) log incompleteness: for a variety of reasons some events could have not been recorded in the log ("missing events"); (iv) clarity and understandability of a declarative model, which might be undermined by the large number of its constraints.

1. Introduction and Problem Description

The management of business processes is of extreme importance for supporting efficiency improvements in organisations. Process Mining (PM) deals with the discovery and representation of process models from event data collected from organisations about the executions of their processes. The overall doctoral research is performed in the context of the PRIN2022 project "PRObabilistic DEclarative process mining (PRODE)"[1]. The project addresses the task of learning and reasoning upon *declarative* process models, within the setting of binary supervised learning, taking into account also uncertainty. With the aim of providing viable solutions, the research project will focus on the following issues in particular:

1. **Exploit the availability of positive and negative examples**: in many cases, user experts provide examples with desired and undesired behaviour (hence the labels "positive" and "negative"), but the majority of the discovery approaches exploits only the positive set;
2. **Precision and understandability of discovered models**: precise models could perfectly discriminate between positive and negative examples, but might turn out to be too complex for the final user. We might want to learn models which do not perfectly discriminate between positive and negative examples, but which are simpler and understandable for the final user: probabilistic approaches might help to simplify the models, yet providing a clear and formal semantics;
3. **Deal with the uncertainty that real logs usually bear**: on one side, logs are just a partial incomplete view of the reality; on the other side, the information in the log might be incomplete, partially specified, and even non trustable. While many approaches provide a crisp yes/no answer to the question if a trace "is conformant" with a model, we will explore the possibility of returning a score representing the probability/degree of a trace to be compliant to the model;
4. **Preferable Models**: as there can be multiple output models from the process discovery task, with an associated uncertainty as a result of point 3), we might want to identify the preferable models for the user.

The doctoral research will take advantage of the development of works in the fields of Probabilistic Logic Programming (PLP) and Answer Set Programming (ASP), and will build a set of techniques that

Joint Proceedings of the Workshops and Doctoral Consortium of the 41st International Conference on Logic Programming, September 9–13, 2025, Rende, Italy

michela.vespa@unife.it (M. Vespa)

[1]https://prode.unife.it/

target the issues above by means of new combinations of declarative Process Mining with probabilistic and combinatorial approaches. The final aim is to produce more verifiable and understandable explanations of its processes to an organisation.

2. Background and overview of the existing literature

The Business Process Mining (BPM) community sees the integration of probability into various aspects of process modeling as a growing research topic, reflecting an interest in enhancing process analysis under conditions of uncertainty in real-world domains. Uncertainty can be found at various levels, such as process constraints, events, event attributes, or traces. For instance, in declarative PM, [1] introduced the notion of probabilistic process constraints by associating probabilities to DECLARE constraints with a frequentist approach. Also, they propose how to discover such constraints with traditional process mining algorithms, and explain how to carry out monitoring and conformance checking. In procedural PM, probabilities are handled at the level of traces or events in a trace [2]: they consider the analysis of a specific class of event logs, those containing uncertain events, i.e. recordings of executions of specific activities in a process which are enclosed with an indication of uncertainty in the event attributes. Uncertainty in traces is embedded in a process model called behavior net, a Petri net that can replay all and only the realizations of an uncertain trace. The conformance score is calculated by assessing upper and lower bounds on the conformance score for possible values of the attributes in an uncertain trace. In [3], traces are generated by a stochastic process model, which outputs a variety of possible sequences, each associated with a certain probability. Then, probabilistic trace alignment - the comparison between the model and the observed trace - is done by identifying the k model traces nearest to a particular observed trace.

Differently from previous work, we propose a novel semantics to handle uncertainty at the level of: i) events in a trace, ii) traces in a log, iii) and constraints in process models, in a declarative PM setting.

In Process Mining a *trace* or process instance represents a distinct execution of a process, potentially repeated multiple times. A trace typically consists of a sequence of activity executions, each identified by a distinct name and associated with temporal information that defines their order.

Definition 1 (Trace t and Log $\mathscr{L}$)**.** *Given a finite set $\mathscr{A}$ of symbols (i.e., activity names), a* trace t *is a finite, ordered sequence of symbols over $\mathscr{A}$, i.e. $t \in \mathscr{A}^*$, where $\mathscr{A}^*$ is the infinite set of all the possible finite sentences over $\mathscr{A}$. A* log $\mathscr{L}$ *is a finite set of traces, such as*

$$\mathscr{L} = \{t_1 = \langle a, b, c\rangle, \quad t_2 = \langle a, b, a, d\rangle, \quad t_3 = \langle a, a, d\rangle, \quad t_4 = \langle a, b, c\rangle\}$$

Declarative PM represents a branch of PM that prioritizes flexibility rather than strictly defined procedural workflows. Unlike procedural approaches that prescribe exact execution sequences, declarative PM defines *constraints* that specify conditions that must or must not be satisfied during process execution. Our work builds upon DECLARE [4], the most prominent declarative modeling formalism, which offers a collection of (graphical) constraint templates with formal semantics grounded in LTL_f logic [5]. Some representative constraints include "*activity a or b eventually occur in the process instance*" (referred to as *co-existence(a,b)*), and "*activity* a *must be executed at least once during the process*" (referred to as *existence(a)*). The semantics is based on the principle that each DECLARE template can be translated into a logical formula φ. A trace t is considered *compliant* with a constraint if it *logically entails* the corresponding formula φ.

Definition 2 (Compliance of a trace with a constraint)**.** *A trace t is compliant with a DECLARE constraint if it satisfies the corresponding logical formula φ, denoted as $t \models \varphi$. Conversely, if $t \not\models \varphi$, we say t violates the constraint.*

In the following, we extend this notion with respect to a set of constraints, i.e. a process model, that we formally call Declarative Process Specification.

Definition 3 (Declarative Process Specification)**.** *A Declarative Process Specification is a triple* $DS = (T, \mathcal{A}, C)$*, where* T *is a finite set of constraint templates* $c(x_1, \ldots, x_m)$ *(with arity* $m \in \mathbb{N}$*),* $\mathcal{A}$ *is a finite set of activity names, and* C *is a finite set of instantiated constraints* $c(a_1, \ldots, a_m)$ *with* $a_i \in \mathcal{A}$*.*

Definition 4 (Compliance of a trace versus a Declarative Process Specification)**.** *A trace is compliant with a DS if it entails the conjunction of the formulas* φ_i *corresponding to the* $c_i \in C$*:* $t \models \varphi_1 \wedge \ldots \wedge \varphi_n$ *where* n *is the cardinality of* C*.*

3. Goal of the research

The proposed goal of this research is to develop a probabilistic framework for declarative Process Mining that integrates uncertainty in both event data and model specifications. The framework aims to support both validation (via conformance checking) and model extraction (through learning algorithms) under uncertainty. The research goals of this work are the following: **RG1**) how uncertainty can be represented across events, traces, and constraints in declarative PM; **RG2**) defining compliance evaluation on both uncertain traces and probabilistic constraints; **RG3**) learning probabilistic declarative models from uncertain logs; **RG4**) model evaluation and selection based on user preferences. These goals are detailed in the Introduction.

4. Current status of the research

This section outlines our recent contributions, which are inspired by the Distribution Semantics [6] and enable the handling of uncertainty across multiple levels: events, traces, logs, and process constraints.

Definition 5 (Probabilistic Event [7])**.** *A* Probabilistic Event *is a couple Prob:EventDescription, where EventDescription is a symbol describing an event (EventDescription* $\in \mathcal{A}$*), while* Prob $\in [0, 1]$ *represents our degree of confidence in its occurrence. When* $p = 1$ *the event is* certain*.*

Definition 6 (Probabilistic Trace [7])**.** *A* Probabilistic Trace *is a trace where at least one event is probabilistic.*

Example 1. *The trace:* $t = \langle 0.9 : \textit{register\_order}, \textit{approve\_order}, \textit{schedule\_delivery}, \textit{invoice\_customer} \rangle$ *describes the situation where register_order was not logged, however it is very probable that it happened due to the standard process (the associated probability is high). register_order is a probabilistic event, and* t *is a probabilistic trace.*

Definition 7 (Probabilistic Log [8])**.** *A* probabilistic log $\mathcal{L}_p$ *is a log where at least one trace* t *is annotated with a probability* p*. A probability value of 1 means the trace certainly happened and the value will be omitted.*

Example 2. *The probabilistic log* $\mathcal{L}_p = \{t_1, 0.9 : t_2, t_3\}$ *describes the case in which the process instances* t_1 *and* t_3 *were observed and recorded, while* t_2 *was not observed but there is a high probability (0.9) that it happened.*

Definition 8 (Probabilistic Declarative Specification [9])**.** *A* Probabilistic Declarative Process Specification *PDS is a Declarative Process Specification DS where each constraint* $c_i \in DS$ *is a probabilistic constraint.*

Example 3. *The following PDS:*

$$C = \{ \quad 0.8 :: \textit{response(register\_order, approve\_order)} \quad (c_1) \quad \}$$

includes one probabilistic constraint c_1 *which indicates that the fact that register_order is potentially followed by approve_order carries relatively high importance in the business process.*

Constraints with probability $p_i = 1$ are termed *crisp* constraints. When all constraints are crisp, the specification is equivalent to a standard Declarative Specification. Following Distribution Semantics principles, a PDS defines a probability distribution over possible worlds where constraint c_i is included with probability p_i or excluded with probability $1 - p_i$. The probability of each resulting DS, denoted $P(DS)$, is the product of the probabilities p_i for selected, and $1 - p_i$ for excluded constraints. This distribution ensures $\sum_i P(DS_i) = 1$.

In [9] we introduced how to compute the compliance of a *certain* trace versus a PDS:

Definition 9 (Compliance of a trace versus a PDS [9])**.** *Given a PDS, the probability of compliance of a trace t w.r.t. a PDS is defined as:*

$$Comp(t, PDS) = \sum_{t \models DS_i} P(DS_i), \tag{1}$$

where DS_i represents each deterministic process specification induced by the selection of probabilistic constraints over the PDS, and $P(DS_i)$ is the probability associated with each deterministic specification.

Example 4. *Let us consider the following PDS:*

$$\begin{aligned} C = \{ \quad & 0.8 :: response(register\_order, approve\_order) && (c_1) \\ & 0.9 :: init(register\_order) && (c_2) \quad \} \end{aligned}$$

Such a PDS leads to 4 possible worlds according to the Distribution Semantics; so, 4 different regular DSs are possible, each one corresponding to a world, as shown in Table 1.

Consider the trace $t = \langle receive\_order\_request, register\_order, approve\_order \rangle$.

t is compliant with DS_2 and DS_4, i.e. those specifications that do not contain c_2, since the first event of the trace is not register_order. t's probability of compliance is $Comp(t, PDS) = P(DS_2) + P(DS_4) = 0.08 + 0.02 = 0.1$.

Declarative Process Specification (DS)	$P(DS_i)$
DS_1 = {response(register_order, approve_order), init(register_order)}	$P(DS_1) = 0.8 \times 0.9 = 0.72$
DS_2 = {response(register_order, approve_order)}	$P(DS_2) = 0.8 \times 0.1 = 0.08$
DS_3 = {init(register_order)}	$P(DS_3) = 0.2 \times 0.9 = 0.18$
DS_4 = { }	$P(DS_4) = 0.1 \times 0.2 = 0.02$

Table 1
Declarative Process Specifications generated by the PDS of Example 4.

In the following we present our current research on determining compliance of probabilistic traces versus a PDS. Following again the Distribution Semantics, each probabilistic event in a trace requires a "selection" determining its inclusion or exclusion in the resulting non-probabilistic traces $w_i(t)$. Similarly to PDS generating regular/non-probabilistic DSs, probabilistic events generate regular traces. Building on Definition 9, we assess compliance by examining each regular trace generated from a probabilistic trace against every Declarative Specification DS derived from the PDS.

Definition 10 (Compliance of a probabilistic trace versus a PDS)**.** *Given a a Probabilistic Declarative Process Specification PDS and a Probabilistic Trace t, let us consider all possible selections over the PDS and all possible selections over t. Let us consider all the Declarative Specifications DS_i generated from PDS and all regular traces $w_i(t)$ generated from t as a result of the selections.*

We define the compliance Comp(t,PDS) of a probabilistic trace t w.r.t. PDS as:

$$Comp(t, PDS) = \sum_{\substack{w_i(t) \\ DS_i}} \begin{cases} P(w_i(t)) \cdot P(DS_i) & \text{if } w_i(t) \text{ is compliant with } DS_i, \\ 0 & \text{otherwise.} \end{cases}$$

Unlike traditional conformance checking, which yields a binary outcome (either compliant or non-compliant), compliance in the probabilistic setting becomes a weighted evaluation over all possible DSs, where the weight is the probability of a regular trace compliant to the i_{th} DS.

5. Preliminary results

Up to now we used the implementation already available in [10], which allows us to perform compliance and computation of the probability of compliance of a certain or probabilistic trace versus a PDS, on a case study based on the ERAS® colorectal-surgery protocol [11]. We created a PDS of 21 constraints and a 21-event patient trace. In our latest experiments we simultaneously varied both the number of probabilistic constraints and probabilistic events (1 to 21), assigning user-defined probabilities to each, in order to compute $Comp(t, PDS)$ as in Def. 10. Experiments ran on a Linux machine with dual AMD® EPYC 9124 16-core CPUs, a 60 GB Prolog stack, with a 24-hour timeout, and results are shown in Table 2. As expected, enumerating all possible combinations of regular traces and regular Declarative

Experiment	# Probabilistic Constraints	# Probabilistic Events	Time (s)
1	1	1	793.96
2	2	2	1450.72
3	3	3	3503.28
4	4	4	7896.75
5	5	5	17751.73
6	6	6	44133.15
7	7	7	T.O.

Table 2
Execution time (in seconds) for computing the probability of compliance as the number of probabilistic constraints and events increases. T.O. indicates a timeout.

Process Specifications results in an exponential growth in execution time as the number of probabilistic constraints and events increases. We are currently exploring more efficient methods for computing compliance that bypass the need for full enumeration.

6. Open issues and expected achievements

This research aims to develop a comprehensive probabilistic framework for declarative Process Mining that integrates uncertainty in both event data and model specifications, and returns to the user preferable models. So far, it has provided a formal probabilistic semantics that captures how uncertainty can be represented across events, traces, and constraints (addressing **RG1**), along with a definition of probabilistic compliance that accounts for uncertainty in both traces and process models (addressing **RG2**). Preliminary implementations demonstrate the feasibility of our approach on realistic datasets, including a medical protocol case study, validating the practical relevance of the proposed semantics. However, open issues in **RG2** regard the development of scalable algorithms to avoid the exponential cost of enumerating all possible worlds to perform the task of compliance. To address **RG3**, we plan to design learning methods that extract probabilistic constraints from uncertain logs. We aim to leverage both positive and negative traces to improve learning accuracy and model quality. In line with **RG4**, we are exploring how to incorporate user preferences into model evaluation to enhance interpretability and trust.

7. Acknowledgments

Research funded by the Italian Ministerial grant PRIN 2022 “Probabilistic Declarative Process Mining (PRODE)”, n. 20224C9HXA - CUP F53D23004240006, funded by European Union – Next Generation EU. Research funded by the Italian Ministry of University and Research through PNRR - M4C2 - Investimento 1.3 (Decreto Direttoriale MUR n. 341 del 15/03/2022), Partenariato Esteso PE00000013 - ”FAIR - Future Artificial Intelligence Research” - Spoke 8 ”Pervasive AI” - CUP J33C22002830006, funded by the European Union under the NextGeneration EU programme”.

Declaration on Generative AI

The authors declare that in the planning, drafting, and/or revision of the work have made no use of generative AI tools.

References

[1] A. Alman, F. M. Maggi, M. Montali, R. Peñaloza, Probabilistic declarative process mining, Inf. Syst. 109 (2022) 102033. doi:10.1016/J.IS.2022.102033.

[2] M. Pegoraro, M. S. Uysal, W. M. van der Aalst, Conformance checking over uncertain event data, Information Systems 102 (2021) 101810. URL: https://www.sciencedirect.com/science/article/pii/S0306437921000582. doi:https://doi.org/10.1016/j.is.2021.101810.

[3] G. Bergami, F. M. Maggi, M. Montali, R. Peñaloza, Probabilistic trace alignment, in: 2021 3rd International Conference on Process Mining (ICPM), 2021, pp. 9–16. doi:10.1109/ICPM53251.2021.9576856.

[4] M. Pesic, H. Schonenberg, W. M. van der Aalst, Declare: Full support for loosely-structured processes, in: 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), 2007, pp. 287–287. doi:10.1109/EDOC.2007.14.

[5] G. D. Giacomo, M. Y. Vardi, Linear temporal logic and linear dynamic logic on finite traces, in: F. Rossi (Ed.), IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013, IJCAI/AAAI, 2013, pp. 854–860.

[6] T. Sato, A statistical learning method for logic programs with distribution semantics, in: L. Sterling (Ed.), Logic Programming, Proceedings of the Twelfth International Conference on Logic Programming, Tokyo, Japan, June 13-16, 1995, MIT Press, 1995, pp. 715–729.

[7] M. Vespa, E. Bellodi, F. Chesani, D. Loreti, P. Mello, E. Lamma, A. Ciampolini, M. Gavanelli, R. Zese, Probabilistic traces in declarative process mining, in: AIxIA 2024 Proceedings, volume 15450 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 330–345.

[8] M. Vespa, A probabilistic semantics for process mining, in: D. Bacciu, I. Donadello (Eds.), Proceedings of the AIxIA Doctoral Consortium 2024 co-located with the 23rd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2024), volume 3914, CEUR-WS, 2024, pp. 1–6.

[9] M. Vespa, E. Bellodi, F. Chesani, D. Loreti, P. Mello, E. Lamma, A. Ciampolini, Probabilistic compliance in declarative process mining, in: Proceedings of the 3rd International Workshop on Process Management in the AI Era (PMAI@ECAI 2024), volume 3779 of *CEUR Workshop Proceedings*, 2024, pp. 11–22. URL: https://ceur-ws.org/Vol-3779/paper1.pdf.

[10] E. Bellodi, M. Gavanelli, R. Zese, E. Lamma, F. Riguzzi, Nonground abductive logic programming with probabilistic integrity constraints, Theory and Practice of Logic Programming 21 (2021) 557–574. doi:10.1017/S1471068421000417.

[11] U. O. Gustafsson, et al., Guidelines for perioperative care in elective colorectal surgery: Enhanced recovery after surgery (eras®) society recommendations: 2018, World Journal of Surgery 43 (2019) 659–695. doi:10.1007/s00268-018-4844-y.