Learning execution contexts from event logs (Extended Abstract)*

Jing Yang[1,2,*], Chun Ouyang[1,2] and Arthur H. M. ter Hofstede[1,2]

[1] School of Information Systems, Queensland University of Technology, 2 George St, Brisbane City 4000, Australia
[2] Centre for Data Science, Queensland University of Technology, 2 George St, Brisbane City 4000, Australia

Abstract
Embedding artificial intelligence (AI) capability into business process management systems (BPMS) paves the way for these systems to realize complex process automation, taking into account different dynamics in process execution. Human resource behavior is one of the complex factors that contribute to those dynamics. Therefore, it is necessary to be able to have a precise characterization of the multidimensional process execution contexts in which human resources perform, and subsequently how execution contexts are linked with organizational grouping of those resources. In our work [1], we introduce the problem of learning such execution contexts from event logs and propose an approach based on decision tree learning, which utilizes discriminative event information embedded in an event log and domain knowledge about the process. The proposal of this problem and its solution contribute to the accurate understanding of variable human resource behavior in business process execution, and thus toward developing more adaptable and intelligent future BPMS.

Keywords
execution contexts, organizational model, event cube, event log, multidimensional process mining

1. Introduction

In an era of rapid development of artificial intelligence (AI), there is a growing need for business process management systems (BPMS) to achieve complex automation that considers dynamic contexts of process execution [2]. The variability of human resource behavior is a crucial contributing factor behind those dynamics, e.g., ad hoc decisions made by workers causing the usual execution path to diverge.

Process mining can be applied to discover employee-related insights from event logs recording the reality of process execution. Often, resources from the same organizational group (department, business role, etc.) share similar behavior. For example, resources with the same role are usually in charge of the same set of process activities, and those who take on the same time shifts tend to have similar active hours. Execution contexts [3] is a notion proposed to enable a precise characterization of those behavioral similarities of resources.

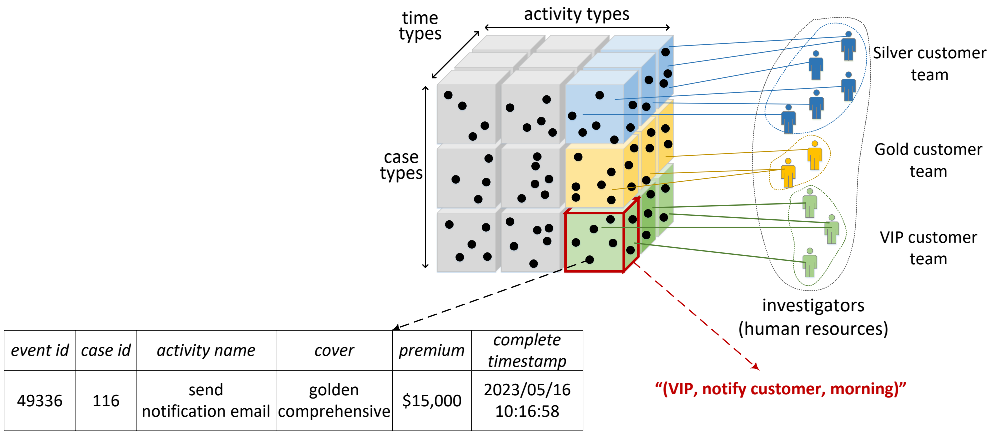
*Extended Abstract: This article is based on the original work [1] by the authors.

PMAI23@IJCAI: 2nd International Workshop on Process Management in the AI era, August 19, 2023, Macao, S.A.R

roy.j.yang@qut.edu.au (J. Yang); c.ouyang@qut.edu.au (C. Ouyang); a.terhofstede@qut.edu.au (A. H. M. ter Hofstede)
0000-0001-9218-6954 (J. Yang); 0000-0001-7098-5480 (C. Ouyang); 0000-0002-2730-0201 (A. H. M. ter Hofstede)

A set of execution contexts is defined by classifying and combining case types, activity types, and time types, which represent categories of process instances, activities, and periods, respectively. The classification of types is mapped onto event attributes in an event log, such that events can be partitioned according to the execution contexts.

Figure 1 illustrates the notion of execution contexts, in relation to events in a log and the organizational grouping of resources as event originators. Consider for example an insurance company where investigators are organized into different customer teams to handle insurance claims. "(VIP, notify customer, morning)" is one of the execution contexts, where "VIP" is a case type, "notify customer" is an activity type, and "morning" is a time type. These types are mapped onto event attributes indicating the type of insurance cover (a case attribute), the premium amount (another case attribute), the process activity, and the timestamp of activity completion. As a result, events like "(*cover* = 'golden comprehensive', *premium* = $15,000, *activity name* = 'send notification email', *complete timestamp* = 2023/05/16 10:16:58)" will correspond to this execution context.



event id	*case id*	*activity name*	*cover*	*premium*	*complete timestamp*
49336	116	send notification email	golden comprehensive	$15,000	2023/05/16 10:16:58

Figure 1: Events (illustrated as dots) in a log are partitioned by execution contexts (illustrated as cubes), and can then be linked to groups of resources as a natural consequence of the specialization of work (figure sourced from [1], with adaptation)

Modeling such contexts and their connection to organizational groupings is a prerequisite to accurate understanding of the routines and particularities of human resource behavior in process execution [3]. Given a process and its event log, execution contexts can be manually specified by process analysts [3], who usually possess the relevant domain information or have clear guiding questions. However, manual specification is not always an option, since the required domain knowledge cannot be assumed readily available or sufficiently concrete [4].

To tackle this challenge, we proposed an approach to *learning execution contexts from event logs*, which incorporates user-input domain knowledge and discriminative information embedded in the event data [1]. The approach is built on a customized decision tree learning algorithm and is capable of automatically extracting logic rules from an input log, which can then be used to define high-quality execution contexts.

2. Approach

We begin with the representation of execution contexts using categorization rules. A categorization rule (Def. 5 in [1]) is a Boolean formula in conjunctive normal form, consisting of one or more clauses. Each clause can evaluate an event by its value of some *type-defining attribute* (Def. 4 in [1]). Evaluating a set of categorization rules may dice a collection of events into sub-collections. Given an event log, a set of categorization rules may be used to define a set of types on the case, activity, or time dimension, respectively. When combined, three sets of such categorization rules define a set of execution contexts. For example, the execution context shown in Figure 1 may be defined by three categorization rules in conjunction:

- Case type "VIP" is defined as *cover* $=$ 'golden comprehensive' $\wedge$ *premium* $>= \$12,000$;
- Activity type "notify customer" is defined as *activity name* $\in \{$ 'send notification email', 'phone customer', 'send notification text message' $\}$; and
- Time type "morning" is defined as *complete timestamp.hour* $>= 8 \wedge$ *complete timestamp.hour* < 12.

Note that many various categorization rules may be constructed, hence resulting in many candidate sets of execution contexts for any given event log. We propose two measures that consider the quality of execution contexts based on how well they capture the specialization among resources. Impurity [1] measures the extent to which the same execution context contains events originated by different resources. Dispersal [1] measures the extent to which events originated by the same resource disperse across different execution contexts. A set of execution contexts of high quality should have low impurity and low dispersal. That is, (i) events in the same execution contexts should be originated by few resources, and (ii) events originated by the same resource should be partitioned into few execution contexts.

The problem of learning execution contexts from an event log is then formalized as the following. Given an event log, derive three sets of categorization rules that define case types, activity types, and time types, respectively, such that the resulting execution contexts have low impurity and low dispersal, i.e., high quality, with respect to the input log.

Our work [1] introduces the first approach to this problem. Given an event log, the approach begins with deriving an attribute specification to capture user domain knowledge about the events attributes in the log — it specifies which of the event attributes are relevant to defining case, activity, and time types; it also records categorization rules supplied by users to capture any existing categorization of attribute values. Then, guided by the attribute specification, a customized decision tree learning algorithm is used to iteratively induce categorization rules from the log, such that each selected split leads to a finer set of execution contexts with the lowest harmonic mean of dispersal and impurity.

We conducted experiments using a real-life event log (BPIC 2015 [5]), which demonstrates the feasibility of our approach and how execution contexts learned from an event log can be applied for resource-oriented analyses.

References

[1] J. Yang, C. Ouyang, A. H. M. ter Hofstede, W. M. P. van der Aalst, No Time to Dice: Learning Execution Contexts from Event Logs for Resource-Oriented Process Mining, in: International Conference on Business Process Management, Springer International Publishing, 2022, pp. 163–180. doi:10.1007/978-3-031-16103-2_13.

[2] M. Dumas, F. Fournier, L. Limonad, A. Marrella, M. Montali, J.-R. Rehse, R. Accorsi, D. Calvanese, G. De Giacomo, D. Fahland, A. Gal, M. La Rosa, H. Völzer, I. Weber, AI-augmented Business Process Management Systems: A Research Manifesto, ACM Trans. Manage. Inf. Syst. 14 (2023) 1–19. doi:10.1145/3576047.

[3] J. Yang, C. Ouyang, W. M. P. van der Aalst, A. H. M. ter Hofstede, Y. Yu, OrdinoR: A framework for discovering, evaluating, and analyzing organizational models using event logs, Decis. Support Syst. 158 (2022) 113771. doi:10.1016/j.dss.2022.113771.

[4] M. L. van Eck, X. Lu, S. J. J. Leemans, W. M. P. van der Aalst, PM^2: A Process Mining Project Methodology, in: Advanced Information Systems Engineering, Springer International Publishing, 2015, pp. 297–313. doi:10.1007/978-3-319-19069-3_19.

[5] B. van Dongen, BPI Challenge 2015 (2015). URL: https://data.4tu.nl/collections/BPI_Challenge_2015/5065424. doi:10.4121/uuid:31a308ef-c844-48da-948c-305d167a0ec1.