# Kairos: A Tool for Prescriptive Monitoring of Business Processes*

Kateryna Kubrak[1,*], Lana Botchorishvili[1], Fredrik Milani[1], Marlon Dumas[1], Alexander Nolte[1,2], Mahmoud Shoush[1] and Zhaosi Qu[1]

[1]University of Tartu, Estonia
[2]Carnegie Mellon University, Pittsburgh, PA, USA

**Abstract**

Kairos is a web-based tool that uses prescriptive process monitoring techniques to provide recommendations for ongoing cases based on execution data (event logs). Upon uploading an event log and specifying the recommendation parameters, Kairos provides the users with several recommendations for ongoing cases that are aimed at achieving positive case outcomes. The target users of Kairos are process analysts, operational workers, and tactical managers of business processes. In this paper, we describe the functionality of Kairos for process analysts.

**Keywords**

process mining, prescriptive process monitoring, user interface

## 1. Introduction

Prescriptive process monitoring is a family of techniques that recommend actions during the execution of a case that, if followed, increase the probability of positive case outcomes [1]. Examples of such actions could be the next activity to execute or a resource to allocate to a specific case. Prescriptive process monitoring techniques differ in their algorithms that produce recommendations [1]. For instance, some techniques are guiding (e.g., prescribe next best activity based on KNN-algorithm [2]), others are correlation-based (e.g., use predictive models [3]), or causality-based (e.g., use causal inference [4]).

In this paper, we present Kairos, a prescriptive monitoring tool for business processes that provides users with recommendations for ongoing cases. In [5], a wireframe concept of the tool was developed and evaluated with process mining experts. The findings of the evaluation suggested that three distinct user groups could benefit from the tool (process analysts, operational users, and tactical managers). Currently, Kairos supports functionality for process analysts. The process analysts can review completed and ongoing cases in the process, get an overview of prescribed recommendations, and explore the details of recommendations given for each case.

*ICPM Doctoral Consortium and Demo Track 2023, ICPM 2023, October 23-27, 2023, Rome, Italy*

*This research is supported by the Estonian Research Council (PRG1226) and the European Research Council (PIX Project).

*Corresponding author.

kateryna.kubrak@ut.ee (K. Kubrak)

0000-0002-4607-4000 (K. Kubrak); 0000-0002-1322-915X (F. Milani); 0000-0002-9247-7476 (M. Dumas); 0000-0003-1255-824X (A. Nolte); 0000-0002-7423-9909 (M. Shoush)



## 2. Architecture

Kairos is a web-based tool for prescriptive process monitoring. It consists of two main layers: **PrCore** (the engine that generates the prescriptions) and a **visualization layer** for the generated prescriptions (see Figure 1).

The backend logic (VL3) in the **visualization layer** coordinates the connection between the database (VL4), PrCore, and the frontend (VL1). The visualization layer receives data from PrCore using API endpoints (PC1). The data is processed by the backend logic (VL1), and recorded in the database (VL4), and sent to the frontend (VL1) when requested.

The visualization layer takes the output of PrCore as input for constructing visualizations. To support the construction of visualizations (which takes place in frontend), a backend is built as a middleware that connects to PrCore and a database to receive and store all the necessary data.

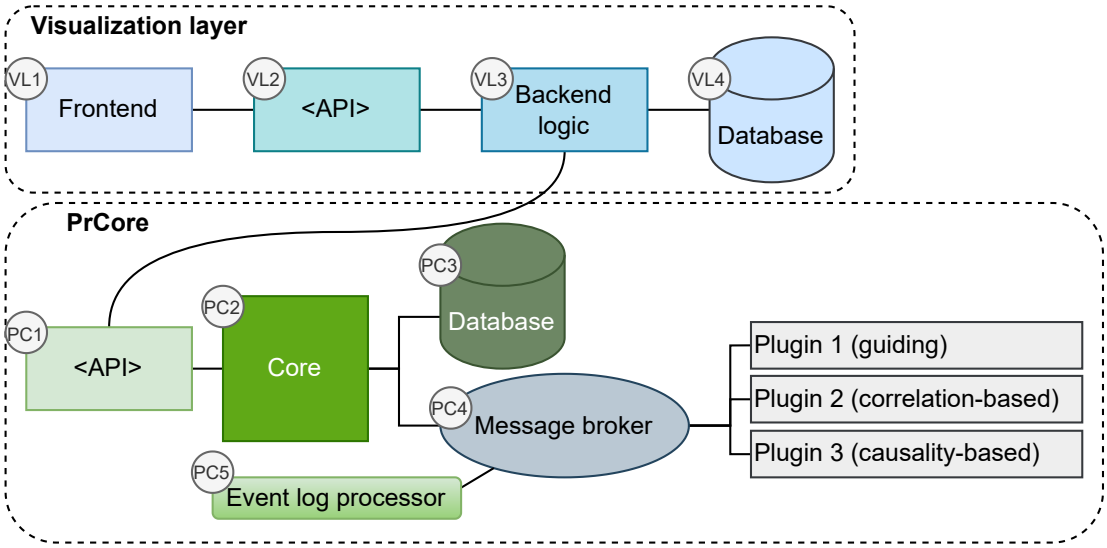


**Figure 1:** Overview of Kairos architecture.

The core application (PC2) coordinates **PrCore**. The core also utilizes an event log preprocessor (PC5) to preprocess the uploaded event log, and the message broker (PC4) to communicate with the three plugins that return recommendations based on different prescriptive techniques.

As such, plugin #1 provides guiding recommendations, i.e., it predicts the next best activity to execute based on a KNN algorithm [2]. Plugin #2 predicts an outcome of the ongoing case and triggers an alarm when the probability of a negative outcome hits the user-specified threshold value [3]. An alarm informs the user that an action is required. However, the specific action to take is not specified. Plugin #3 provides a causality-based recommendation. Specifically, the algorithm calculates the treatment effect of a user-specified action and prescribes it when the effect is positive (above zero) [4]. Each plugin necessitates data encoding to generate a fixed-size feature vector for training distinct predictive models. Consequently, PrCore is equipped with three established encoding methods, as emphasized by [6]: boolean encoding, frequency-based encoding, and simple index encoding. However, this feature is currently not user-configurable.

## 3. Functionality

Below, we elaborate on the main functionalities of Kairos. We describe event log upload, specifying parameters, overview of completed cases, overview of ongoing cases, and a detailed view of separate cases and recommendations.

- *Event Log Upload.* To start using Kairos, the user must upload an event log. The accepted file types are .csv, .xes, and .zip, with the file size limited to 100 Mb. The user can choose to either upload one event log (which is then split into train and test sets) or two event logs: one for training (that contains only completed cases), and the other for testing (that contains not completed cases). In the first option, the data is streamed, and in the second, it is delivered as a single response. More specifically, in the first option, we employ an 80%-20% temporal partitioning of the log. The initial 80% is used for training and validation of predictive models for all plugins, while the remaining 20% is reserved for streaming data. Conversely, in the second option, streaming data can be sourced from either temporal partitioning or user input. It is important to note that during the operational phase, we assume that new events from ongoing cases (streamed from the streaming data) continuously arrive, thus requiring recommendations from each plugin. With the streaming, we simulate how new cases would be constantly appearing in real-life scenarios. After the file upload, the user must specify data column types (e.g., case ID, text, numerical, etc.)
- *Parameters Definition.* Next, the user defines the parameters required for the plugins in PrCore. These parameters are: 1) The activity that marks the completion of a case. 2) A condition marking the positive outcome of a case. The outcome types are extracted from the event log by PrCore. The outcome types can be one of the columns in the event log (such as cost, specific activity, resource) or the duration of a case. For example, if a positive outcome is specified as equal to "duration less than or equal to 10 days", all cases lasting 10 or fewer days are marked to have a positive outcome, while the rest are marked as negative. 3) An intervention for plugin #3 [4], where the list of options is also provided by PrCore. For example, if the intervention is specified as "Activity equals [Contact_Customer]", then an algorithm estimates the causal effect of performing the activity [Contact_Customer] at a given point in time. 4) The threshold for the probability of a negative outcome, used to trigger an alarm for plugin #2 [3].
- *Completed Cases Overview.* Kairos provides users with an overview of completed cases. Here, three graphs are displayed. The first graph is a pie chart that shows the ratio of recommendations leading to positive or negative outcomes. The second one is a table indicating the number of positive outcomes for each recommendation type in the given process. The third graph displays the acceptance rate (by operational workers) of past recommendations, i.e., how many past recommendations were followed or rejected. Additionally, the user is presented with a table of completed cases.
- *Ongoing Cases Overview.* The aggregated view for ongoing cases contains two graphs: the first depicts the current distribution of the types of recommendations in all ongoing cases, and the other presents the current acceptance rate for the three types. Additionally, in the Cases Overview Table, the user can gain an overview of ongoing cases and select

the one(s) they wish to inspect in detail.

- *Individual Ongoing Case.* In this view (Figure 2), the user is presented with case-specific attributes and current case performance (w.r.t. to the target measure). The list of recommendations on the left displays information about available recommendations for the case in its current state. Each recommendation is marked with its type (guiding as "next activity", correlation-based as "alarm", and causality-based as "intervention"). The user can read about the different types by clicking on the info button above the recommendations. The recommendations of different types are shown when they are applicable. The recommendations are depicted on the process model to the right. The recommendation types have the same color as in the list on the left. The user can also see the past prescribed recommendations by toggling the "Show past recommendations" button. For these, it is also indicated if the recommendation was accepted or discarded by an operational worker. Last, the basic description of the machine learning models is accessible when clicking on Calculation Details.

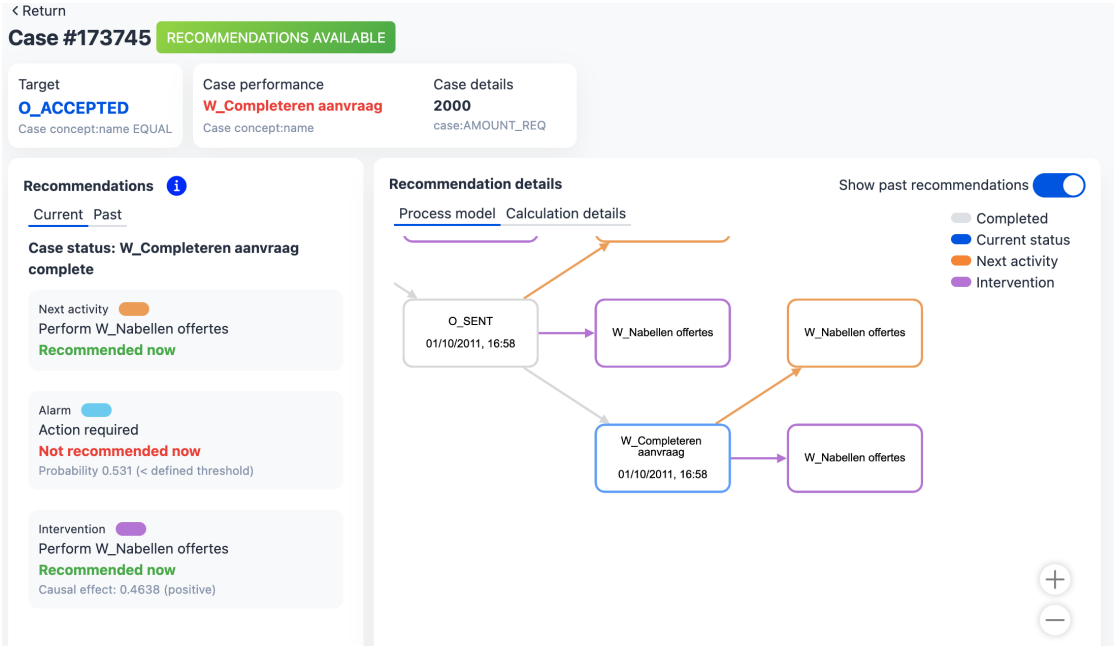


**Figure 2:** Individual case overview: recommendations list on the left, process model on the right.

## 4. Maturity & Availability

We evaluated the performance of PrCore engine on BPIC2012 and Road Traffic Fine Management Process event logs. For plugins #1 and #2, their accuracy scores serve as evaluation metrics (the number of accurate predictions made by a model relative to the total predictions). We report on the average prediction accuracy across the entire simulation for these plugins. Conversely, for

plugin #3, ground truth estimates for the causal effect are unavailable, making it impossible to estimate model accuracy. Instead, we report the average CATE throughout the entire simulation. Kairos utilizes a process pool for parallel processing (the dataset is partitioned based on available CPU cores, and a process pool is employed for concurrent processing). However, the log size influences both training and recommendation times. For instance, for BPIC2012 log, training takes circa 10 s, while recommendation circa 20 s. Conversely, for the larger BPIC2017 log, training and recommendation times extend to circa 35 s. On the user interface side, we evaluated the relevancy of the content included in the tool [5]. In future work, we plan to evaluate the usefulness and usability of the tool with potential end users and incorporate the suggestions that emerge from the evaluation in an improved version of the tool. In addition, we plan to implement and evaluate the functionality for operational workers and tactical managers [5].

| | Plugins | Accuracy | Recall | Precesion | F1-score | CATE |
|---|---|---|---|---|---|---|
| **bpic2012** | Plugin #1 | 0.72 | 0.72 | 0.77 | 0.68 | - |
| | Plugin #2 | 0.67 | 0.67 | 0.66 | 0.61 | - |
| | Plugin #3 | - | - | - | - | 0.41 |
| **traffic** | Plugin #1 | 0.57 | 0.57 | 0.96 | 0.54 | - |
| | Plugin #2 | 0.74 | 0.74 | 0.81 | 0.64 | - |
| | Plugin #3 | - | - | - | - | -0.3 |

Kairos can be accessed as a cloud service at https://kairos.cloud.ut.ee/. Examples of event logs and configurations to test with Kairos are available at https://github.com/AutomatedProcessImprovement/kairos-frontend/blob/master/README.md. In the same github space, documentation and source code of *PrCore*, and *frontend* and *backend* of the visualization layer are available. A demo video of Kairos is available at https://youtu.be/51zgZw40ZzA

## References


[1] K. Kubrak, F. Milani, A. Nolte, M. Dumas, Prescriptive process monitoring: *Quo vadis*?, PeerJ Comput. Sci. 8 (2022) e1097.

[2] M. Le, B. Gabrys, D. D. Nauck, A hybrid model for business process event and outcome prediction, Expert Syst. J. Knowl. Eng. 34 (2017).

[3] I. Teinemaa, N. Tax, M. de Leoni, M. Dumas, F. M. Maggi, Alarm-based prescriptive process monitoring, in: BPM (Forum), volume 329 of *Lecture Notes in Business Information Processing*, Springer, 2018, pp. 91–107.

[4] Z. D. Bozorgi, I. Teinemaa, M. Dumas, M. L. Rosa, A. Polyvyanyy, Prescriptive process monitoring based on causal effect estimation, Inf. Syst. 116 (2023) 102198.

[5] K. Kubrak, F. Milani, A. Nolte, M. Dumas, Design and evaluation of a user interface concept for prescriptive process monitoring, in: CAiSE, volume 13901 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 347–363.

[6] A. Leontjeva, R. Conforti, C. D. Francescomarino, M. Dumas, F. M. Maggi, Complex symbolic sequence encodings for predictive monitoring of business processes, in: BPM, volume 9253 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 297–313.