UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Software Engineering Curriculum

Lana Botchorishvili

Visualising Prescriptive Process Monitoring Output

Master's Thesis (30 ECTS)

Supervisors: Fredrik Payman Milani, PhD
Alexander Udo Nolte, PhD
Kateryna Kubrak, MA

Tartu 2023

Visualising Prescriptive Process Monitoring Output

Abstract:
The emergence of process mining in business process management has given way to developing algorithms that predict future processes based on historical event logs to improve operational efficiency. Further improvements have followed in the form of real-time prescriptive process monitoring tools, which identify volatile process instances and suggest interventions for upcoming activities to prevent risky execution. One of the main challenges of this practice is communicating interventions to end users in a clear and concise manner. This research was conducted to develop a real-time application that visualises available interventions and their impact on a process instance. The tool helps support process analysts in identifying activities that improve a KPI or circumvent a sub-optimal outcome in a process. A resulting artefact is a form of an operational dashboard - Kairos - that visualises prescriptive process monitoring output.

Keywords:
business processes, visualisation, prescriptive process monitoring

CERCS: T120, Systems engineering, computer technology

Preskriptiivse Protsessi Jälgimise Väljundi Visualiseerimine

Lühikokkuvõte:
Protsessikaevanduse tekkimine äriprotsesside haldamises on loonud võimaluse arendada algoritme, mis ennustavad tulevasi protsesse ajalooliste sündmuslogide põhjal, et parandada operatiivset efektiivsust. Edasised täiustused on tulnud reaalajas preskriptiivsete protsessi jälgimise tööriistade kujul, mis tuvastavad volatiilseid protsessi juhtumeid ja soovitavad sekkumisi tulevaste tegevuste ennetamiseks riskantseks muutumise korral. Üks peamisi väljakutseid selles valdkonnas on sekkumiste selge ja lihtne edastamine lõppkasutajatele. Käesolev uurimus viidi läbi reaalajas tööriista arendamiseks, mis visualiseerib saadaolevaid sekkumisi ja nende mõju protsessi juhtumile. Tööriist aitab toetada protsessianalüütikuid tegevuste tuvastamisel, mis parandavad KPI-d või takistavad protsessi alaoptimaalse tulemuse saavutamist. Selle tulemusel valmib operatiivne juhtpaneel - Kairos - mis visualiseerib preskriptiivse protsessi jälgimise väljundit.

Võtmesõnad:
äriprotsessid, visualiseerimine, preskriptiivne protsessi jälgimine

CERCS: T120, Süsteemitehnoloogia, arvutitehnoloogia

Contents

1 Introduction

Business processes represent a crucial asset for organisations. They describe the jobs, responsibilities, and tasks carried out in tandem to ensure consistent results and produce value for the corporation [15]. Business processes can take many forms, one of the most common types being *order-to-cash process*, which starts when a vendor submits a request to purchase a product or service and ends when the vendor receives the requested goods and makes payment. Any interruptions to the business process may lead to the organizational ecosystem coming to a standstill, influencing customer experiences and the attractiveness of products and services, ultimately impacting revenue within an organisation. Poorly designed processes, which lead to operational failure, create a competitive gap within a business sector: one that is filled by organizations that improve their business process design and eliminate inefficiencies.

The race for competitive advantage coupled with the recent trends of globalization, standardization, innovation and agility has increased the demand for business process improvements, giving way to Business Process Management (BPM) [15]. The BPM lifecycle comprises six phases: process identification, discovery, analysis, redesign, implementation, and monitoring. The cycle aims to analyze current processes to propose and implement improvements that address the discovered failures in a manner that optimises performance measures.

As the organizations grow and their processes become more complicated, there emerges a need for automation of BPM lifecycle phases that are too resource-heavy. One way to do this is process mining, during which, the data generated by process executions is used to gain insight into process performance and verify its compliance with the regulations [15]. Observing and detecting deviations from the intended performance or execution allow process workers, participants and analysts to identify how and why these aberrations occur and the best ways to fix them. The analysis can be done after the completion of processes or during their run-time. Process mining has been further expanded to not only provide performance measures and other important analytics during the process run-time but also use machine learning techniques to generate predictions about future performance. Predictive process monitoring can be seen implemented in a business process performance dashboard like Nirdizati [26], which allows process analysts to see in real-time how the execution of a business process instance is predicted to develop in the future.

Efforts have also been made to expand the scope of predictive process monitoring by developing prescriptive process monitoring (PrPM), which focuses on recommending future steps based on machine learning algorithms. PrPM aims to support process analysts in making correct decisions in a business process instance to optimise a key performance indicator (KPI) of interest [25] or ensure a desirable outcome. The systems implementing PrPM offer three functional components: monitoring ongoing cases in the background (process monitoring), predicting future activities (predictive process monitoring) and prescribing recommendations for better performance [23].

Monitoring and predictive analytics have been at the forefront of academic research, with little resource directed towards exploring how to best communicate recommendations to the end users [23, 24, 18], resulting in almost all prescriptions being recommended using a textual synopsis. Research also claims that process participants do not always follow the recommendations when interacting with prescribed interventions, which could be due to their subjective perception of the process or misinterpretation of the presented information [21]. Thus, there exists a problem of representing PrPM output effectively.

In light of the aforementioned, the research goal of this thesis is defined as follows: **RG.** *implement a useful and user-friendly application that visualises PrPM output.*

The contribution of this thesis is an artefact that is able to receive, process and format the output of a PrPM tool and construct a useful and user-friendly visual overview for each ongoing case in real-time, including case performance, events associated with the case, possible prescriptions and metrics relevant to business process improvement decision-making. Process analysts could use the contributions to identify improvement opportunities in the process based on the recommendations given in the cases in order to optimize a KPI, or mitigate an unfavourable execution. Since PrPM methods have not yet reached the maturity to be implemented in the organizations [21] and, thus, be used by operational workers or tactical managers, process analysts can be expected to benefit from the methods first, making them the focus of this research.

The artefact is implemented using the design-science approach. First, we elicit the requirements based on previous research [38]. Next, the visualisation tool is implemented using a modern JavaScript web development framework (Vue), Python framework (Flask) and a NoSQL database (MongoDB) to accommodate the variable nature of the prescriptive output. The artefact is then evaluated for effectiveness and ease of use by conducting testing with its potential users (process analysts). The results demonstrate that the visualisations are user-friendly and useful, as well as provide insights into how to improve their design and functionality.

The thesis is structured as follows: section 2 outlines the background, with relevant concepts, such as business process management, PrPM and visualisation, as well as works related to this thesis. Section 3 describes the methodology and approach to research and implementation. Requirements are elicited and prioritized in section

4. Section 5 details the development process and results. In section 6 we conduct testing, discuss the results of the thesis in the context of relevant fields, outline potential improvements and present limitations. Finally, Section 7 summarises the findings of the study.

2 Background and Related Work

This section provides insight into the key concepts used in this thesis, such as business process management, process mining, prescriptive process monitoring and visualisation, as well as related work.

2.1 Background

2.1.1 Business Process Management

Business processes are an integral part of any organization, regardless of its field or specialization. They represent the routine operations performed to ensure seamless delivery of products or services. Dumas et al. [15] define business processes as "the chain of activities, events, and decisions which are performed by the organisation to deliver value to its customers". Some of the most common types of processes include *order-to-cash* process, which starts when a customer submits a request to purchase and ends when the goods are delivered and payment made; and *issue-to-resolution* process - starting when a customer files a complaint and ending when all parties involved deem the issue resolved [15]. Regardless of direct customer involvement, business processes impact the quality of service and overall organizational efficiency, which can affect profitability. Consequently, organisations seek business process improvement opportunities to eliminate inefficiencies in their daily operations and gain a competitive advantage.

To achieve this, organisations employ business process management (BPM), defined as "a body of principles, methods, and tools to discover, analyse, redesign, implement, and monitor business processes" [15]. The definition directly describes the BPM life cycle, which consists of 6 phases (figure 1). In the first phase - *Process identification* - a business problem is posed and the processes relevant to it are identified. This phase serves to better understand the architecture and inter-relations of the business processes in an organization, as well as select which processes to scrutinize in the later stages of the life cycle. *Process discovery* aims to document the current state of identified and selected processes in the form of an as-is-process model, commonly by using business process management notation (BPMN). In *process analysis* issues associated with the aforementioned model are identified by quantifying them using performance measures and prioritising them according to their potential impact and the estimated effort required to resolve them. *Process redesign*, also known as *process improvement* aims to suggest improvements needed to address the issues identified in the preceding step. The improvement opportunities are analyzed using performance measures and retained with the goal of achieving organizational objectives. The retained changes are combined with the *as-is process model* to document a *to-be process model*. During *process implementation* the new process model is implemented in two stages: organisational change management, which aims to alter how the participants in the process work, and automation, which refers to the development and deployment of the systems that support the process. *Process monitoring* is the last phase of the life cycle, where the process implemented in the previous step is monitored during its execution to collect data and analyse it with respect to performance measures and objectives. Data helps identify inefficiencies, bottlenecks, and possible deviations from the intended process. For example: in the executed process an invoice is paid out before its approval, contradicting the as-is model.

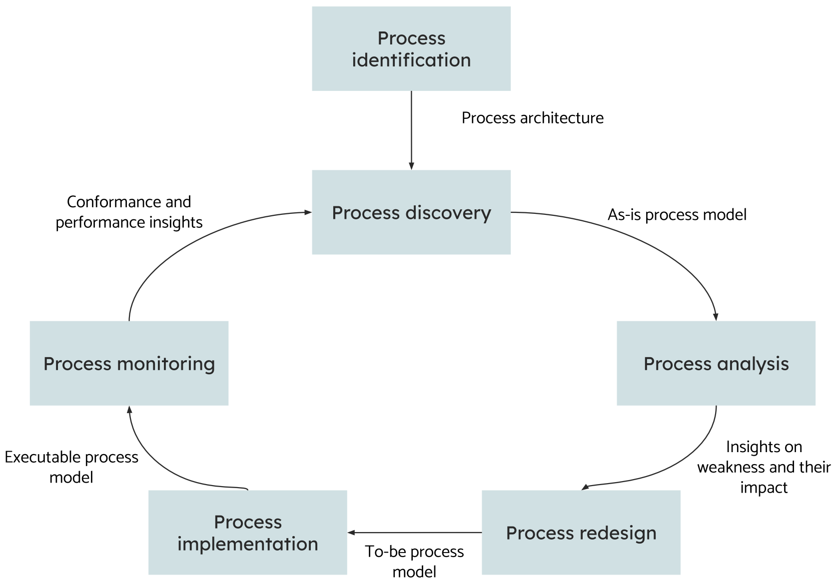


Figure 1. BPM life cycle [15]

The results of process monitoring can be used as input for process discovery, hence the cyclical nature of BPM. As companies grow and their processes become more complicated, it becomes more important to maintain a streamlined operation of all organisational components. To implement improvement opportunities that complement the complexity of the business process, BPM uses precise health measurements. These metrics are derived from common process performance dimensions: time, cost, quality, and flexibility - usually represented as *Devil's Quadrangle* [15]. In an ideal setting, one would want to decrease time and cost and increase quality and flexibility. Each aspect of performance can be broken down into process performance measures, known as key performance indicators (KPIs). Given enough data to perform the calculations, the KPI can be determined for any given business process [15]. To better link BPM to the improvement of process KPIs, Aalst et al. [14] suggest exploiting event logs, which are chronological collections of recorded events. Each event within the log represents a state change that has occurred in the context of a single business process instance [15]. According to [23] "Each trace describes the life-cycle of a particular process instance (i.e., a case)", which is ascribed case attributes, a set of events (i.e., executed activities) and their properties (e.g. resource that executed the event). While the formatting of event logs can vary across business entities, it is crucial that the following variables are present in the data:

1. Activity that was executed,
2. Identification of the case the activity belongs to,
3. Timestamp at which the activity was executed.

Event logs are a starting point of process mining.

2.1.2 Process Mining

Until recently, most of the stages of the BPM life cycle were considered to be time-consuming and resource-heavy, as the model construction and issue identification had to be done manually. However, the emergence of *process mining* alleviated and even eliminated the need for human involvement in some stages of BPM life cycle and provided a way to automate most tasks. Process mining is defined as "a family of techniques to analyze the performance and conformance of business processes based on event logs produced during their execution" [15]. It is a data-driven technique used by organizations to gain insight into their business processes. It involves extracting information from event logs and analyzing it to discover process models, identify bottlenecks, and improve process performance. By visualising and analysing the actual behaviour of processes, organisations can identify inefficiencies, deviations, and variations in their operations, which can help them optimise their processes and make informed decisions about potential improvements. Process mining enables organisations to achieve greater efficiency, reduce costs, and improve customer satisfaction. With the increasing use of automation and digital technologies, process mining has become an essential tool for organisations looking to streamline their operations and gain a competitive edge.

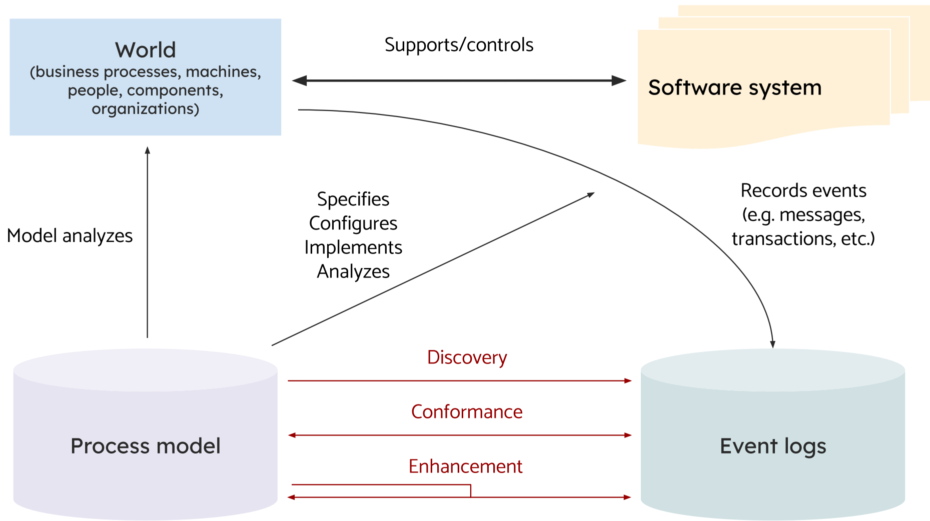


Figure 2. Aspects of process mining [9]

Process mining is designed to take an event log and perform process-oriented analysis, which consists of 3 main parts (figure 2): process discovery, conformance checking and operational support [24, 25]. Process discovery extracts models from event logs that closely describe the underlying structure of a business process. This is done by taking an event log and processing it using machine learning algorithms to construct a diagram in a variety of notations: petri-nets, BPMN, directly-follows graphs, and process trees, of which the former two are most commonly used. [20, 24]. Conformance checking aims to show how well the discovered model captures the process data or vice versa: to what extent the event data is in accordance with the process model [25, 24]. Deviations from the process model are discovered in this step. Finally, operational support aims to offer additional information on the trends in the discovered process to support enhancement or betterment of the model [24]. The information could include execution frequencies, times, or other important KPIs, as well as bottlenecks discovered in the process. This information is used to improve processes and mitigate problems that have occurred previously. The result of this step is an enhanced process model.

2.1.3 Prescriptive process monitoring

While process mining focuses on static process analysis, the main purpose of prescriptive process monitoring is to provide proactive and corrective measures for the betterment of a process instance. Eili et al. [24] discover that there are two main formats for prescriptive process monitoring:

1. *prescriptive process monitoring for process design*, which focuses on providing recommendations during business process modelling, making the time-consuming and error-prone task less resource-intensive and manual; and

2. *prescriptive process monitoring for real-time process execution*, which aims to identify volatile cases and automatically provides a recommendation for possible optimal activities or resources to the ongoing case instance, with the aim of optimising a KPI or minimising a negative result.

This thesis focuses on the latter.

Objectives of prescriptive process monitoring can be divided into two categories: reducing the defect rate and optimizing quantitative case performance [25]. The former refers to mitigating a risky outcome, while the latter deals with optimizing a KPI of interest. Kubrak et al. [25] highlight several examples of defect rate objectives: violation of cycle or processing time being one of them. While case duration can be used as a quantitative case performance indicator, it can also be ascribed the role of a categorical outcome. More examples of this objective include: a customer rejecting delivery in a manufacturing company, a patient entering the critical stage in a hospital, or medical mistakes due to patient restrictions [25]. Optimising quantitative case performance, on the other hand, refers to optimising a temporal indicator (i.e. a KPI): usually the cycle time or the processing time. However, Kubrak et al. [25] mention several other indicators, such as quality as perceived by the customer and customer lifetime value.

The result of prescriptive process monitoring are prescribed interventions - recommendations made for the next potentially best activity or resource in a process instance when a possibility of a risky execution is detected. The interventions are categorized as follows: control-flow perspective and resource perspective [25, 24]. Interventions concerning control flow prescribe recommendations as the next activity or a sequence of activities in a process instance. Following the intervention can improve execution time, treatment quality or customer service [25, 24]. The resource perspective aims to assign a specific resource to the next activity, and, thus, its goal is to find the most suitable resource for a specific task given their schedule, availability, and experience [25, 24].

Another aspect to consider is intervention frequency, which refers to when the interventions are prescribed. Kubrak et al. [25] define two main frequency methods: discrete and continuous. The former method prescribes interventions only when a risky execution is detected (e.g. probability of a negative outcome exceeds a threshold), while the latter does so for all activities in an ongoing process instance.

2.1.4 Visualisation

The improvement opportunities suggested by the prescriptive process monitoring system need to be analysed by process analysts before they are put into practice. visualisation can aid in facilitating the understanding of improvement implementation, as it improves perception and comprehension of structures and patterns [22]. Further research reveals the principles for designing effective visualisations that aid users in quickly and easily interpreting complex data. *Visual information seeking mantra* by Shneiderman [4] and visualisation *rules of thumb* by Munzer [12] provide guidelines on how to approach graphic representations in broad strokes. The former consists of four rules that design effective visualisations: first, the *overview-first* principle suggests providing an overview of the entire dataset to give users a sense of its size and organisation. Secondly, the principle of *zoom and filter* allows users to zoom in and out of the data and filter out irrelevant information. Thirdly, the *details on demand* principle allows users to retrieve more detailed information as needed. Finally, the *relate* principle emphasises the ability to link related information across multiple views or datasets, allowing users to see the intra-relations within the data.

On the other hand, Munzer's rules of thumb present six design principles for the creation of effective visualisations. These principles include: knowing the audience and designing the visualisation to meet their needs and level of expertise; selecting the appropriate chart or graph type based on the data and the message being conveyed; using colour and contrast effectively to guide the user's attention; avoiding clutter and unnecessary complexity; ensuring that the visualisation communicates a clear message and supports the intended interpretation of the data; and iterating and refining the design based on user feedback until the visualisation effectively communicates the intended message.

Previous research into the topic of visualising prescriptive process monitoring output [38], in which this thesis finds its basis, leverages these visualisation principles to create wireframes of the tool described in section 5. The guidelines are also highlighted in the research conducted by Kubrak et al. [27] in another study for designing IRVIN - a prototype for supporting process analysts in working with process improvement opportunities.

2.2 Related Work

Similar tools and frameworks are developed by the authors of several different papers, which are discussed in this section.

Seeliger et al. [18] present a tool that visualises PrPM output by implementing a visual recommender system, which prescribes interventions during "interactive visual inspection of discovered process model". Implemented as a ProM plugin, the tool provides a visual interface for viewing interventions for discovered process models and allows process workers to configure the view by adjusting trace- and cluster-based measures, which are used to calculate the interestingness score, defined as "the difference of means between the process instances in a cluster and a reference set" [18]. However, the plugin is meant to be used as PrPM for process design rather than real-time process execution, which is the focus of this paper.

Schonenberg et al. [8] propose an intelligent process worker assistance in process-aware information system (PAIS) that provides recommendations for possible next steps in the process. The approach details a ProM plugin which uses information from past event logs to guide a process worker through a process execution. This log-based tool is later updated to suggest additional strategies for providing recommendations [10]. However, the research does not focus on how the recommendations are delivered to the end user, as there is no framework put into place for visualising the prescribed activities.

A prototype for the next step recommendation system is designed and implemented by Huber et al. [13]. The tool uses process mining to provide recommendations and predictions in Adaptive Case Management (ACM) systems. The results of the analysis can be viewed on the Collaborative Case Management (CoCaMa) platform, the functionality of which is extended by a prototype of custom plugins. In the user interface, the tool displays possible deadline violations, prediction on how much time is remaining in each case, and how well the case execution supports given case goals. Process workers are able to select the optimal activity for each case by evaluating the information on similar case executions [13]. However, the tool is limited to prescribing one recommendation at a time and does not show proof of useful assistance for process analysts, who are the focus of this study.

Sirgmets et al. [19] conduct an analysis on how existing process mining output is visualised and the underlying principles of design. The visualisations analysed by the study, which most commonly utilize process diagrams, do not often follow visualisation guidelines, and thus, Sirgmets et al. [19] propose a framework to support developers in designing process mining output. While the framework is based on common practices of data visualisation, it proposes guidelines for design rather than an artefact and emphasises process mining rather than PrPM.

Kubrak et al. [27] propose a prototype of an artefact that helps process analysts identify, prioritise and communicate business process improvement opportunities. The study conducts interviews to elicit requirements and provide insight into how process analysts work with process improvements using the output of process mining tools. The findings are consolidated into a set of guidelines for visualising improvement opportunities; based on them, the authors propose a mockup - IRVIN, which is evaluated by industry practitioners to determine the usefulness of the prototype. Despite the contribution of the research, it does not propose an artefact for visualising PrPM output.

3 Methodology

This section provides an outline of a design science research methodology (DSRM), the motivation for its selection, and a description of its implementation for this thesis.

Design Science focuses on the desired state, its objective is achieved by designing artefacts, which can be instantiations, methods, models, or concepts, that solve unresolved problems [7]. Thus, DSRM was selected as a viable approach to solving the problem of a lack of effective visualisations of PrPM output. According to the DSRM guidelines [7], this thesis aims to achieve its goal - to implement a user-friendly and useful application that visualises PrPM output - by following the outline: *problem identification*, *defining objectives*, *design and development* and *evaluation*. The complete structure of the research process is defined in figure 3.

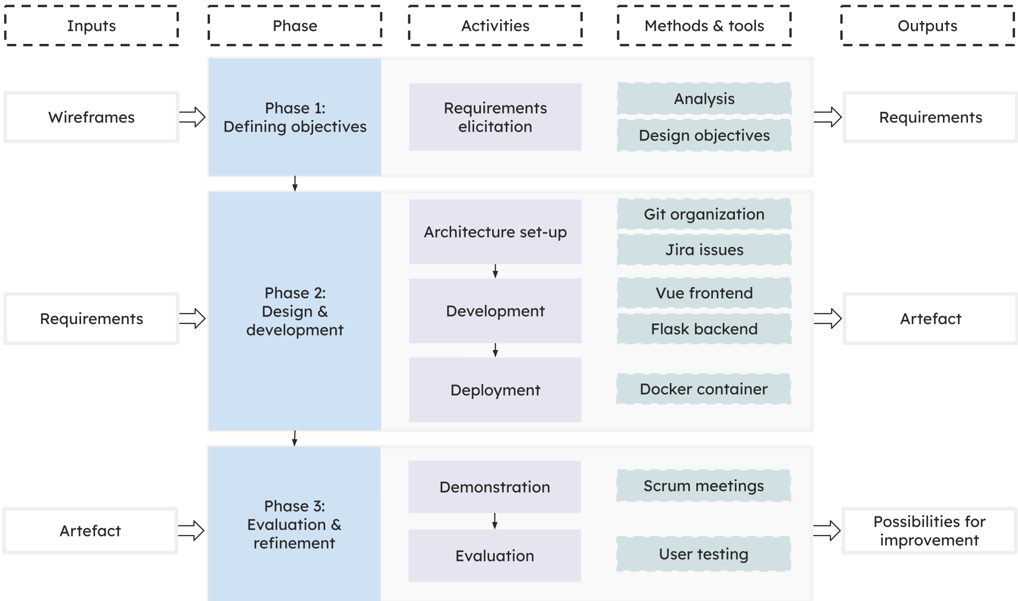


Figure 3. Research process

3.1 Problem identification

In the DSRM methodology, our first objective is to define the research problem and justify the value of a solution [7].

Exploration of related work reveals that there are few PrPM tools that explore how recommendations should be communicated to the user since most academic research in the field of business process management is focused on process discovery and mining. Most of the solutions choose to implement textual synopsis, including only the next prescribed activity (or resource). While it can be useful, the recommendations prescribed in this manner are sometimes difficult to understand, visualise within the process or prioritize over other recommendations without any metrics that assist in decision-making (such as effect and accuracy). This might hinder the decision-making of process analysts, who are the focus of this study since they are conjectured to benefit from PrPM methods first [21].

Previous research demonstrates the implementation of recommender systems that prescribe interventions in a business process instance, with little attention paid to visualising prescriptive solutions for process analysts in real-time. Dees et al. [21] claim that most recommender systems assume that the process participants follow the prescribed recommendations, but this is not always the case, since interventions are selected based on human judgement, which relies on a subjective perception of the process. The study also shows that the manner in which the recommendations are delivered might not allow for correct interpretation or ease of understanding, thus leading to the problem of representing business process prescriptions effectively.

3.2 Defining objectives

Objectives definition aims to specify the objectives of a solution based on the problem definition [7].

To alleviate the problem mentioned in problem identification (section 3.1), the goal of this paper is to implement a useful and user-friendly application that visualises PrPM output. The visualisations differ based on the target user, but, for the scope of this thesis, an emphasis is made on the process analysts. In previous unpublished research, conducted by Kubrak et al. [38], the authors create wireframes that visualise PrPM output, which is used as a baseline for the design and implementation of the tool presented in section 5. The wireframes are designed using visualisation principles, including those of Munzner [12] and Shneiderman [4] to communicate information on case-specific prescriptions effectively and efficiently. Thus, the following design objectives (DOs) are inferred from the study:

1. **DO1**: Display an individual case along with completed activities in the trace, case attributes and prescriptions recommended at the current stage in the process.

2. **DO2**: Construct a BPMN diagram, as it is a commonly used notation for

presenting process models [20, 24], to display the ongoing process instance. Diagrams must be constructed in a way that differentiates between completed activities and interventions, ensuring that recommendations are highlighted and discernible. The difference can be seen in the user interface of a similar tool by Seeliger et al. [18], where the intervention is highlighted in grey colour.

3. **DO3**: Display metrics associated with prescribed recommendations, so that the process analyst is able to evaluate and select one they deem to be most suitable.

4. **DO4**: Update the case information when a new event occurs in the trace.

The design objectives are further refined by considering the tool that secures the provision of PrPM output. The method and usage of this PrPM tool are described in section 5. In order to elicit requirements from the design objectives set forth by Kubrak et al. [38], we break down the DOs into smaller, more specialized tasks following the principle of "one thing per requirement" - this makes the development more streamlined and efficient. The requirements are assigned a unique identifier and categorized as a task using Jira issues. They can be seen in section 4.

3.3 Design and development

Design and development aims to Implement the artefact [7].

The chosen architecture for the development of this visualisation tool follows the common frontend-backend structure, since the baseline uses a webpage format to construct wireframes [38]. The backend is to receive the PrPM output, process it, calculate any necessary additional metrics, and record it for the frontend to access and construct visualisations. The technology stack is selected based on the ease of use, past experience of the author and fit-for-task. Thus, the following technologies are chosen:

- *Git organization/repositories*: for version control, storing and collaboration. Github is one of the most popular version control frameworks [31].
- *Jira issues*: Backlog manager and Kanban board for managing and prioritizing tasks for lean software development [37].
- *Vue*: Javascript framework for frontend architecture, with component-based structure, which encourages modularity [36].

- *Flask*: Python framework for backend architecture, with an extensive pool of libraries for working with machine learning algorithms - beneficial for additional features in the future [30].
- *MongoDB*: A NoSQL document-oriented database program that uses json-like objects to store data. Especially useful if an application requires flexibility and uncertainty of inputs [42].
- *Docker*: A platform for delivering applications in packages called containers, which makes them easy to install and manage [29].

Additionally to the technologies chosen for implementing the visualisation tool, we choose software for providing PrPM output, since our focus is the visualisation and delivery of information and the implementation of prescriptive algorithms is out of the scope of this thesis. We consult industry experts for potential PrPM tools to secure the provision of prescriptive output.

The process of implementation employs the methodology of agile development since the project does not have a rigid structure and its details are subject to change, which calls for a flexible approach. Each iteration is to last 2 weeks, followed by a meeting with the research team to review the progress and provide updated requirements, or feedback for improvement.

3.4 Evaluation

Evaluation aims to observe and measure how well the artefact supports a solution to the problem [7].

As the artefact is implemented to assist process analysts, they must be the ones to provide opinions about its effectiveness. To evaluate whether the visualisations are helpful in practice and identify the strengths and weaknesses of the interface, we conduct user testing of the application. We coordinate the user study by letting industry practitioners evaluate the application developed in section 5. The evaluation is conducted in collaboration with the research team.

There are 3 practitioners who participate in the evaluation; the interviewee overviews with their ascribed codes can be viewed in table 1. The interviews with each participant are held online (Zoom, Skype) and last for 20 to 40 minutes. The interviews are recorded and later transcribed using otter.ai [32] to use for thematic analysis of the text [6]. This helps us identify themes and patterns in qualitative research in a rigorous and systematic manner without sacrificing transparency.

Code	Domain	Experience (years)
I1	IT industry	4
I2	Banking, Finance services, or Insurance	6
I3	Consultancy services	5

Table 1. Evaluation user study participants

The interviews are conducted in a mixed approach: first, the interviewer provides an introduction to the visualisation tool and its basic features. Then the participants are given the freedom to explore the recommendations for two cases of their choice. They are encouraged to vocalize their thoughts while interacting with the interface: this allows the interviewer to conduct an inspection of the interaction by remarking on the following three categories: "Cases list" (how the interviewees analyse the cases list and its elements), "Decision-making for recommendations" (what criteria the participants use for determining the optimal recommendation) and "Process model" (how the participants interact with the process model).

During the exploration of the interface, the participant is asked the following questions based on the RG. Starting from questions related to the interaction, e.g. *"Why did you do [X]?"*, progressing to inquiring about the usefulness of the information included in the interface, e.g. *"Which information did you find most/least useful and why?"*. Questions are also asked about the participant's perception of the application and its components, e.g. *"Which aspects of the interface did you have trouble with?"* and *"What worked in the way you expected it to and what did not?"*, as well as whether the visualisation tool would be useful in a daily work of a process analyst: *"Would this tool have been helpful in the project? What tasks could you imagine using it for?"*. The final questions aim to let the participants voice their opinions on limitations and possible improvement opportunities: *"What information is missing in the interface?"* and *"What could be improved about the tool?"*.

Following the demonstration and the interview, the participants are given a questionnaire to describe their experience with Kairos. The questions are constructed using the System Usability Scale [2] for ease of use and the 5-point Likert scale [3] for the perceived usefulness [1] of the system. The results are used for calculating usability and usefulness scores in accordance with the method presented in [2]. While perceived usefulness scores are not commonly derived this way, this approach allows for standardization of the results. The score contributions of all positive statements (1a, 1c, 1e, 1g, 1i, 2a, 2b, 2c, 2d, 2e) in the user survey (appendix 7) are scale positions minus one. For example: if the survey participant selects "strongly agree" as their answer, the score contribution of the question will be 4. For negative statements (1b, 1d, 1f, 1h, 1j) the contribution is five minus the scale position. For example: with the answer of "strongly disagree" to one of these questions, the score contribution will be 4. The scores of each question are summarized separately for the usefulness and usability portions of the survey.

The survey also includes demographic questions for estimating the domain of operation and experience of the interviewee with machine learning, process mining and PrPM.

The findings of the evaluation are presented in section 6.

3.5 Use of AI

The study makes use of a text-generating AI language model - ChatGPT [35] to obtain a source overview of some of the cited references (e.g. "Summarize Munzer's 'Visualisation Analysis and Design'." [12]), get specialized answers faster than Google would be able to provide (e.g. "How long should a Software Engineering master's thesis be at the University of Tartu?", "How do I conditionally format a Latex table?") or ascertain structural correctness of written text (e.g. "Is this a correct statement? 'These visualisation principles are utilized by previous research about visualising prescriptive process monitoring output, which this thesis finds its basis in.'"). The output of these questions is not utilized as a part of the text in this thesis.

The study also makes use of AI-based aid - Grammarly - for optimizing spelling, grammar, punctuation and clarity of the used language [33].

4 Defining objectives

This section describes the implemented visualisation tool and all its features. The application is a form of an *operational dashboard* [15] since it displays the status, performance and recommended actions for ongoing or completed cases to facilitate planning and decision-making for process participants, more precisely - process analysts.

4.1 Requirements elicitation

In order to define the objectives of this software, we analyze the wireframes produced by Kubrak et al. [38] and, depending on the DOs presented in the methodology, propose requirements for implementing the application. The requirements are categorized into the following epics:

1. *Uploading*: uploading a log file(s), defining column types and parameters.
2. *Dashboard*: selecting an event log, viewing the log details (status, parameters), starting/stopping the simulation and clearing the streamed data or getting all prescriptions at once.
3. *Cases*: displaying, sorting and filtering the table of cases.
4. *Case*: displaying a view of a single case, including case performance, case attributes, process diagram and recommendations.
5. *Recommendations*: displaying all current recommendations and exporting them as a .csv.
6. *Deployment*: deploying the application to a public URL.

The requirements are elicited in table 2 and prioritized using the MoSCoW approach to define the ordering for executing requirements; this nominal scale prioritization mechanism helps to define the most valuable features that are critical to implement first and differentiates them from the trivial ones [5]. MoSCoW technique is one of the most commonly used methods for requirements prioritization. It classifies requirements into the following four categories: must have (requirements crucial to a successful project), should have (high-priority requirements of import), could have (desirable but not necessary requirements) and won't have (requirements that will not be implemented) [16]. Previous research utilizes the same prioritization method [38, 27].

ID	Requirement	Prioriti-zation
	Uploading	
R1	The user can upload a log in .csv, .xes or .zip format	must
R2	The user can upload a test log in .csv, .xes or .zip format	should
R3	The user can define column types	must
R4	The user can define parameters: case completion, positive case outcome, intervention and alarm threshold	must
R5	The user can assign a description to the group of defined parameters	could
	Dashboard	
R6	The system displays the uploaded log on the dashboard	must
R7	The system displays the status of the selected log in real-time	should
R8	The system displays the defined parameters on the dashboard	should
R9	The user can select a log on the dashboard	must
R10	The user can start the simulation of the selected log on the dashboard if the log does not have a test set	must
R11	The user can stop the simulation of the selected log if the log does not have a test set	must
R12	The user can clear streamed data of the selected log on the dashboard	could
R13	The user can delete the selected log on the dashboard	could
	Cases	
R14	The system displays the count of ongoing and completed cases	could
R15	The system displays the selected KPI	could
R16	The system displays ongoing cases with case ID, recommendation availability, KPI, intervened status and case attributes	should
R17	The system displays completed cases with case ID, recommendation availability, KPI, intervened status, outcome and case attributes	should
R18	The system can sort the table of cases by KPI	should
R19	The system displays a pie chart for showing how many recommendations were successful	could
R20	The system updates the view whenever a new event is received	must
	Case	

R21	The system displays the defined KPI	should
R22	The system displays the case performance	must
R23	The system displays the case details (attributes)	should
R24	The system updates the view whenever a new case event is received	must
Process analyst view		
R25	The system displays the current recommendations	must
R26	The system displays the past recommendations	must
R27	The system displays the status of the past recommendations	should
R28	The system displays whether a recommendation is recommended now or not	must
R29	The system displays the calculation details of a selected recommendation	must
R30	The system displays the process model	must
R31	The system displays the activity name and timestamp on the past activity nodes in the diagram	must
R32	The system displays past activity, current activity, next activity and intervention in different colours	must
R33	The system displays a legend for differentiating between the different diagram nodes	must
Recommendations		
R34	The system displays the count of total current recommendations	should
R35	The system displays all current recommendations with case ID, KPI, recommendation and recommendation details as a table	must
R36	The system can sort the table by KPI	should
R37	The user can export the recommendations as a .csv	should
Deployment		
R38	The user can access the application from a public URL	must

Table 2. Requirements

5 Design and development

In this section, we describe the implemented artefact and explain how inputs and outputs are processed in order to construct the visualisations. The application is referred to as Kairos from this point onward.

It is to be noted that while the application has three views (process analyst, operational worker and tactical manager, as present in the sidebar on figure 9) for inspecting individual cases (figure 12), this research focuses on only one: process analyst view. The evaluation is also conducted for process analysts.

The source code of the application is available at Kairos Github Organization.

The developed application is available for inspection at Kairos website.

5.1 Architecture

Since the application is a multi-layered piece of software, its architecture is best described using a diagram. Figure 4 presents the high-level structure of Kairos, as well as its integration with a PrPM tool - PrCore [41], which provides the data to construct visualisations. PrCore is further discussed in section 5.1.1. The visualisation layer represents the research conducted in this thesis, namely the visualisation tool - Kairos interface, while PrCore represents the PrPM tool that was selected to provide input for constructing visualisations.

As depicted in the diagram, the visualisation layer comprises a frontend, a REST API, a backend logic and a database. The application receives data from PrCore using API endpoints presented in 7. The components of this application are further explored in the following sections.

5.1.1 Prescriptive process monitoring tool

Kairos takes PrPM output as input for constructing visualisations. In order to secure the provision of data, Kairos makes API calls to a separate PrPM tool - PrCore [41], which is briefly described in this section.

According to its documentation, "PrCore is a backend software that offers APIs to implement certain features of PrPM" [40]. The API endpoints allow users to upload logs, provide definitions for column types and parameters, and get results of the training in the form of a simulated event stream or a dataset containing recommendations for all cases. PrCore uses 4 algorithms for producing recommen-

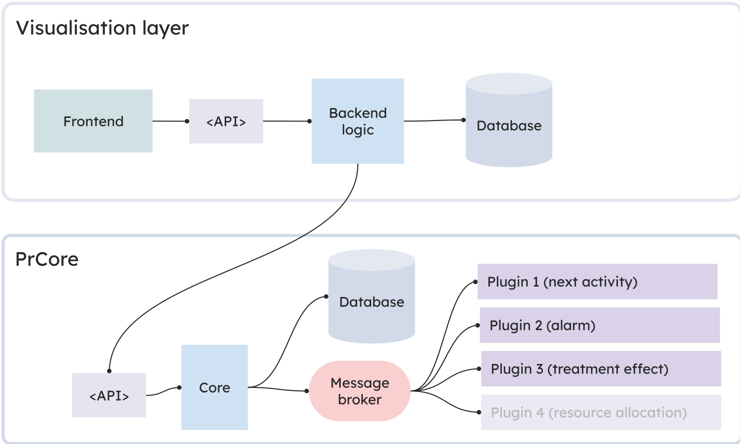


Figure 4. Application architecture with PrCore

dations for an event, however, currently, Kairos only supports three - as indicated in figure 4. The data is sent and accessed using API endpoints, which can be seen in appendix 7.

The integration between the visualisation layer and the PrPM tool can be viewed in figure 5, where sub-figure 5a described the process of uploading one log and receiving streamed events, while sub-figure 5b describes the process of uploading a train and test logs and receiving all events, as well as their prescriptions all at once.

5.1.2 Backend

In order to support the construction of visualisations (which takes place in frontend), we build a backend that connects to PrCore and a database to receive and store all the necessary data. The backend is written in Python using the Flask micro web development framework, as it allows for quick implementation of web applications using REST architecture [30]. Python is chosen for its rich codebase of machine learning and process mining libraries, which are useful for implementing additional features, something that is explored in section 6.3. For this research, it is suitable for implementing a middleware between the frontend and a PrPM tool (PrCore).

Endpoint	Method	Description
	Event logs	

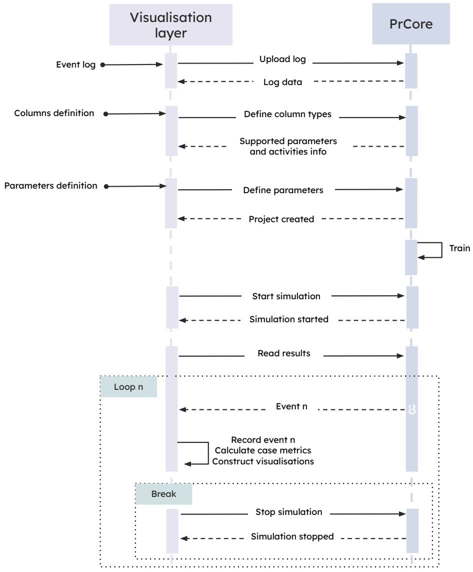
/event_logs	POST	Upload event log
/event_logs	GET	Get all event logs
/event_logs/<event_log_id>	GET	Get an event log
/event_logs/<event_log_id>	DELETE	Delete an event log
/event_logs/<event_log_id> /column_types	PUT	Define column types of an event log
/event_logs/<event_log_id> /parameters	POST	Define parameters of an event log
/event_logs/<event_log_id> /prescriptions	GET	Get prescriptions of an event log
/event_logs/<event_log_id> /status	GET	Get the status of an event log
/event_logs/<event_log_id> /simulate/start	PUT	Start simulating an event log
/event_logs/<event_log_id> /simulate/stop	PUT	Stop simulating an event log
/event_logs/<event_log_id> /simulate/clear	PUT	Clear the streamed data of an event log
/event_logs/<event_log_id> /results	GET	Get prescription results of a test event log
Cases		
/cases	GET	Get all cases
/event_logs/<event_log_id> /cases	GET	Get all cases of an event log
/cases/<case_id>	GET	Get a case

Table 3. API endpoints of the Kairos app

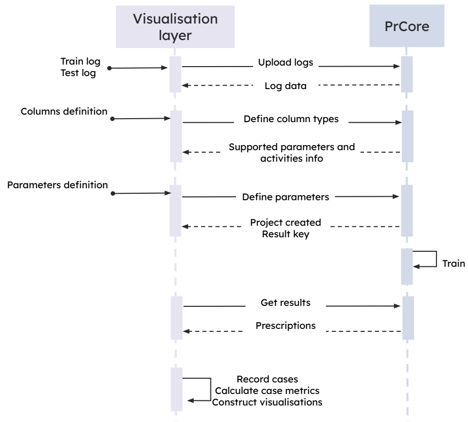
The Flask API consists of the endpoints presented in table 3. This includes the endpoints for accessing and modifying event logs and cases. The Rest of the notable functionality is concentrated around connecting to PrCore using its endpoints (appendix 7), receiving data for events and prescriptions, calculating case performance and recording necessary information in the database.

5.1.3 Frontend

The frontend of the application is written in Vue.js [36]. This Javascript framework is used for building versatile, and modular single-page applications (SPAs), which can be more fluid and responsive than multi-page applications [36]. Additionally,



(a) Uploading one log



(b) Uploading two logs

Figure 5. Sequence diagrams for two scenarios

its lightweight nature offers great performance and speed, something that is crucial for implementing a real-time application.

Table 4 outlines the structure of the frontend layer.

View/component	**Description**
HomePage.vue	Upload event log
ColumnsDefinitionPage.vue	Define columns
ParametersDefinitionPage.vue	Define parameters
Dashboard.vue	View all event logs, select event log, start/stop simulation, clear streamed data, delete event log
CasesPage.vue	View all the cases (ongoing and completed)
CasePage.vue	View individual case details, case performance, KPI
RecommendationsPage.vue	View all current recommendations
AnalyticalView.vue	View recommendations, process model, recommendation metrics

Table 4. Views and components of the Vue app

The application uses a set of libraries and frameworks to support the construction of visualisations. The process diagrams in individual case views are created using Cytoscape, a platform for visualising complex networks and integrating them with any type of attribute data [28]. The library offers a myriad of customization possibilities for the elements and the layout of the diagram, which allows us to create BPMN-like process models. Other notable libraries include ApexCharts and vue3-table-lite, for constructing common types of charts and tables respectively.

5.1.4 Database

The application uses MongoDB Atlas to store all its data [42]. This database has a flexible data model, which means that the variable nature of event and prescription data is accommodated, alleviating the need for maintaining a rigid data structure. MongoDB is also suitable for real-time applications due to its ability to handle large volumes of data and high traffic loads at a time.

5.1.5 Deployment

In order to make the application available to the public for testing purposes, Kairos is deployed using Docker [29] and a University of Tartu server. First, we build a docker container of the backend using gunicorn [34], which is a Python WSGI HTTP Server for UNIX and suitable for use for this application, since the University of Tartu server is a Linux distribution. Next, we build a docker container of the frontend using nginx, an HTTP and reverse proxy server [39]. Nginx allows for reverse proxying the requests to the backend from the client, increasing scalability, security and performance [39]. The two containers are pushed to docker hub to then be pulled into the University of tartu server. To deploy the application we create a docker-compose file and compose it after changing the routing in the server configuration so that the base URL (kairos.cloud.ut.ee) points to the frontend of our application. The docker-compose file composes the backend docker container and frontend docker container, initializing them within the same network. The deployed application can be viewed at Kairos.

5.2 Artefact

The functionality of the application can be divided into multiple parts, which are presented in this section.

5.2.1 Uploading a log

The process of using the application begins when the user uploads an event log (figure 6). The accepted file types include .csv, .xes and .zip with the file size limited to 100 Mb with the intention of minimizing performance issues and keeping run-time manageable. It is permissible that the user uploads two event logs: one for training and the other for testing; otherwise the uploaded event log is split into train and test sets. Depending on whether the test set is uploaded, the data is either streamed or delivered as a single response.

Upon uploading a .csv log, the user is prompted to input a delimiter; unless otherwise specified, the default delimiter for .csv files is a comma.

Uploading incorrect file types or files that are too large results in a warning.

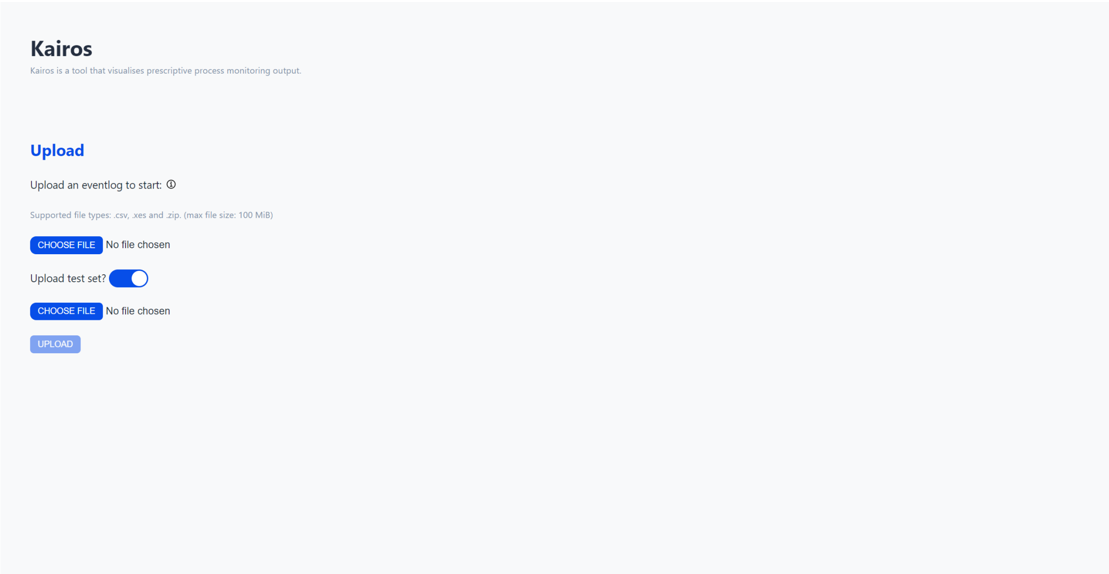


Figure 6. Upload page

Once the file(s) are uploaded, the system displays the page for defining column types along with five rows from the uploaded event log, included in the response from PrCore (figure 5). Acceptable column types are provided by the PrCore documentation [40]. The definition of available column types is available with a click on the information button next to the heading (figure 7). PrCore infers column definitions while processing the log, which means some of the column types may already be selected, but the user may rewrite them according to their preference.

By enabling the button under the type definition, the user can also define which columns are case attributes. Case attributes are displayed along with the case performance on individual case view (figure 12).

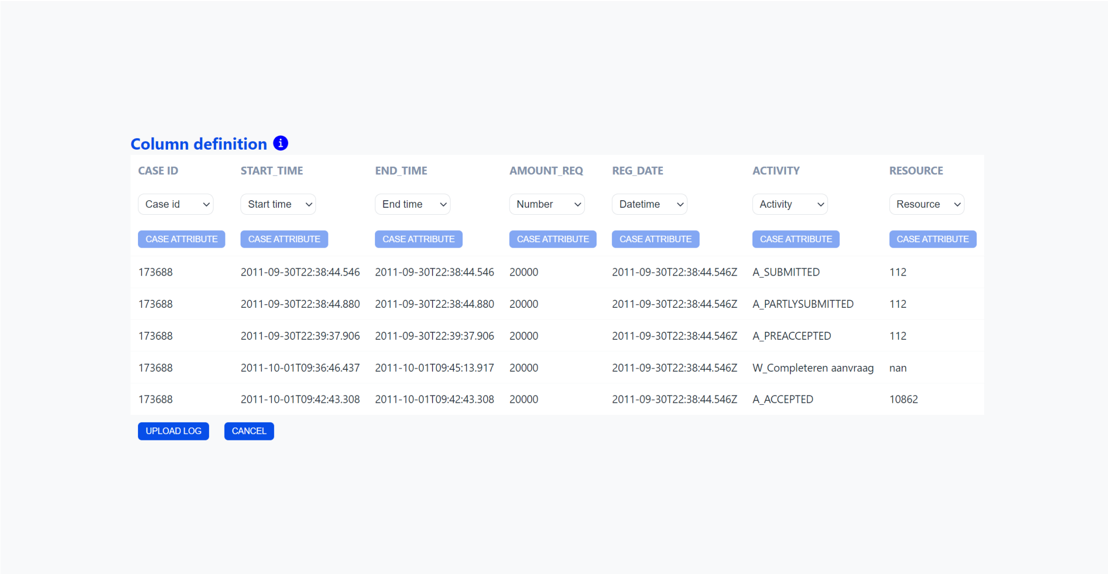


Figure 7. Column definition page

In order to train the models, PrCore requires the process participant to specify several parameters [40], which are defined as follows:

- *Case completion*: an activity marking the completion of a case. E.g. if "Application completed" is selected as a condition for case completion, then the case is marked complete when this activity is the last event in the trace. The options for case completion include all the unique activity names from the event log.
- *Positive case outcome*: a condition marking the positive outcome of a case, otherwise referred to as KPI. E.g., if equal to "duration less than or equal to 10 days", all cases that last for 10 or fewer days are marked to have a positive outcome, while the rest are marked negative. It is also used to measure case performance.
- *Intervention*: the best possible course of action for achieving a positive outcome as perceived by the user. E.g., if the intervention is "Activity equals offer-sent", then an algorithm estimates the causal effect of performing this activity at a given point in time. The causal effect may be positive or negative. If positive, then performing this activity will affect the case positively.
- *Alarm threshold*: describes the threshold for the probability of a negative outcome. E.g., if the threshold equals 0.5 and the alarm score is 0.56, then it

is recommended that the user take necessary precautions and intervene in this case to avoid a negative outcome.

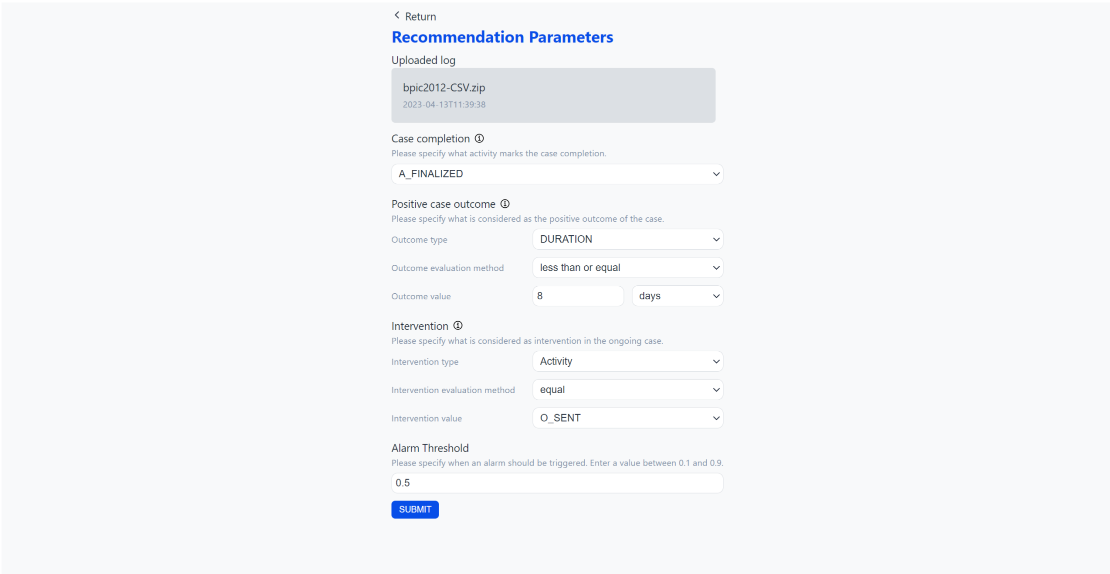


Figure 8. Parameters definition page

The defined columns and parameters are stored in the database and readily available for alterations should the user choose to make any. After defining the parameters, the user is asked to define the description of the parameters group, which is later displayed on the dashboard along with the other log information and aims to facilitate making distinctions between the event logs.

5.2.2 Event log operations

The uploaded logs along with the defined parameters are displayed in the dashboard view, as evident in figure 9. The event log operations can be found in the event log details section, which can be made use of after selecting a log from the list. The access to operations is determined by the event log status, which can take the following values (table 5).

The values presented in table 5 are not exhaustive and were modified to complement the interface. A full list of project statuses can be found in the official PrCore documentation [40].

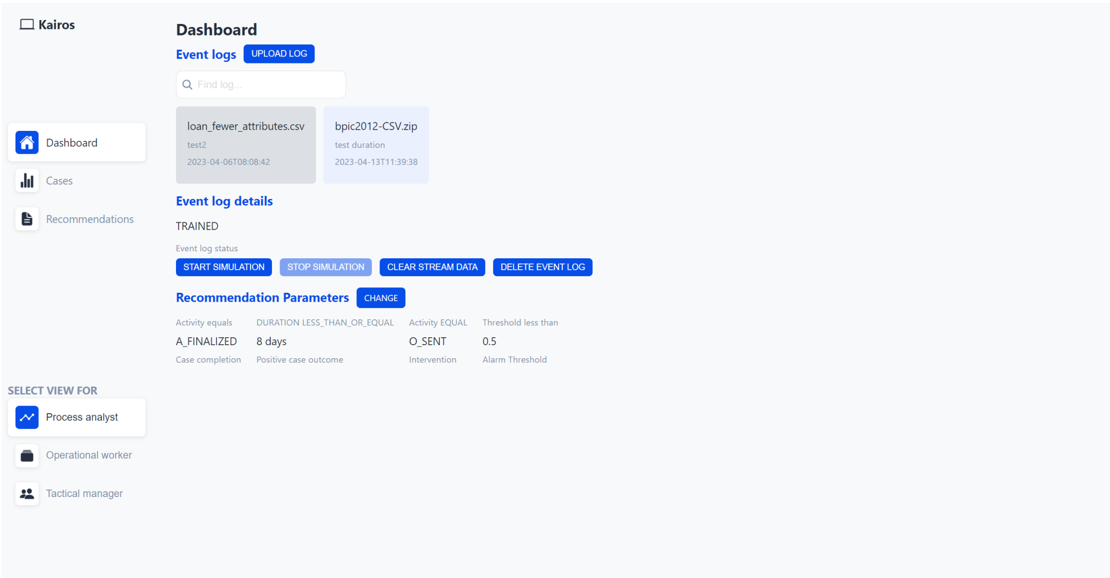


Figure 9. Dashboard view

Status	Description
`WAITING`	The event log is waiting for the new definition to be submitted
`PREPROCESSING`	The uploaded data is being preprocessed
`TRAINING`	The data is being processed by the plugins
`TRAINED`	The models of plugins are trained and ready to be used
`SIMULATING`	The log events are being streamed, and the simulation is not finished yet
`NULL`	The event log is in an error state

Table 5. Possible values of event log status [40]

5.2.3 Getting prescriptions

After a log is selected from the list, the application continuously checks for the event log status to update the view if any changes have been detected - this ensures that the process analyst is aware of the updates as they occur without having to refresh the page. The application implements control structures that prevent the user from utilizing some operations depending on the status of the event log to ensure the safe functioning of Kairos. These conditions can be viewed in table 6. It is to be noted that the `GET RESULTS` operation is only available to the user if they have uploaded a test set with the original event log: in this case, the application

Operation	Description	Allowed status
`START SIMULATION`	Start receiving streaming events and their prescriptions from PrCore	`TRAINED`
`STOP SIMULATION`	Stop receiving streaming events and their prescriptions from PrCore	`SIMULATING`
`CLEAR STREAMED DATA`	Delete all events (and cases) associated with the selected event log	`TRAINED`
`DELETE EVENT LOG`	Delete event log and any events (and cases) associated with it	`WAITING`, `PREPROCESSING`, `TRAINING`, `TRAINED`, `NULL`
`GET RESULTS*`	Get all events and prescriptions at once	`TRAINED`

* *Only available if the user uploads two logs.*

Table 6. Allowed event log operations [40]

does not stream the events with prescriptions individually, but offers the entire test set with the prescribed interventions at once - we call this a static result. If the user chooses not to upload a test set, then they are able to start receiving streamed events from the log, along with the prescribed recommendations.

The streamed cases (or static results) can be viewed from the "Cases" page, where a summary of received results is displayed (figure 10, 11). The top of the view shows ongoing and completed cases so that the process analyst is able to quickly distinguish which cases require immediate attention and which cases can be analyzed for the outcome. Switching between them can be done by clicking on either card. The KPI defined by the user is also present in this section, as it allows the process analyst to quickly compare the case performance (in this case, presented in column *Duration*) with the desired outcome. The table includes the following columns:

- *Case Id*: A unique identifier of a single trace.
- *Recommendations*: Whether the last event in the case has any prescribed recommendations or not. If so, the availability is presented in green, otherwise in grey.
- *Performance* (In this case *Duration*): Presents the case performance, which is calculated using the KPI defined by the user. Since, in this case, the user

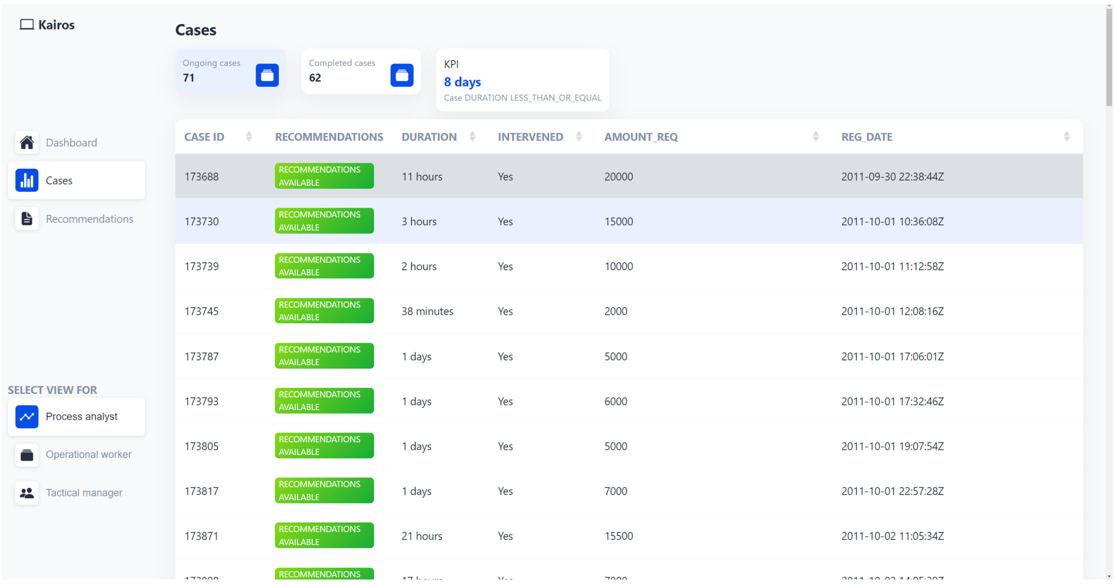


Figure 10. Ongoing cases view

chose a positive case outcome to be a "duration of less than or equal to 8 days", we calculate the duration of each case to compare to the defined KPI, which enables us to determine whether the case has a positive outcome or not.

- *Intervened*: Defined as whether any recommendation suggested by the system has been accepted, the *Intervened* column enables the process analyst to determine if the system has intervened in the case.
- *Positive outcome*: Presents whether the case has a positive or negative outcome. Can only be seen if completed cases are selected (figure 11).
- *Additional columns*: Represent the case attributes selected by the user at the columns-definition stage (figure 7).

The completed cases view has one additional feature, which can be viewed in figure 11. The section *Recommendations history* shows a pie chart for completed cases where the system has intervened and whether the outcome of these cases is negative. The graph offers process analysts a summary of the performance of the prescribed interventions and how effective they can be for a positive outcome. The possible values are described as follows:

- *Was successful*: *Intervened*: yes and *Positive outcome*: yes

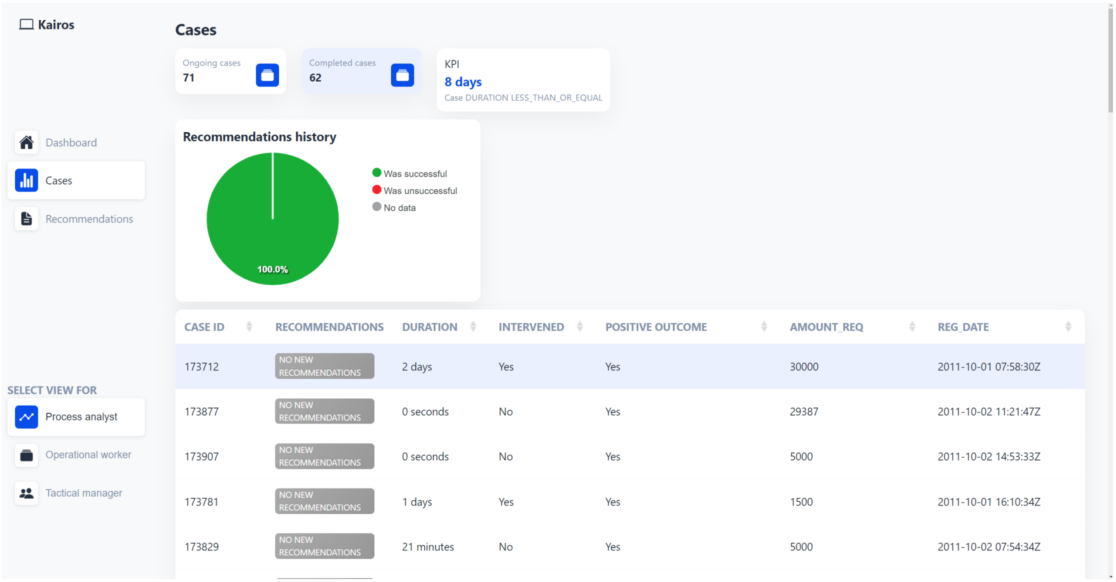


Figure 11. Completed cases view

- *Was unsuccessful*: *Intervened*: yes and *Positive outcome*: No
- *No data*: *Intervened*: yes and *Positive outcome*: unknown

In order to view the individual cases the user may click on the rows in the table. By doing so, the visited cases are marked in light blue and the case visited the latest is marked in grey.

The individual case page reveals all the details associated with a single trace (figure 12). The top of the view includes all the necessary information for case overview: KPI defined by the user, case performance based on the defined KPI and the case details (attributes). Case performance is marked in green or red colour, depending on whether the outcome of a case is positive or negative respectively.

The lower left part of the view presents the recommendations prescribed by the PrPM tool (PrCore). They are divided into two categories: current and past. Current prescriptions are associated with the latest event of the selected case, while past recommendations are associated with the rest of the events. Each event in the trace is marked as a batch and presented as a case status on top of the recommendations which were prescribed at this stage. In the case on figure 12, this is marked with "Case status: **W_Nabellen offertes** complete".

Possible prescriptions are present below the displayed case status. PrCore offers the possibility of using four different prescriptive algorithms, but the Kairos in-

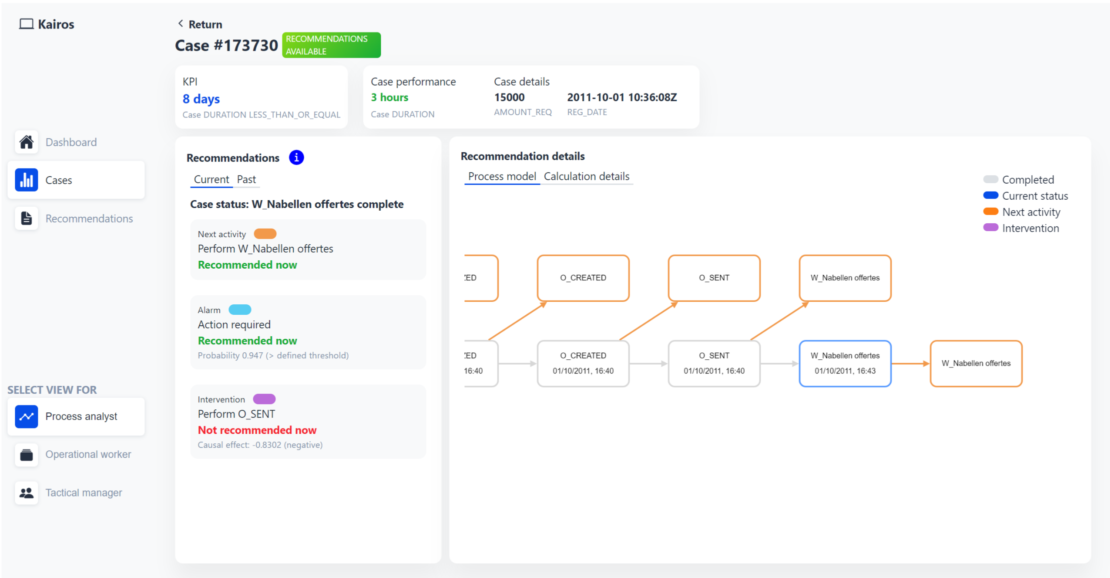


Figure 12. Individual case view with a process model

terface currently only implements three of them. The algorithm and their output descriptions are as follows [40]:

- NEXT_ACTIVITY: Predicts the next most probable activity. Returns a string, which is the name of the activity. E.g. "Perform A_Closed". This prescription type is always marked as "Recommended now".
- ALARM: Returns the probability of a negative outcome. A threshold is defined during parameters-definition (figure 8) for when an alarm should be triggered. When an alarm is triggered (the probability is more than the threshold), it means that the case is likely to lead to an undesired outcome and the prescription is presented as "Recommended now". This means that the worker should look closer into the case.
- TREATMENT_EFFECT: Estimates the causal effect (CATE score) of performing the intervention at a given point in time. The causal effect can be positive or negative. If positive, then the intervention defined by the user will have a positive effect towards achieving a positive case outcome and is marked with "Recommended now". The intervention is specified at the parameters-definition stage (figure 8).

The three recommendations are present as separate clickable cards in the recommendations list with the prescription type (e.g. "next activity", "alarm", "intervention"), the assigned colour, description (e.g. "Perform W_Nabellen offertes"), recommendation status (e.g. "Recommended now", "Not recommended now") and a metric (figure 12).

In the individual case view (figure 12) the user can also explore a process model of the selected trace. The process model comprises all the activities associated with the trace, with the latest activity highlighted in blue, as well as all the recommended activities highlighted in orange (next predicted activity) and purple (intervention). The upper left corner of the process model displays the legend for differentiating between completed activities, current activity (activity status), next most probable activity and intervention. The process model allows process analysts to better identify bottlenecks [38] and visualize an improved process model.

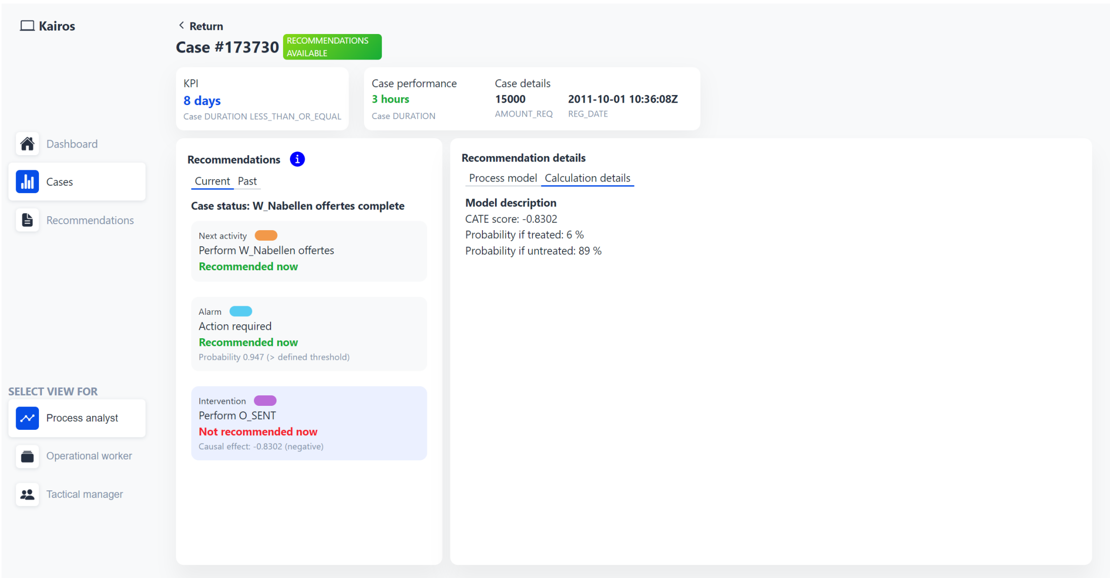


Figure 13. Individual case view with calculation details

The user may switch to the *Calculation details* perspective in the *Recommendation details* panel, allowing the process analyst to explore the metrics associated with an individual recommendation (figure 13).

In *Recommendations* view the user can view a comprehensive list of current recommendations associated with an event log. As stated before, current recommendations are associated with the final event in a single trace. Figure 14 displays the following data:

- *Case Id*: A unique identifier of a single trace.

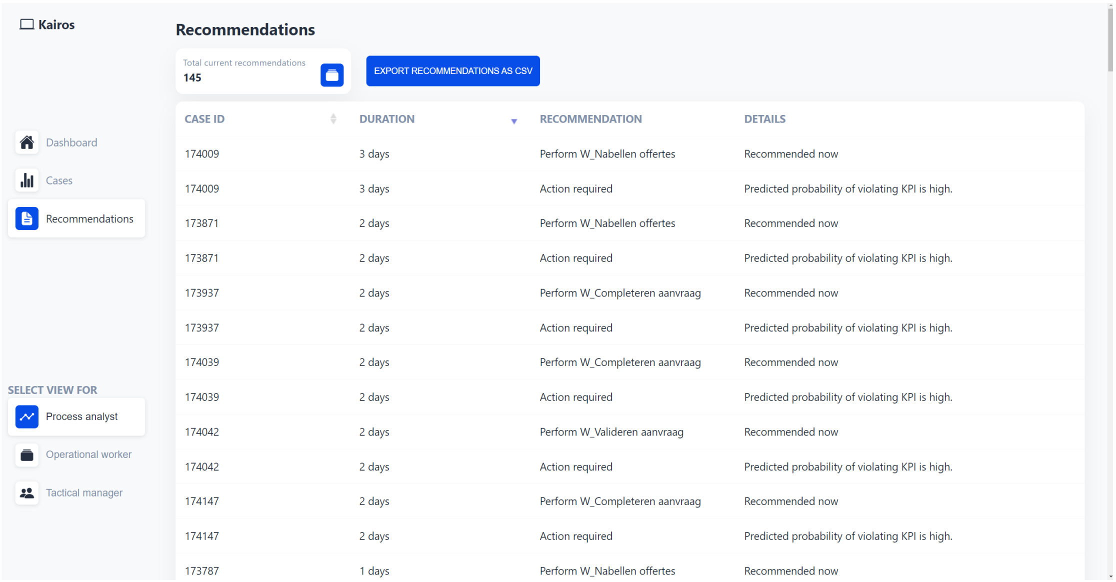


Figure 14. Recommendations view

- *Performance* (In this case *Duration*): Presents the case performance, which is calculated using the KPI defined by the user. Since, in this case, the user chose a positive case outcome to be a "duration of less than or equal to 8 days", we calculate the duration of each case to compare to the defined KPI, which enables us to determine whether the case has a positive outcome or not.
- *Recommendation*: What the recommendation suggests.
- *Details*: Additional details associated with a recommendation.

The view allows the user to export the recommendations as a .csv file if necessary.

6 Evaluation

In this section, we describe the testing process of the application, the limitations of the solution and improvement opportunities for future implementations.

6.1 Evaluation goals

The evaluation goals stem from our research goal of implementing a useful and user-friendly tool for visualising PrPM output. From this, we can isolate the code words, which will be used as a qualitative scale for analysing the results. In particular, we note the words: *"useful"* and *"user-friendly"* and define our goals as follows:

- **G1. Usability**: The application is user-friendly for navigating, accessing data, viewing information and most importantly exploring recommendations.
- **G2. Usefulness**: The application is useful for process analysts for inspecting cases overview, determining case performance and positive outcomes compared with a selected KPI, seeing new and past recommendations and their metrics for each individual case, utilizing a process model for visualising them within a process, understanding and prioritizing prescribed interventions and identifying bottlenecks for process improvement.

Additionally, we define the themes for analysing the transcribed interviewers. For this, we consider issues and improvement opportunities, as these two themes emerge repeatedly during the evaluation process. Finally, we define the following evaluation categories:

- **T1. Usefulness**: This theme consolidates all the comments made about the usefulness of the interface. In particular, this includes the comments about if and how the process analyst would find the information present in the interface useful.
- **T2. Usability**: This theme consolidates all the comments made about the ease of navigating the application, including the colours, legends, guides, information tooltips, etc.
- **T3. Issues**: This theme covers all the negative observations made by the participants, including comments about the layout, content and information displayed in the interface.

- **T4. Suggestions**: This theme describes the suggestions for improving the interface by adding, removing or changing the content or structure.

6.2 Results of the evaluation

In this section, we analyse the interviews presented in section 3.4 and discuss their implications.

6.2.1 User interviews

The results of the evaluation show that the process analysts find the overall layout of the application user-friendly: they are able to determine what the recommendations are, how the application is structured, how to navigate it to different views and perspectives: *"There's a logical use of space with sort of filters or selections on the side. And on top. Logical sort of going back buttons. Return button here."* (I2), *"I [...] like the layout, how here you can go [to] case, recommendations and then also from different roles perspective, because, of course, a process analyst [is] not the same as a tactical manager or operational worker."* (I3). The process model is mentioned to clearly depict the different components of a trace: *"[...] really liked working with the process model because from the process model, [you can] clearly see the recommendations, the interventions, where we currently [are] in the ongoing application."* (I1). However, the participants demonstrate issues with identifying different recommendation types and understanding how they are calculated, even after consulting the information tooltip next to the recommendations list (figure 12), which is rarely detected unless pointed out by the interviewer. Moreover, some participants suggest adding a filtering mechanism for the tables (figures 10, 14), so that the user is able to filter on a case, or the presence of recommendations and their types: *"I think a filter for all the columns would be helpful [...] if I just want to filter on one case."* (I3).

The participants note that KPI and the case performance presented in the individual case view are useful for determining whether the case satisfies the positive case outcome condition: *"Of course, the parts of the dashboard at the top, like the KPIs [and] case performance [are] very useful."* (I3) and *"this information is also useful, which is the KPI which is the case performance, just to make sure [...] if this case satisfied this KPI"* (I1). Additionally, the case details (attributes) displayed along with the case performance are noted to be a good addition: *"Basic information about the case. I also think it's an important thing"* (I1). In the view for displaying a list of cases (figure 10, the columns for *Duration* and *Intervened* allow the process analyst to speculate about the positive outcome of the case *"[this] information [displaying] all the current duration is useful. [...] intervened or not is also very useful [for] a positive outcome."* (I1).

While the participants of our study find it useful to have the recommendations present in each individual case view, they also note that the metrics presented with the calculations do not provide a sufficient level of insight for a process analyst (*"process analyst [...] are not necessarily also data scientist [...] or statistically educated. So this wouldn't say anything to them."* I2), but would also not be suitable for justifying improvement opportunity suggestions to the client or a process owner, thus, rendering them mostly useless.

Finally, the participants express the need to have an insight into the inner working of the algorithms, and how exactly the prescribed recommendations might affect the case in the future. Comments such as *"I would be more interested in how these recommendations [and] algorithms work."* (I2) or *"how are you predicting the next activity? [...] I [want to] have more explanation on why I should believe that whatever you predicted, could be the right answer."* (I3) and *"I think numbers [are more useful] than the algorithm itself: [...] more details [on] calculations or numbers that would validate the answer"* (I3) solidify the fact that the process analysts expect more justifications for choosing recommendations, as well as more explanation for the metrics already displayed in the interface. The process analysts also suggest displaying the precise impact of the recommendation on the case, rather than vague suggestions about a recommendation helping in achieving a positive case outcome: *"if you recommended an activity, [...] what analysts or client like to see [...] is how does that affect my [...] agenda process, [...] whether it's preventing some deadlocks, or preventing some rework or like loops? [...] I would like to see a ROI to present to leadership to be able to convince them: this is how you make more money"* (I3).

6.2.2 User survey

Following the demonstration and the interview, the participants are given a questionnaire to describe their experience with Kairos. The questions are constructed using the System Usability Scale [2] for ease of use and the 5-point Likert scale [3] for the perceived usefulness [1] of the system. The survey includes demographic questions for estimating the domain of operation and experience of the interviewee with machine learning, process mining and PrPM.

The answers to the questionnaire are used to calculate usefulness and usability scores [2] for each interviewee. The usefulness scores are rescaled, to standardize

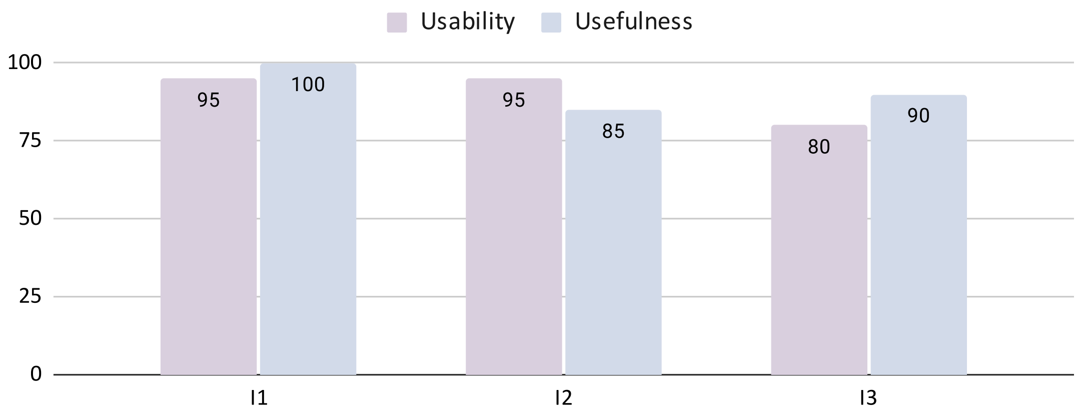


Figure 15. Usability and usefulness scores of Kairos according to the interviewees

the results and displayed in figure 15 along with the usability scores. According to the findings, the interviewees find the interface easy to use without significant input from a technical person to guide the usage or having to learn a lot of additional rules. The application is not unnecessarily complex or too cumbersome to navigate, which allows the user to learn the visualisation tool quickly and feel confident using it.

The high scores in usefulness also suggest that the interviewees would find Kairos useful at work, making their job easier and increasing productivity. The participants find that the application would help them complete their tasks somewhat faster by aiding them in identifying improvement opportunities in the process.

ID	Description
F1	Process analysts need to understand the motivation behind recommendations' calculation
F2	Process analysts need to see a tangible impact of a recommendation on the case
F3	Process analysts need to know that the prescriptive algorithms are reliable

Table 7. Evaluation analysis findings summary

Finally, we consolidate the evaluation analysis by summarizing the findings in table 7, wherein lie the potential improvements for the application.

6.3 Discussion

In this section, we discuss the results of the evaluation in conjunction with the goals of the research and possible improvements.

In light of our goal to implement a useful and user-friendly application that visualises PrPM output, we conduct user evaluation. The view of the case performance coupled with the recommendations displayed in a process model is proved to be useful according to the participants of our study, which is in line with the results presented by Kubrak et al. [27], who argue that the interface should capture the process and its performance in tandem to provide a holistic view of improvement opportunities.

While the findings suggest that the participants of our study find the application largely satisfies the evaluation goals we set, further improvements can be made towards assisting process analysts in evaluating and selecting recommendations from visual representation, which requires them to understand two main things: (1) the logic behind recommendation calculation, and (2) the benefits of selecting a recommendation.

It is evident that process analysts strive to understand the reasoning behind a recommendation when reviewing a process. By seeking knowledge about the origin of the recommendation, the process analysts establish trust in the application and gain insight into what the recommendation actually prescribes. According to the findings of the user interviews, the interface does not offer sufficient information to understand the prescribed interventions and the motivations behind them. This could be rectified with additional informational tooltips and user guides.

The findings also indicate that process analysts deem it important to understand the potential benefits of a recommendation. The motivation for this is twofold: they seek to understand what tangible effect a recommendation has on a process instance, as well as the process itself; and how the recommendations can be prioritized depending on their potential impact. While this is partially addressed with the selection of a KPI of interest and, later, calculation of case performance/outcome in relation to the KPI, there are no concrete claims made about the tangible benefits of a recommendation to the outcome of a single process instance or the overall process. This could be addressed by displaying information about the past occurrences of the recommendation in other similar process instances and their outcomes. This is in line with the research conducted by Huber et al. [13], who implement a variation of this solution by incorporating the reasoning behind each recommendation into their prototype. The grounds for a recommendation can be the shortening of cycle time, mitigation of a risky execution, supporting case goals, or precedent [13].

6.3.1 Limitations

We identify limitations in this study to help ensure the accuracy and reliability of our findings and facilitate the advancement of the presented solution.

In order to achieve the goal of implementing a useful and user-friendly application that visualises PrPM output, we followed the design science approach, which may inherently pose some limitations. We may conjecture that picking other approaches to designing, developing and evaluating this application could result in a more effective solution [11].

The artefact was implemented according to the requirements elicited in the *Defining objectives* part of the research, on the basis of past research into the topic. There exists a risk that some design objectives were not elicited or prioritized incorrectly due to bias or subjectivity, or that they were implemented in a sub-optimal manner. To minimize this risk, the requirements and their implementation were discussed at the meetings after development iterations; the feedback was taken into account for improvements to be made in the following iterations.

Another possible limitation of the study is the insufficient amount of interviews for reaching data saturation in user evaluation, which could impact the quality of the conducted research [17]. The presence of bias and subjectivity in this qualitative research poses a threat to the integrity of the evaluation. The threat was alleviated by collaborating on the user study with the research team and discussing the findings to eliminate biases, but the exact effects of these limitations remain to be determined.

7 Conclusion

This thesis aimed to develop software that visualises PrPM outputs in a useful and user-friendly way. The application was created considering the output of a PrPM tool - PrCore [41], and design objectives set forth by previous research into visualising PrPM output [38]. The goal of this research was achieved in three stages: eliciting requirements, developing the application and conducting an evaluation. The resulting software - Kairos - was presented in section 5.

In order to capture the needs of the users, we defined requirements for the Kairos interface using previous research into the field [38]. There were a total of 39 requirements specified after analyzing the baseline. They were then prioritized using the MoSCoW method, which highlighted the requirements crucial for delivering the core functionality of the application. The requirements were then implemented using the visualisation principles and a common structure of frontend-backend-database used for developing web applications. The format was used as the previous research designs wireframes in a form of a website [38]. The application was integrated with a PrPM tool to receive prescriptive output, which was recorded and used for the construction of visualisations. We then conducted an evaluation of Kairos using a user study, where three practising process analysts participated by interacting with the application and filling out a questionnaire for determining the ease of use and usefulness of the visualisation tool. We found that while Kairos satisfied the aforesaid conditions, there were some weaknesses that needed to be addressed.

Thus, we present potential opportunities for improvement. First, the visualisation application clearly states the logic behind the algorithms and recommendations by implementing a guide for a new user. The goal of this would be to eliminate any confusion about the prescriptive algorithm types and ensure that the process analyst understands the motivation behind the recommendations. The second improvement could be to offer more tangible insight into the impact of a recommendation on a case, in the form of a tangible measure, or past occurrences of the recommendation in similar business process instances. This would offer a more justified reason for prioritizing one recommendation over another or justifying the need for improvement opportunity of an entire process to a client.

Acknowledgements

We are thankful to all the interview participants for finding time to provide valuable insights during the user evaluation stages of this research.

References

[1] Fred D Davis. “Perceived usefulness, perceived ease of use, and user acceptance of information technology”. In: *MIS quarterly* (1989), pp. 319–340.

[2] John Brooke. “SUS: A quick and dirty usability scale”. In: *Usability Eval. Ind.* 189 (Nov. 1995).

[3] Anol Bhattacherjee. “Understanding Information Systems Continuance: An Expectation-Confirmation Model”. In: *MIS Quarterly* 25.3 (2001), pp. 351–370. ISSN: 02767783. URL: http://www.jstor.org/stable/3250921 (visited on 05/02/2023).

[4] Ben Shneiderman. “The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations”. In: *IEEE Symposium on Visual Languages, Proceedings* (2003).

[5] Patrik Berander and Anneliese Andrews. “Requirements Prioritization”. In: *Engineering and Managing Software Requirements*. Ed. by Aybüke Aurum and Claes Wohlin. "Berlin, Heidelberg": Springer Berlin Heidelberg, 2005, "69–94". ISBN: "978-3-540-28244-0". DOI: "10.1007/3-540-28244-0_4". URL: https://doi.org/10.1007/3-540-28244-0_4.

[6] Virginia Braun and Victoria Clarke. “Using thematic analysis in psychology”. In: *Qualitative Research in Psychology* 3 (Jan. 2006), pp. 77–101. DOI: 10.1191/1478088706qp063oa.

[7] Ken Peffers et al. “A design science research methodology for information systems research”. In: *Journal of Management Information Systems* 24 (Jan. 2007), pp. 45–77.

[8] Helen Schonenberg et al. “Supporting Flexible Processes through Recommendations Based on History”. In: *Business Process Management*. Ed. by Marlon Dumas, Manfred Reichert, and Ming-Chien Shan. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 51–66. ISBN: 978-3-540-85758-7.

[9] Wil Aalst et al. “Process Mining Manifesto”. In: vol. 99. Aug. 2011, pp. 169–194. ISBN: 978-3-642-28107-5. DOI: 10.1007/978-3-642-28108-2_19.

[10] Christian Haisjackl and Barbara Weber. “User Assistance during Process Execution - An Experimental Evaluation of Recommendation Strategies”. In: *Business Process Management Workshops*. Ed. by Michael zur Muehlen and Jianwen Su. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 134–145. ISBN: 978-3-642-20511-8.

[11] Shirley Gregor and Alan Hevner. "Positioning and Presenting Design Science Research for Maximum Impact". In: *MIS Quarterly* 37 (June 2013), pp. 337–356. DOI: 10.25300/MISQ/2013/37.2.01.

[12] Tamara Munzner. *Visualization Analysis and Design*. A K Peters/CRC Press, 2014. ISBN: 9780429088902.

[13] Sebastian Huber, Marian Fietta, and Sebastian Hof. "Next Step Recommendation and Prediction Based on Process Mining in Adaptive Case Management". In: *Proceedings of the 7th International Conference on Subject-Oriented Business Process Management*. S-BPM ONE '15. Kiel, Germany: Association for Computing Machinery, 2015. ISBN: 9781450333122. DOI: 10.1145/2723839.2723842. URL: https://doi.org/10.1145/2723839.2723842.

[14] Wil M. P. Van Der Aalst, Marcello La Rosa, and Flávia Maria Santoro. "Business Process Management". In: *Business & Information Systems Engineering* 58 (2016), pp. 1–6. DOI: 10.1007/s12599-015-0409-x. URL: https://doi.org/10.1007/s12599-015-0409-x.

[15] Marlon Dumas et al. *Fundamentals of Business Process Management*. Springer, Berlin, Heidelberg, 2018. ISBN: 9783662565087.

[16] Amjad Hudaib et al. "Requirements Prioritization Techniques Comparison". In: *Modern Applied Science* 12 (Jan. 2018). DOI: 10.5539/mas.v12n2p62.

[17] Benjamin Saunders et al. "Saturation in qualitative research: exploring its conceptualization and operationalization". In: *Quality & Quantity* 52 (Jan. 2018), pp. 1893–1907. DOI: 10.1007/s11135-017-0574-8. URL: https://doi.org/10.1007/s11135-017-0574-8.

[18] Alexander Seeliger, Timo Nolle, and M. Mühlhäuser. "ProcessExplorer: An Interactive Visual Recommendation System for Process Mining". In: 2018.

[19] Marit Sirgmets et al. "Designing Process Diagrams – A Framework for Making Design Choices When Visualizing Process Mining Outputs". In: *Confederated International Conferences: CoopIS, C&TC, and ODBASE*. Oct. 2018, pp. 463–480. ISBN: 978-3-030-02609-7. DOI: 10.1007/978-3-030-02610-3_26.

[20] Adriano Augusto et al. "Automated Discovery of Process Models from Event Logs: Review and Benchmark". In: *IEEE Transactions on Knowledge and Data Engineering* 31.4 (2019), pp. 686–705. DOI: 10.1109/TKDE.2018.2841877.

[21] Marcus Dees et al. "What if Process Predictions are not followed by Good Recommendations?" In: *17th International Conference on Business Process Management 2019 Industry Forum*. 2019, pp. 61–72.

[22] Vinicius Stein Dani, Carla Maria Dal Sasso Freitas, and Lucinéia Heloisa Thom. "Ten years of visualization of business process models: A systematic literature review". In: *Computer Standards & Interfaces* 66 (2019), p. 103347. ISSN: 0920-5489. DOI: https://doi.org/10.1016/j.csi.2019.04.006. URL: https://www.sciencedirect.com/science/article/pii/S0920548918303295.

[23] Massimiliano de Leoni, Marcus Dees, and Laurens Reulink. "Design and Evaluation of a Process-aware Recommender System based on Prescriptive Analytics". In: *2020 2nd International Conference on Process Mining (ICPM)*. 2020, pp. 9–16. DOI: 10.1109/ICPM49681.2020.00013.

[24] Mansoureh Yari Eili, Jalal Rezaeenour, and Mohammadreza Fani Sani. "A Systematic Literature Review on Process-Aware Recommender Systems". In: *CoRR* abs/2103.16654 (2021). arXiv: 2103.16654. URL: https://arxiv.org/abs/2103.16654.

[25] Kateryna Kubrak et al. "Prescriptive process monitoring: *Quo vadis*?" In: *PeerJ Comput. Sci.* 8 (2022), e1097.

[26] Williams Rizzi et al. *Nirdizati: an Advanced Predictive Process Monitoring Toolkit*. 2022. arXiv: 2210.09688 [cs.AI].

[27] Kateryna Kubrak, Fredrik Milani, and Alexander Nolte. "A visual approach to support process analysts in working with process improvement opportunities". In: *Business Process Management Journal* 29 (Apr. 2023), pp. 101–132. DOI: 10.1108/BPMJ-10-2021-0631.

[28] *Cytoscape.js*. URL: https://js.cytoscape.org/.

[29] *Docker docs*. URL: https://docs.docker.com/.

[30] *Flask*. URL: https://flask.palletsprojects.com/en/2.3.x/.

[31] *Github Docs*. URL: https://docs.github.com/en.

[32] *Goodbye manual notes. Hello OtterPilot*. URL: https://otter.ai/.

[33] *Great Writing, Simplified*. URL: https://www.grammarly.com/.

[34] *Gunicorn - WSGI server*. URL: https://docs.gunicorn.org/en/stable/.

[35] *Introducing ChatGPT*. URL: https://openai.com/blog/chatgpt.

[36] *Introduction*. URL: https://vuejs.org/guide/introduction.html.

[37] *Jira Documentation*. URL: https://confluence.atlassian.com/jira.

[38] Kateryna Kubrak et al. "Design and Evaluation of a User Interface Concept for Prescriptive Process Monitoring".

[39] *nginx documentation*. URL: https://nginx.org/en/docs/.

[40] Zhaosi Qu. *Getting Started*. URL: https://prcore-docs.chaos.run/getting-started/.

[41] Zhaosi Qu. *PrCore*. URL: https://github.com/prcore.

[42] *What is MongoDB Atlas?* URL: https://www.mongodb.com/docs/atlas/.

Appendix

I. PrCore API

Endpoint	Method	Request body	Description
/event_log	POST	json	Upload event log
/event_log/event_log_id	PUT	json	Columns configuration
/project	POST	json	Outcome and treatment definition, creating a project
/project/project_id	GET	none	Get project details
/project/project_id	DELETE	none	Delete the project
/project/project_id /stream/start	PUT	none	Start simulating a project
/project/project_id /stream/stop	PUT	none	Stop simulating a project
/project/project_id /stream/clear	PUT	none	Remove all streamed data and results
/project/project_id /re-sult/result_key	GET	none	Get static results of a dataset
/project/project_id /stream/result	GET	none	Get event stream of the prescriptions

Table 8. PrCore API endpoints

II. User survey

1. Please rate the following statements about the usability of Kairos on a scale from 1 (strongly disagree) to 5 (strongly agree).
   (a) I think that I would like to use Kairos frequently.
   (b) I found the tool unnecessarily complex.
   (c) I thought the tool was easy to use.
   (d) I think that I would need the support of a technical person to be able to use this tool.
   (e) I found the various functions in this tool were well integrated.
   (f) I thought there was too much inconsistency in this tool.
   (g) I found the various functions in this tool were well integrated.
   (h) I found the tool very cumbersome to use.
   (i) I felt very confident using the tool.
   (j) I needed to learn a lot of things before I could get going with this tool.
2. Please rate the following statements about the usefulness of Kairos on a scale from 1 (strongly disagree) to 5 (strongly agree)
   (a) Using this tool at work would help me complete tasks faster.
   (b) Using Kairos would aid me in identifying improvement opportunities in the process.
   (c) Using this tool would increase my productivity.
   (d) Using this tool would make it easier to do my job.
   (e) I would find this tool useful at work.

III. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Lana Botchorishvili**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,
   Visualising PrPM Output,
   supervised by Fredrik Payman Milani, Alexander Udo Nolte and Kateryna Kubrak.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Lana Botchorishvili

09/05/2023