Nirdizati: A Web-based Tool for Predictive Process Monitoring

Andrii Rozumnyi[1], Chiara Di Francescomarino[2], Fabrizio Maria Maggi[1],
Ilya Verenich[3,1], Kerwin Jorbina[1], Marcello La Rosa[3],
Marlon Dumas[1], and Simon Raboczi[3]*

[1] University of Tartu, Estonia
{andriiro,f.m.maggi,marlon.dumas}@ut.ee,kerwin.jorbina@gmail.com
[2] FBK IRST, Trento, Italy
dfmchiara@fbk.eu
[3] Queensland University of Technology, Australia
{ilya.verenich,m.larosa,simon.raboczi}@qut.edu.au

Abstract. In this paper, we present Nirdizati, an open-source tool for predictive monitoring of business processes. The tool can be used to predict the outcome, the next events, or the remaining time of each case of a process. For example, in an lead-to-order process, Nirdizati can predict which customer leads will convert to purchase orders and when. In a claims handling process, Nirdizati can predict if a claim decision will be made on time or late. Based on these predictions, process workers and operational managers can act proactively to resolve or mitigate potential process performance violations. The target audience of this demonstration includes process mining researchers as well as practitioners interested in exploring the potential of process monitoring.

Keywords: Process Mining, Predictive Process Monitoring, Machine Learning

1 Introduction

Nirdizati consists of two components: Nirdizati Training and Nirdizati Runtime (Figure 1). Nirdizati Training takes as input a business process event log and produces a prediction model. This model can then be deployed in Nirdizati Runtime. Once a model is deployed, Nirdizati Runtime listens to a stream of events related to a business process, and produces a stream of predictions. These predictions are then visualized in a continuously updated Web dashboard.

2 Nirdizati Training

First, we have the Front-end application which acts as the interface for the user to select settings used for prediction and to analyze the prediction results. The
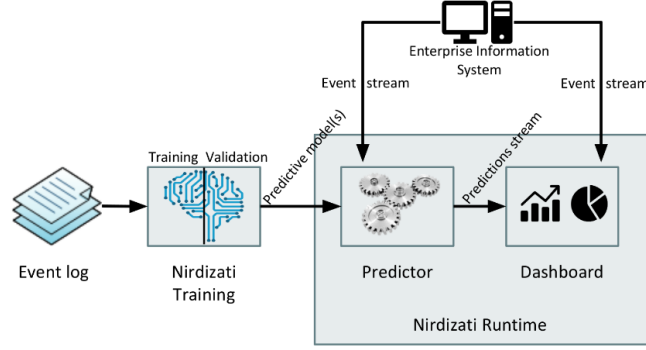
---

* Author names are in alphabetical order

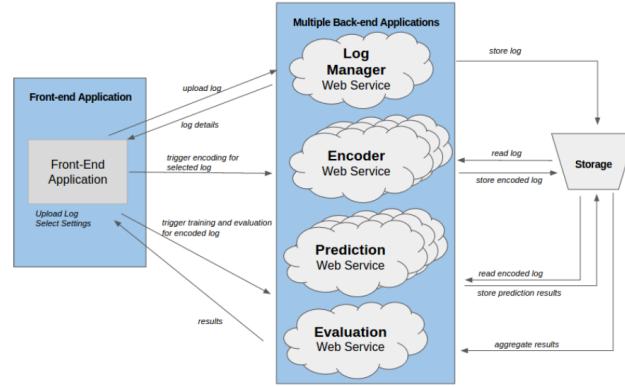**Fig. 1.** A high-level overview of Nirdizati.

Fig. 2. A high-level overview of Nirdizati Training.

second module is the Log Manager which is responsible for managing the logs. Uploading and retrieving the logs are the basic operations of this module. The third module is the Encoder which retrieves the log from the storage, parses it and prepares the log for the training phase in the Predictive Module. In this part, we retrieve the encoded data, split it into training and test set for evaluation, and build the predictive model from the training data. Finally, using the test data, we test it against the model created to get its accuracy. The aggregation of the results are done in the Evaluation Module which is tasked to calculate the error of the prediction model created.

3   Nirdizati Runtime

Figure 3 illustrates the flow of data through the proposed system. A workflow management system (WFMS) continuously registers tasks performed in the organization. This serves as an input to our system, in the form of a stream of

events. Next, the produced stream is consumed by a web server which is responsible for storing data and further information processing. Web server, in turn, communicates with a predictive engine which applies pre-trained models for a business process and returns all the results of calculations. After web server receives the outputs from each predictive model, it writes them to a database and updates corresponding clients. From the user's side, the results are visualized via a dashboard-based interface which is capable of sending queries requesting various visualization options.
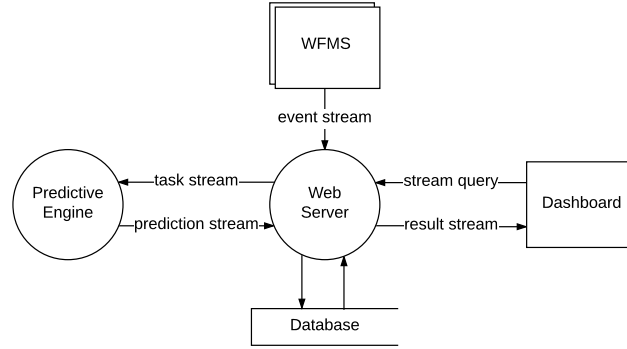
Fig. 3. High-level data flow diagram in a Yourdon-Coad notation

Based on this data flow diagram, we devise the main parts and high-level components of the system (Figure 4):

- streaming platform
- web server with load balancer and internal components
- predictive engine
- client-side user interfaces

Process workers and operational managers typical users of such system can set the process performance targets and subscribe to a stream of warnings and alerts generated whenever these targets are predicted to be violated. Thus, process workers will be capable of making more informed, data-driven decisions to get a better control of the process execution. This is especially beneficial for processes where process workers have more leeway to make corrective actions, for example, in a lead management process.

4   Conclusion

The developed solution, named *Nirdizati*, is a configurable full-stack web application that supports users in selecting the preferred prediction method from the
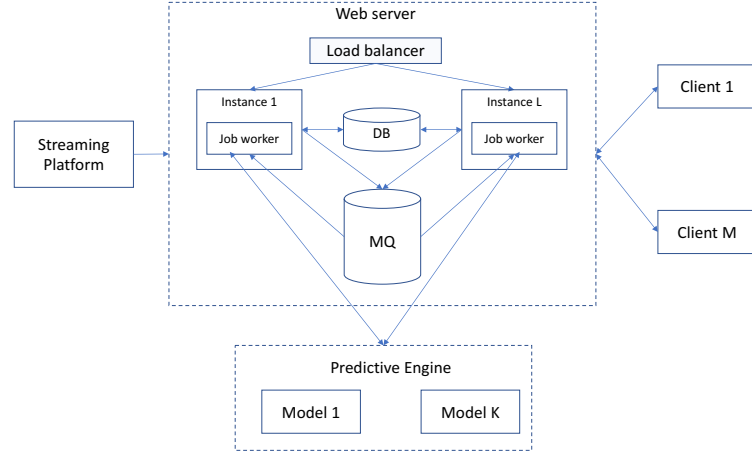
Fig. 4. Proposed system design architecture

list of implemented methods and enables the continuous prediction of various performance indicators at runtime. The results of the predictions, as well as the real-time summary statistics about the process execution, are presented in a dashboard that offers multiple visualization options.

The source code has been released as open source software under the Lesser GNU Public License (L-GPL) at `http://github.com/nirdizati`. The developed engine has been deployed on the server belonging to the Institute of Computer Science and can be accessed at `http://nirdizati.com`.