preCEP: Facilitating Predictive Event-Driven Process Analytics

Bernd Schwegmann[1], Martin Matzner[1], and Christian Janiesch[2]

[1] University of Münster, ERCIS, Muenster, Germany
bernd.schwegmann@uni-muenster.de,
martin.matzner@ercis.uni-muenster.de
[2] Karlsruhe Institute of Technology, AIFB, Karlsruhe, Germany
christian.janiesch@kit.edu

Abstract. The earlier critical decision can be made, the more business value can be retained or even earned. The goal of this research is to reduce a decision maker's action distance to the observation of critical events. We report on the development of the software tool preCEP that facilitates predictive event-driven process analytics (edPA). The tool enriches business activity monitoring with prediction capabilities. It is implemented by using complex event processing technology (CEP). The prediction component is trained with event log data of completed process instances. The knowledge obtained from this training, combined with event data of running process instances, allows for making predictions at intermediate execution stages on a currently running process instance's future behavior and on process metrics. preCEP comprises a learning component, a run-time environment as well as a modeling environment, and a visualization component of the predictions.

Keywords: Event-driven Process Analytics, Business Activity Monitoring, Complex Event Processing, Business Process Management, Operational Business Intelligence.

1 Introduction

Business value can be lost if a decision maker's action distance to the observation of a business event is too high. Action distance comprises the time needed to capture and process data as well as the time needed to decide which action to take. So far, two classes of information systems, which promise to assist decision makers, have only been discussed independently of each other: business intelligence systems, which query historic business event data in order to prepare predictions of future process behavior, and real-time monitoring systems such as business activity monitoring (BAM). We developed a method and implemented software which allows using real-time data for predictions in an event-driven approach. Our predictive event-driven process analytics (edPA) software preCEP brings together features of BAM and process intelligence with the intention to reduce action distance. In Section 2, we elaborate on the design of the prototypical artifact. Section 3 discusses the software's

J. vom Brocke et al. (Eds.): DESRIST 2013, LNCS 7939, pp. 448–455, 2013.

significance to research and practice. Section 4 discusses one of the steps undertaken to evaluate the usefulness of the tool.

2 Design of the Artifact

2.1 Problem Statement

The development of preCEP is motivated by two questions which rose while attempting to reduce action distance with current BAM approaches:

Question 1: "How can the BAM approach be further developed so that predictions of process instance behavior and estimations of process metrics are supported?"

We identified event-driven architecture to be a promising paradigm for exploiting event log data from historical process instances. By now, BAM systems report on the current situation of a process context only and therefore analyze events from running instances. In order to add prediction capability to BAM, it also needs to analyze event data from completed process instances so that it can learn about past process behavior (e.g., by using data mining models on historic data). However, data mining requires large data sets to form accurate prediction models and extensive data preparation is needed (such as a uniform format, variable selection, substitution techniques, etc.).

Question 2: "How can the dynamically increasing number of attributes of running instances be handled?" Data mining models typically require a set of predictors with a fixed length. In contrast, running processes dynamically generate new data.

Against this backdrop, we saw that there is currently no solution available for predictive edPA yet. Such a solution would include a *prediction runtime*, which analyzes runtime behavior of process instances, a *modeling component*, i.e. software support to specify the analytical approach to predicting process behavior and process metrics, and a *visualization component* which informs decision makers about predicted behavior and/ or process metrics.

2.2 Approach

In response to question 1, we developed a conceptual architecture for predictive edPA (A1) which uses CEP technology. In response to question 2, we developed a modeling approach for predictive edPA (A2). We chose to implement the architecture with standards of the shelf technology (S1) and developed a modeling environment for predictive edPA (S2).

Architecture (A1): The predictive edPA architecture comprises a CEP engine and a prediction runtime (cf. Fig. 1). Input adapters let the CEP engine tap into event data from persistent stores (historical events), events currently produced by operational systems (current events) or events with predicted values from the prediction runtime (predicted events). The CEP engine executes an event processing network (EPN) which transforms events into meaningful process or instance level key performance

indicators (KPI) to assist detecting, analyzing, diagnosing, and resolving critical process behavior. The components are loosely coupled by publish and subscribe mechanisms for push-based exchange of the latest KPI or predictions. The prediction runtime can further pull KPI from historical events.

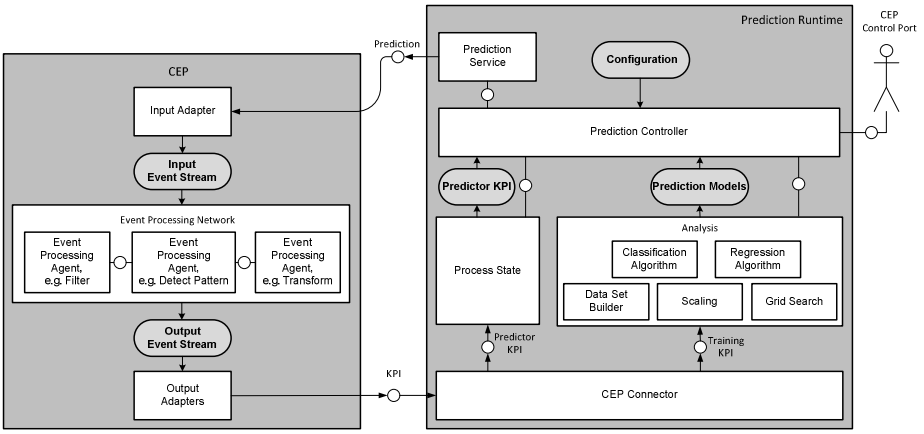

Fig. 1. preCEP architecture for integrating predictive analytics and CEP [1]

The prediction runtime is an event consumer of the CEP engine as it trains a prediction model with historic event data and as it makes predictions with current events. The CEP connector and the prediction service handle the cross-system event exchange while the prediction controller orchestrates the subcomponents internally. The analysis component calls a data set builder to link training KPI available in each execution stage with a response KPI. An execution stage holds a set of KPI available for a process at one point during its execution while a response stage holds a label for a set of training KPI in an execution stage. Both are modeled in the EPN and are accessible by the prediction runtime. The prepared data sets are scaled and a grid search for the best parameters of the algorithm compares measurements on the prediction quality of each model. Possible algorithms include support vector machines (SVM), decision trees and rule based techniques for classification and prediction of numeric values [2]. The classification accuracy and the mean square error (MSE) for numeric predictions are used as measures of the predictive power [3]. The prediction models are then stored for each execution stage. In case the prediction runtime consumes KPI from current events the process state component looks up the execution stage corresponding to the set of predictor KPI and executes the related prediction model with these values.

The prediction runtime also acts as event producer when making a prediction. Since the prediction is fed back into the CEP system, it is available for further processing such as providing context for the event, passing the prediction event to a dashboard or triggering an automated process improvement in a BPMS.

Modeling approach (A2): We organize metrics in execution stages to better handle the dynamically increasing number of KPI for running process instances – an idea that is adopted from process intelligence and process mining [2, 4–6]. In the EPN, events are processed by event processing agents (EPA) such as detect, filter, and transform. To find causalities or to derive complex events, EPAs may also relate single events to other events [7, 8]. We propose to model KPI through EPN (cf. Fig. 2). After events have been observed, they are transformed to numeric or categorical metrics which are related to an execution stage. Execution stages can be related to single activities, groups of activities or entire sub processes of the control flow structure. Next, the KPI are transferred from stage to stage so that at the end of the execution, the maximum data base is available. Execution stages refer to activities of a conceptual workflow model. Therefore, they may encounter abundance (e.g., loops) or incompleteness (e.g., XOR split) of related events. Such situations should be addressed by basic workflow patterns such as parallel split, synchronization, exclusive choice, simple merge, and iteration by means of data validation techniques such as elimination, inspection, identification, and substitution of incomplete records [9]. Only the KPI in the execution stages are needed to finally make a prediction. However, in order to train a prediction model, a dependent variable is needed which is made available in the response stage. All execution stages of an EPN can be related to the same response stage. It is also possible to assign them to different response stages. This enables more than one and also different predictions for each execution stage.

In contrast to instance metrics, process metrics constitute time series of values with the same general structure for each metric. Therefore, we propose to use the sliding input window feature of the EPA in order to define a time series of fixed length. This information is then used as input for process behavior prediction.

Implementation of the Architecture (S1): The presented architecture has been implemented on top of an internal release of a mayor software vendor's BPMS which also integrates CEP functionality [10]. We deliberately refrain from a detailed discussion of the flowchart of the prediction runtime and point only to some techniques that have not been described in the general architecture description so far.

In push based execution, the process state component receives predictor KPI always from the latest execution stage. In pull based execution, the execution stage is determined by first checking whether there is already a record in an execution stage for the instance of interest. Second, the process state component finds the latest predictor KPI by comparing all relevant execution stages timestamps. Independent of that, the prediction model is then applied for the predictor KPI received from the execution stage.

The analysis component of the prototype utilizes LIBSVM. LIBSVM is a popular library for support vector classification and support vector regression [11] which recommends scaling of the input data [12]. Therefore, all attributes are scaled to the range of -1 to 1. On the prepared data set, a grid search is conducted in order to find the optimal parameters for the SVM.

Modeling environment (S2): We built our modeling environment upon an existing design component for edPA that is an Eclipse based editor. It defines monitoring

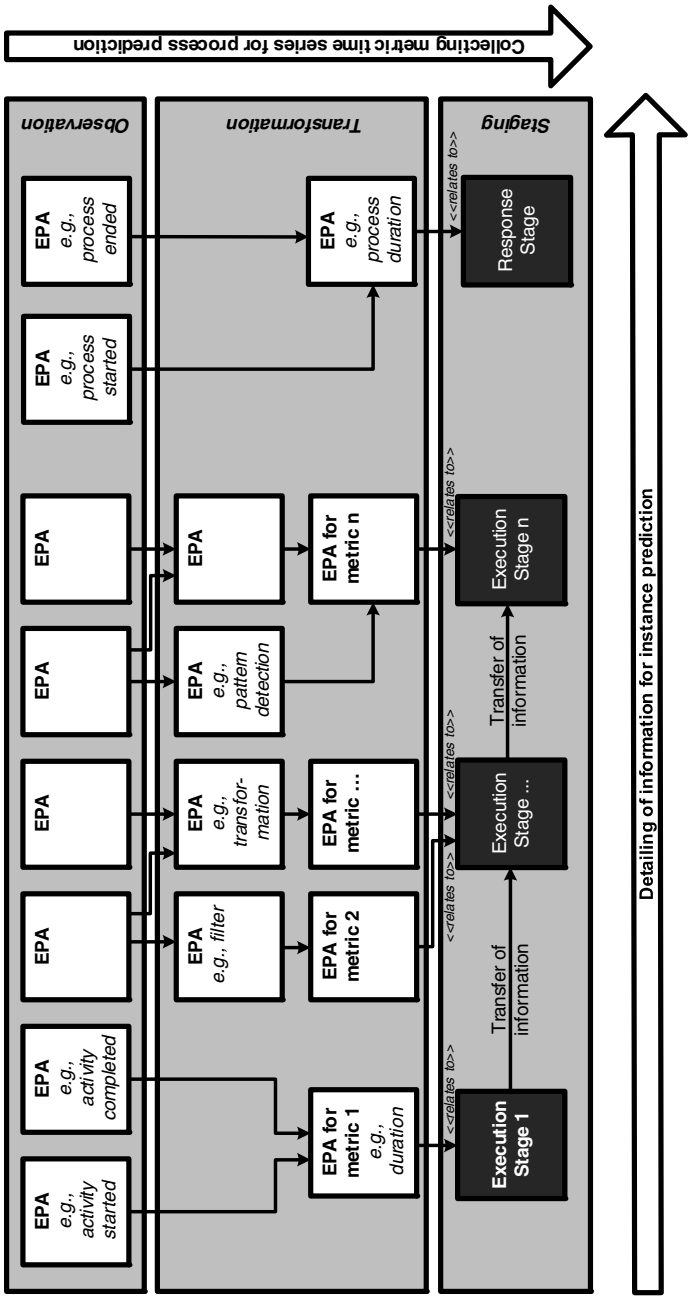

Fig. 2. Modeling predictor and training KPI in the EPN

models for conceptual workflow models. So far, its EPAs have not been used as representations for execution or response stages. Therefore, we introduce a new property section where the modeler can select the representation type and the prediction algorithm. This design information is stored in Java Script Object Notation (JSON) format.

Data mining procedures require extensive data preparation including data exploration and significance tests [13]. A design tool for predictive edPA should support this analysis by uploading historical event data to a preliminary EPN in order to investigate the KPI in the stages by statistical means. Due to limitations of our prototype this component is not implemented yet but could easily be added by using data mining libraries such as JDMP [14] or WEKA [15].

A user interface completes the prototype. Prediction results are either presented in the generic task user interface or as process visualization (along with the process model).

2.3 Use cases

The preCEP is intended to help decision makers in making better decisions faster in order to improve the operational performance of business processes. Examples include predictive maintenance of machinery, fraud detection in financials as well as route scheduling and optimization in logistics.

3 Significance of the Artifact

Significance for research: preCEP combines two areas of process analytics which so far had been studied separately only. Our software and the implemented method both outline, e.g., which data needs to be shared among monitoring components and prediction components in order to facilitate predictive edPA. Also with our approach, we offer a new perspective on how process-aware information systems can work together with analytical systems in near real-time. Future studies might directly take up and improve the IT artifacts suggested in this paper or test their application in enterprise contexts. They might also investigate problems, which occurred while designing and implementing our solution, such as the lack of a uniform event format for historical analysis as well as for the analysis of current events.

Significance for practice: The software prototype demonstrates how analytical systems and process aware IS can be coupled loosely via a CEP engine. This architecture allows for integrating several concrete software systems of each of the three system types. Design decisions such as the dual use of an EPN for historical and current events are made available to practitioners. Further, preCEP indicates options for refined development as regards, e.g., data exploration and visualization. The prototype not only adds to decision making on the operational and tactical level of an organization by providing predictions to business users but also allows for automated decision

making in the CEP engine in a closed loop approach. Both ways, the tool supports proactive process performance management.

4 Evaluation of the Artifact

We analyzed the predictive edPA software tool's prediction quality with synthetic log data of a simple repair process which we obtained from [16]. Table 1 exhibits measurements on prediction quality. For numerical predictions, i.e. the overall duration, the mean square error (MSE) of the regression model of the predictive edPA solution is compared to the MSE of a simple arithmetic mean of the training data. Table 1 also depicts the classification accuracy. All values are related to the execution stages and the number of attributes used in the prediction model. This allows us to assess the relationship between the detail of data preparation in the EPN and the value gained in terms of prediction quality. While the arithmetic mean outperforms the proposed regression approach, the classification model works well. Only Execution Stage 3_1 does not perform. For classification and regression, additional attributes do not necessarily improve the predictive power. The classification accuracy does not increase after the first execution stage: A business user knows early in the process whether the instance will show exceptional behavior. A more detailed report on this analysis and further evaluation measures are given in [1].

Table 1. Prediction errors of edPA with synthetic log data [1]

Execution Stage	Number of Attributes	Regression (MSE)	Arithmetic Mean (MSE)	Classification (Accuracy)
1	7	378.407	377.9681	73.5
2	12	379.434	377.9681	73.5
3_1	16	361.6183	359.8436	49.41
3_2	16	409.2722	405.1011	86.18
4	28	378.791	377.9681	73.5

References

1. Schwegmann, B., Matzner, M., Janiesch, C.: A Method and Tool for Predictive Event-Driven Process Analytics. In: Proceedings of the 11th International Conference on Wirtschaftsinformatik, pp. 721–735. Leipzig, Germany (2013)
2. Castellanos, M., Casati, F., Dayal, U., Shan, M.C.: A comprehensive and automated approach to intelligent business processes execution analysis. Distributed and Parallel Databases 16(3), 239–273 (2004)
3. van der Aalst, W.M.P., Schonenberg, M.H., Song, M.: Time prediction based on process mining. Information Systems 36(2), 450–475 (2011)
4. van Dongen, B.F., Crooy, R.A., van der Aalst, W.M.P.: Cycle Time Prediction: When Will This Case Finally Be Finished? In: Meersman, R., Tari, Z. (eds.) OTM 2008, Part I. LNCS, vol. 5331, pp. 319–336. Springer, Heidelberg (2008)

5. Eder, J., Pichler, H.: Duration Histograms for Workflow Systems. In: Proceedings of the IFIP TC8 / WG 8.1 Working Conference, pp. 239–253 (2002)
6. Castellanos, M., Salazar, N., Casati, F., Dayal, U., Shan, M.-C.: Predictive business operations management. In: Bhalla, S. (ed.) DNIS 2005. LNCS, vol. 3433, pp. 1–14. Springer, Heidelberg (2005)
7. Etzion, O., Niblett, P.: Event Processing in Action. Manning Publications, Cincinnati (2010)
8. Eckert, M.: Complex Event Processing with XChange EQ: Language Design. In: Formal Semantics, and Incremental Evaluation for Querying Events (2008)
9. Shmueli, G., Koppius, O.R.: Predictive Analytics in Information Systems Research. MIS Quarterly 35(3), 553–572 (2011)
10. Janiesch, C., Matzner, M., Müller, O.: Beyond process monitoring: a proof-of-concept of event-driven business activity management. Business Process Management Journal 18(4), 625–643 (2012)
11. Chang, C.-C.C., Lin, C.-J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2(3) (2011)
12. Hsu, C.-W., Chang, C.-C., Lin, C.-J.: A Practical Guide to Support Vector Classification, http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf
13. Vercellis, C.: Business Intelligence: Data Mining and Optimization for Decision Making. John Wiley & Sons, West Sussex (2009)
14. Java Data Mining Package, http://sourceforge.net/projects/jdmp
15. Weka - Machine Learning Software in Java, http://sourceforge.net/projects/weka/
16. Process Mining Group: Process Mining Event logs, http://www.processmining.org/logs/start