Wirtschaftsinformatik Proceedings 2013 Wirtschaftsinformatik

2013

A Method and Tool for Predictive Event-Driven Process Analytics

Bernd Schwegmann
University of Muenster, ERCIS, Münster, Germany, bernd.schwegmann@uni-muenster.de

Martin Matzner
University of Muenster, ERCIS, Münster, Germany, martin.matzner@ercis.uni-muenster.de

Christian Janiesch
Karlsruhe Institute of Technology (KIT), AIFB, Karlsruhe, Germany, christian.janiesch@kit.edu

A Method and Tool for Predictive Event-Driven Process Analytics

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

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

Abstract. Business value can be lost if a decision maker's action distance to the observation of a business event is too high. So far, two classes of information systems, which promise to assist decision makers, have been discussed independently from each other only: business intelligence systems that query historic business event data in order to prepare predictions of future process behavior and real-time monitoring systems. This paper suggests using real-time data for predictions following an event-driven approach. A predictive event-driven process analytics (edPA) method is presented which integrates aspects from business activity monitoring and process intelligence. Needs for procedure integration, metric quality, and the inclusion of actionable improvements are outlined. The method is implemented in the form of a software prototype and evaluated.

Keywords: Operational Business Intelligence, Predictive Event-Driven Process Analytics, Event-Driven Business Process Management

1 Introduction

Software tools and methods for operational Business Intelligence (BI) support have emerged and proliferated. While BI generally refers to a collection of decision support technologies "aimed at enabling knowledge workers […] to make better and faster decisions" [1], operational BI is used "to reduce the latency between when operational data is acquired and when analysis over that data is possible" [1]. The goal is to reduce reaction time [2], [3]. Operational BI focuses on transactional data typically coming from logs of business process management systems (BPMS). BPMS execute technical process models with its runtime component, i.e. a process engine. Process analytics encompasses a set of measurement and analysis techniques to evaluate the past, to understand what is happening at the moment and to predict the future in the context of a business process [3]. However, a review of today's process-aware enterprise systems reveals challenges: First, many BPMS lack sophisticated capabilities to analyze log data [4]. Second, process analytics is limited to analyzing the past and monitoring the present only. Third, process analytics is not able to take action on a

11th International Conference on Wirtschaftsinformatik,
27th February – 01st March 2013, Leipzig, Germany

business process in result to observations made. It would require hardwiring BPMS and process analytics systems to achieve end-to-end insight-to-action.

We propose to assist human decision makers and automated decision making with (near) real-time decision support based on historic and actual process data as well as predictions of the future. Our research goal is to design a method and a system, a closed-looped combination of BPMS and predictive process analytics, which allows an earlier reaction to business events through accurate predictions. The approach strives to concern with analysis and decision activities before the specific business event occurs and thus to omit analysis and decision latency at run-time (Fig. 1).

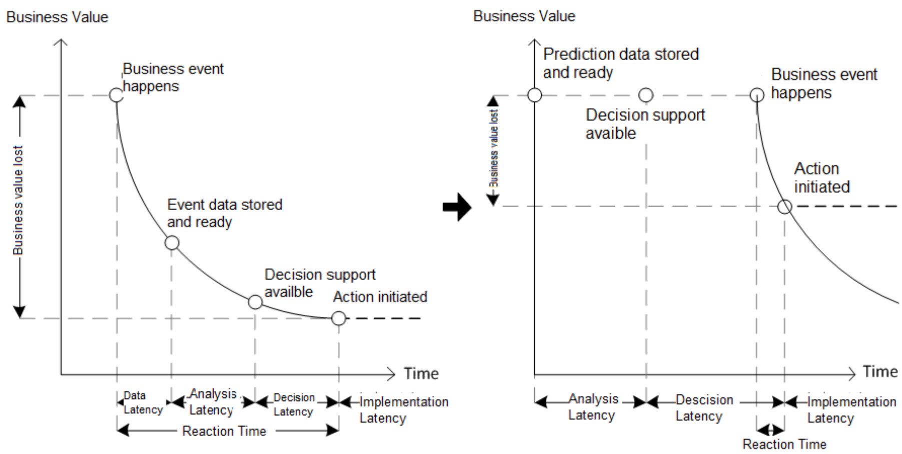

Fig. 1. Reduction of reaction time through predictive event-driven process analytics (cf. [2,3]).

This paper reports on the development of this method and the software prototype for predictive event-driven process analytics (edPA). In Section 2, we discuss related work. Section 3 discusses requirements towards an edPA method. Section 4 presents the edPA method. Section 5 describes the software prototype which implements the method. The paper closes with a summary and conclusions.

2 Related Work on Process Analytics

2.1 Analyzing the Past and the Future of Business Processes

Process controlling involves, apart from standard reporting, explorative data analysis such as *process mining* which “aims at the automatic construction of models explaining the behavior observed in the event log” [5]. Its focus is on concurrent processes rather than on static or mainly sequential structures [6]. To its very nature, process mining is an ex-post analysis of process behavior. Data mining models are used in process mining with continuous, categorical, and numerical attributes of business processes for prediction purposes [7]. Predictors are calculated for event traces which refer to a target variable such as the remaining cycle time. Both, predictors and target variables, are computed for all partial event traces of a process instance. Regression functions are built-up from the resulting data sets for each partial trace, and they enable predictions of future partial traces. Predictive process controlling uses regression algorithms with numerical and categorical variables for predicting a continuous time variable. It makes use of abstract states of a process that may be filled with measures of predicted future behavior. The duration histogram concept [8] provides means to consider the control flow structure for building prediction models.

Business process intelligence (BPI) refers to the post-execution prediction of future process behavior. BPI analyses consolidated data by employing process warehouse (PWH) architectures. Specific applications of the PWH analyze past process instances to predict and prevent unwanted outcomes in running processes [9–11]. Prediction models can help sorting instances into a certain category or to predict a numerical outcome. Since required attributes may yet be missing when analyzing a running instance, a prediction model for each process execution stage based on the available attributes is needed [11]. Predicted metrics may be used to trigger actions on a running instance via the notification engine of a BPMS or operational system. In effect, predictive BPI is aware of abstract execution stages and presents ways to handle the control flow structure. The PWH architecture caters for different predictive application scenarios such as making predictions actionable and providing context for events. It allows for a tight integration with information modeling which is missing in process mining. BPI also uses advanced automation techniques for the data mining procedure.

2.2 Analyzing the Present of Business Processes

Business Activity Monitoring (BAM) provides low latency information about the execution of a business process [12]. Thus, BAM applications require information from BPMS with little delay [13]. BAM can be structured along five phases [14]. In order to observe events from different systems a BAM solution requires Extract-Transform-Load (ETL) capabilities to process the data in a common format. In the *evaluate* phase, “the timely computation of process metrics, such as the execution time or the number of failures” [14 is done. The *detect* phase reasons over present and future process behavior. Detected situations are often *diagnosed* manually to find root causes to a problem and are finally *resolved* according to a resolution strategy.

Complex event processing (CEP) technology has been proposed as state-of-the-art for implementing BAM. CEP engines reason over process-related events in an online, real-time mode [15]. We refer to BAM approaches which employ CEP technology as event-driven BAM. Process-related events are processed in a push-based approach. BAM systems receive single business and technical events and transform them into higher level knowledge [16]. Events typically are sent from a BPMS. They are processed by CEP systems. CEP engines execute an event processing network (EPN) in which individual agents (EPA) detect, filter, and transform events. Events can be related to other events to find causalities or derive new complex events by filtering, transforming, and detecting patterns in single events [15].

3 Designing an Event-Driven Process Analytics Method

3.1 Method Engineering Process

In the IS discipline, methods describe systematic procedures "to perform a systems development project, based on a specific way of thinking, consisting of directions and rules, structured in a systematic way in development activities" [17]. Methods strive to close the gap between current organizational performance and a set of consensual goals [18]. A method should be tool supported in order to make it accessible for the practice [17]. Method engineering comprises all activities related to the development of methods. It has been explained as a process that comprises of three phases [19]: requirements engineering, method design, and method implementation. Requirements engineering encompasses discovering, prioritizing, documenting, representing, and maintaining a set of requirements for a specific method. Method design comprises all activities of the actual method construction. Method implementation and evaluation finally subsumes the activities of implementing the method in an information system and testing it. Accordingly, Section 3.2 presents requirements towards a method for predictive edPA. The method design is discussed in Section 4. Section 5 informs about the prototypical implementation.

3.2 Requirements for Predictive Event-Driven Process Analytics

Event-driven BAM is an instrument to measure process performance while predictive analytics is a development option for event-driven BAM for analyzing observations to make predictions. Process performance management (PPM) yet employs current and target indicators; predicted performance indicators for running processes constitute a new dimension for leveraging operational performance. Thus, procedures and capabilities of predictive analytics and event-driven BAM need to be addressed by a method for predictive edPA. Data mining regards issues such as the availability of data and includes procedures to ensure high quality prediction models. For example an abundance or absence of events can be caused by gateways in the process control flow. Completeness of data needs to be considered in the process of defining prediction models accordingly. The detect phase of the BAM procedure reasons over process behavior. A prediction model might assist this step since it can take the currently available indicators as input while predicting the future behavior of the process. As PPM acts on operational and tactical level mainly it requires providing predictions on both process and instance level. Table 1 subsumes the characteristics in the form of evaluation criteria a predictive edPA method needs to cater for.

So far, there exists no integrated procedure of predictive analytics and event-driven BAM, but few applications on top of a PWH demonstrate the automation of data mining procedures for process prediction. However, yet they are not explicitly embedded in the PPM methodology nor are they a procedure model for predictive edPA.

Concerning the *soundness of predictions*, predictive process controlling uses measurements on events or abstract process states formed by events to predict metrics on instance level only. Measurements over several running process are not yet implemented. In contrast, BPI provides instance and process level predictions. Thus, the capability to predict categorical and numerical process metrics on different levels is also a requirement for predictive edPA. Both, BPI and process controlling do not cater for the timeliness quality requirement on process metrics. The performance evaluation in process mining is pull-based and relies on logs which typically reside in data stores with high latency compared to low latency event-driven BAM. In BPI, near real-time capabilities are used for BAM without predictive analytics only. Additionally, the PWH architecture requires events from operational systems to be processed by other systems and databases before they are actually available as predictions on a dashboard or can be fed back to a BPMS. This results in medium latency and offers an opportunity for improving the timeliness of a process metric through edPA. Accuracy of predictions is a technical challenge if the procedure for building a prediction model is automated. It is also a procedural issue as it involves following good practices in predictive analytics. A prediction goal could require including metrics external to the organization but also internal metrics obtained from a BPMS. The metric type affects the cost-effectiveness criterion for process metrics since internal metrics are typically cheaper. Thus, a solution must be empowered to integrate different event sources and a procedure for predictive edPA must support reasoning about the costs of metrics. Comprehensibility of process metrics must also be ensured by the procedure. Practicability requires the technical abilities to automatically initiate action and the procedural abilities to decide what action is appropriate.

Table 1. Evaluation criteria for predictive edPA

Criterion	**Description**
Procedure integration	Degree of integration between BAM and predictive analytics procedures with respect to PPM methodology, including characteristics of instance prediction
Soundness of predictions	Predictions need to be provided on the instance as well as on the process level following the quality criteria for process metrics
Actionable improvements	Ability to improve the process performance by proactive actions executed on running process instances

Improvements are applied in the existing approaches by evaluating the condition for an improvement and carrying out the improvement on running process instances. Further, different prediction based alternatives might be tested. In process mining this feature is used to recommend future activities. At conditional branches, different alternatives can be compared – resulting in recommendations if the predictions can be ranked according to a business goal [20]. Thus, *actionable improvements* require the BPMS to offer the capability to implement improvements on running business processes which resorts this requirement to a technical challenge. From a procedural view, the conditional improvement is a valuable concept that should be integrated in a method for predictive edPA.

4 Predictive Event-Driven Process Analytics Method

In order to shorten the decision maker's action distance, we propose a method which facilitates integrating BAM and predictive analytics aspects and thus helps to improve organizational performance. The method is based on Six Sigma – a PPM approach for identifying and eliminating unnecessary or inefficient activities from business processes [21]. Six Sigma comprises of five phases: *define*, *measure*, *analyze*, *improve*, and *control* (DMAIC) [22]. Based on DMAIC we propose a method comprising of the phases prediction preparation, predictors modeling, prediction model definition, prediction model application, and prediction model controlling. First, relevant measures are identified and then predictions are defined. Accordingly, the *prediction preparation* phase, the *predictors modeling* phase, and the *prediction model definition* phase, together, address the DMAIC "define" phase. DMAIC's measure phase, the analyze phase, and the improve phase are subsumed in our method's *prediction model application* phase. The control phase is adopted as *prediction model controlling*.

4.1 Prediction Preparation

Domain experts choose adequate predictors according to their usefulness for business users on either operational or tactical level (goal definition). The domain experts accounts for the process metric quality criterion *comprehensibility* and *practicability*.

Processes are executed by a BPMS. Predictive edPA uses process-related event data from BPMS to calculate measures for prediction models (i.e. data collection). The prediction model definition and its application require event data to be represented in the same format. A uniform format facilitates e.g., using logs or a PWH for model building and event-driven BAM for model application. The online monitoring of event-driven BAM and the storing capabilities of logs or PWH could be effectively combined, thus allowing for a rich prediction model based on large data sets.

4.2 Predictors Modeling

Process level metrics change in values over time but not in their general structure. Thus, a time series of these metrics is a good predictor. In contrast process instance level metrics arise during process execution. The number of potential predictors increases dynamically. Data preparation on process level is straightforward. Data needs to be recorded over time and then to be provided to *prediction model definition* and *prediction model application*. A prediction model developer selects the required metrics and defines a sliding input window for the EPA which monitors the measures.

Process instance metrics, instead, require more extensive data preparation as the process state changes during execution. Execution stages can be used to refer either to activities of a workflow, to entire sub-processes or other structures of the control flow which comprise several activities. Execution stages abstract events from the actual workflow model by defining process states. In an event-driven BAM implementation, the events required for a metric or predictions are observed by an EPA, and subsequent EPAs transform them into meaningful categorical or numerical metrics. These metrics are added to an EPA which reflects an execution stage in the process. The information available on each stage is transferred to the next one until the final execution stage contains all the information available about the instance. Finally, an EPA containing the metric values at the end of the execution of the business process needs to be added to the EPN. It labels the predictor metrics of the execution stages with a response variable. Since execution stages refer to activities which are ordered by the control flow of a workflow, an execution stage might encounter abundance (e.g., loops) or incompleteness (e.g., XOR split) of information. 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 [23].

4.3 Prediction Model Definition

In event-driven BAM, the EPN schema can be also used at design time to support data exploration. Then an EPN is provided with historical event data to fill execution stages with metrics from completed executions. These metrics can be tested with data exploration tools on their significance. Visualization can help recognizing outlier and metrics with small sample sizes, and statistical figures such as variance and arithmetic mean can help in reducing the number of dimensions.

From the cleaned data set, final predictors and response variables for instance level or process level prediction need to be selected. The decision on the response variable is guided by the prediction goal and the data availability. The decision on the predictors is guided by data availability at prediction time. Measurement precision issues can be neglected since all data collectors, event logs, the PWH, and the event-driven BAM platform record the events without any loss of precision. Then, a prediction method needs to be selected. Castellanos et al. classify available data mining techniques for business process analysis by their “popularity, intuitive interpretation, or superior performance” [9]. Table 2 shows relevant data mining techniques for instance and process level prediction. The developer chooses from these techniques to build the prediction model according to the prediction goal. For instance, a decision tree has advantages if intuitive interpretation and transparency are prediction goals while a support vector machine (SVM) has advantages in performance and handling of complex relationships. A SVM is also a suitable technique for categorical and numerical metrics on instance level and thus a good candidate for several prediction problems which can also predict time series of process level metrics.

Table 2. Methods available for instance level and process level predictions

	Instance level prediction		**Process level prediction**
	Categorical metric	***Numerical metric***	***Numerical metric***
Decision tree	●		
Rule model	●		
SVM	●	●	●
Regression tree		●	

Model selection reduces prediction errors and over-fitting by comparing the prediction accuracy of prediction models. In general the prediction error is the difference between the real value and the predicted value at the end of execution. For classification tasks the accuracy defines the percentage of correctly classified instances. The prediction function returns "1" for a correctly classified instance, otherwise "0":

$$\text{Accuarcy} = 100\ \%\ \text{x}\ 1\backslash n \sum\nolimits^{n}_{i=1} \text{predict}\ (b_i)_{\text{correct}}. \tag{1}$$

Predictions for numerical values could for instance use the mean square error (*MSE*) measure [20]. We deliberately abstain from a detailed discussion on the diverse prediction quality measures available for numerical values. Using the same data set for building the prediction model and for testing the accuracy/ calculating the *MSE* should be avoided since the model might over-fit the data in the training set and would have low predictive power in fact. To address this challenge the developer can use cross validation methods [24]. We adopt a popular and widely used method. In k-fold cross validation the original data set is divided into k-parts of equal size. One part is used for validating the predictive model, while the others are used to build the predictive model. This procedure is repeated for every part, so that each part is validated using the remaining parts, i.e. there is a predicted and a real value for every record in the original data set. From these values the *MSE* is calculated.

Some predictive techniques require choosing parameters that are unknown for a given problem. For instance, SVM requires selecting a cost parameter *C* and the radial basis function kernel of the SVM requires selecting a parameter γ. To find these parameters the developer can conduct a grid search. Here, a grid of parameter combinations is defined, where each combination is used for cross validation. Grid search finds the smallest *MSE* for numerical problems and the biggest accuracy for classification tasks.

4.4 Prediction Model Application

This phase comprises the measurement of predictors, the analysis of predictors, and the initiation of proactive process improvements. For the measurement of predictors the EPN is used to observe real events and to measure metrics during the execution of business processes. The instantiated EPN receives the events defined in the prediction definition phase. The EPN evaluates the events and transforms them into predictors in real-time. In result, the execution stages hold the latest metrics on executed processes.

Then the predictors need to be analyzed. The attributes measured before are applied to the prediction model. The prediction model makes a prediction in order to detect a future state of the workflow. The prediction itself is a new metric which can be fed into the EPN again. Based on the prediction result, proactive process improvement could be initiated, i.e. the process behavior might be influenced in order to avoid a predicted undesirable behavior. Fig. 2 contains a selection of possible mechanisms for proactive process improvements.

The prediction receiver consults prediction quality information in order to estimate the accuracy of the prediction. A low accuracy may be an advice to ignore the prediction. The prediction receiver needs to decide whether to carry out the improvement on the instance or process level. Typically this is indicated by the metric type. The prediction receiver chooses either an active or passive process improvement. Active improvements change properties of one process instance or for all instances of one model. Passive improvements do not change properties of the process but help reducing potential damage. As a last step before carrying out the action, the prediction receiver needs to estimate the escalation costs.

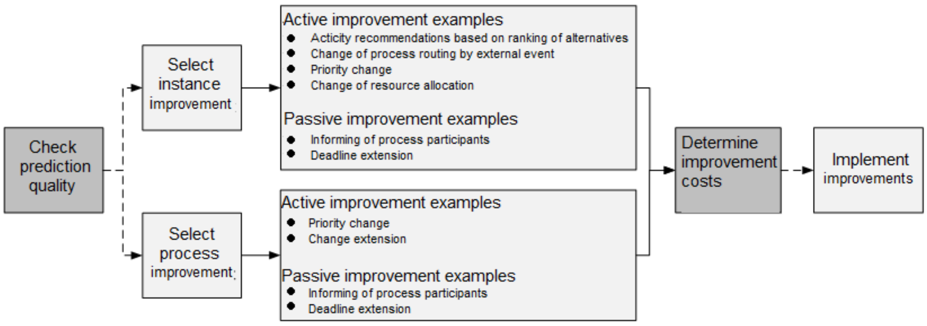

Fig. 2. Potential actions of proactive process improvement

4.5 Prediction Model Controlling

The controlling phase serves two purposes: First, periodical retraining is required if the underlying conceptual model has changed as new observations might be available or some relationships captured in the prediction model might be not valid anymore. Observing these issues is the task of the developer. Heavy use of proactive process improvements may indicate fundamental problems in the process design. Thus, the domain expert needs to be consulted in order to perform a modification or replacement of the process design.

5 Prototypical Implementation, Demonstration, and Evaluation

5.1 System Architecture and Exemplary Application

We have implemented a software prototype which supports the presented method on top of an internal release of a BPMS which integrates CEP functionality [25]. It comprises three main components: (1) a design component for modeling predictors and defining predictions, (2) a prediction runtime for the analysis of key performance indicators (KPI), and (3) a visualization frontend. In the following demonstration, we focus on the prediction runtime and acknowledge that there are modified BPMS design components to model EPNs with prediction capabilities and a visualization frontend which can adequately communicate the prediction results. See Fig. 3 for an overview of the architecture of the prediction runtime.

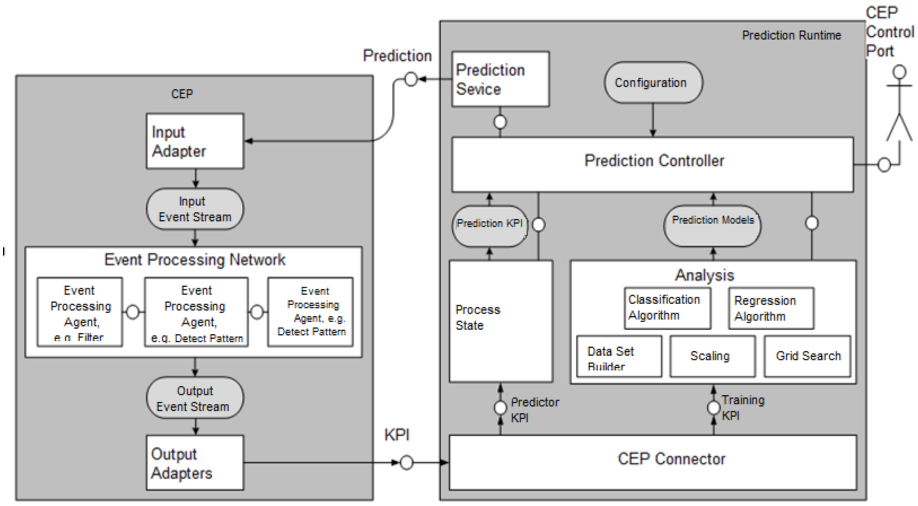

Fig. 3. Detailed prediction runtime architecture

The prediction runtime accounts for two tasks in the CEP engine. First, it acts as an event consumer to train a prediction model and to receive the latest predictor for a given process instance or process level KPI. Second, the prediction runtime acts as event producer when making a prediction through an enactment of the prediction model with the latest predictor metrics. 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 a process improvement in a BPMS. The prediction runtime component contains the main logic of the prototype. It has five subcomponents from which the analysis component and the prediction controller are the major ones. The other components are the process state, the CEP connector and the prediction service component. The analysis component is started by the prediction controller. Depending on the prediction configuration an algorithm for prediction is selected. This algorithm calls a data set builder to form a data set for each execution stage of the process. The data sets are scaled and a grid search including cross-validation is conducted to find the best parameters for the algorithms. The predictions models are then stored and made available for prediction making.

For the demonstration of the prototype, we implemented a *simple repair process* with a synthetic log. A solver and a tester interact in this workflow to repair a defect telephone. It starts with a defect analysis, informs the user about the defect and starts in parallel a repair trial (simple or complex). Then a tester checks the repair for success and if necessary starts another repair trial for the solver.

Prediction preparation. In our demo case, two prediction goals are defined by a domain expert: the reduction of the process duration and the required rework. With continuous predictions for the processes duration (numerical prediction), process participants can satisfy the information need of the customer and adjust internal and external expectations on process performance. They can try to speed up the process if the predicted time indicates a problem, for instance by engaging two solvers for one repair. After the repair a further repair loop (rework) might be required or not (categorical prediction, control flow prediction). The predicted metric is a good indicator to identify “abnormal” process instances at an early stage.

Predictors modeling. First, the execution stages are modeled. A metric of the process start time is defined at the first execution stage. The stage is assigned to the *registration of a repair process*. The resource performing the first activity is added (encoded as six-figure vector, either “0” or “1”, for each tester’s name). The attributes of the first stage are then transferred to the second execution stage. A time metric for the completion of the *analyze defect* activity is added as well as a metric for the defect type and the phone type (both categorical, encoded as before). Thus, the second execution stage relates to metrics referencing events until the completion of *report defect*.

After the report defect activity, the control flow splits. To reflect the current state of the process in one data structure all execution stages after the split contain metrics from both paths until the control flow merges again. Temporarily missing values are filled up by replacement techniques (first non-null expression). These execution stages adequately reflect the process state since both paths are executed concurrently and it is unknown ex-ante which path is executed earlier. This approach ensures that the maximum of metrics is available for a prediction model. This could not be ensured by separate execution stages with separate metrics for each path.

Next, the first main path is split by an exclusive choice into a *simple repair* and a *complex repair* activity which then are merged to *repair test*. Therefore, two different sub execution stages need to be defined, each containing the attributes of the second execution stage. Since *inform user* activities are executed in the second main path, a metric for the activities’ completion time is added to each execution stage. Start and end times of each *repair activity* are also added. An iteration counter is assigned as activities can be looped. For the first main path, the last execution stage is modeled which relates to the completion of the *repair test*. Since the execution stage has to consider a merging control flow, the first non-null expression is used for all attributes that are transferred from preceding sub execution stages. The developer adds categorical attributes for the results of the repair test and the iterations of the *test repair* activity.

Finally, the response stages are modeled. A response stage contains the duration metric which is calculated from the timestamp of the last possible event in the process, i.e. *report result,* and the first event in the process, i.e. *register repair*. The other response stage contains a categorical attribute for the iteration of the test *repair activity* (“1” if more than one iteration, otherwise “-1”).

Prediction model definition. The developer selects the occurrence count, the current attributes of the process object and timestamps of the activities as predictor variables. Response variables are the process duration and an indicator of whether a repair is multiply conducted. The prototype employs a SVM (LIBSVM) which allows for regression and classification. The controls of the design component allow tagging an event stream as execution or response stage. For a response stage the developer selects the algorithm type applied to the response variable and the predictor variables (data sets) in each execution stage. The prototype automatically selects the event attributes containing the metrics. The developer configures the application server and instantiates the monitoring model. Events are uploaded to the instance of this monitoring model. Finally, the developer triggers the automated model generation and selection for each execution stage by informing the prediction runtime about the control port of the EPN.

Prediction model application. The prototype observes process events and measures the metrics defined by the prediction model. The business user receives a prediction which is directly accessible from his user interface. He uses the prediction quality information when deciding on proactive actions. For example, he may select an instance improvement and inform the customer about the expected process runtime. The process quality as perceived by the customer is improved since the customer can plan his activities accordingly. In case the prediction quality would have been less accurate or would indicate problems the business user might have raised processing priorities to satisfy the customer.

Prediction model controlling. In this last phase of the procedure model, the developer retrains the prediction model periodically and evaluates the use of the prediction by the business users in order to further refine the prediction models or to trigger a process redesign.

5.2 Evaluation

In order to assess the problem solving capability of our approach we, first, compare the presented edPA method with the requirements as presented in Section 3. The method's problem solving capability is analyzed following a quantitative approach.

Comparison with requirements. The presented edPA method integrates predictive analytics and event-driven BAM procedures. In terms of the DMAIC cycle, the developer defines predictors as demonstrated with the repair process according to the proposed structures and details the information from the first to the last execution stage. The prototype executes the measure and analyze phase and the business user uses the prediction in the improvement phase and applies proactive actions. Cost-effectiveness of the process metrics is demonstrated by using internal process metrics. Comprehensibility is demonstrated by using metrics which are easy to understand for all process participants. Practicability involves informing the customer. Due to limitations of the prototype, the control phase is underrepresented in the demo and data exploration steps could not be demonstrated. With regard to the software prototype, the design component proved its ability to configure execution stages and predictions on process measures. The prediction runtime demonstrated the ability to predict categorical and numerical outcomes and the use of numerical and categorical attributes. This ability is important in a business process context since it allows predicting arbitrary data from process events. The runtime component demonstrated that it succeeds in capturing the state of a business process to make a prediction in real-time. The visualization frontend makes the prediction comprehensible for business users.

Measurements on prediction quality. The overall fitting of the procedure and the prototype can be estimated by measurements of the prediction quality. These were captured while building the prediction models for the execution stages. Table 3 shows measurements for the prediction quality in the demo scenario. For numerical predictions, i.e. the overall repair process duration, it compares *MSE* of the regression model of the proposed solution with MSE in case a simple arithmetic mean of the training set is used as prediction. In addition it depicts the accuracy for classification, indicating how precisely the proposed solution can predict whether more than one repair trial will be required in the running process. These values are related to the execution stages and the number of attributes used in the prediction model. This allows assessing the relationship between complex data preparation in the EPN, which is necessary for a big number of attributes, and the value gained in terms of prediction quality.

Table 3. Prediction errors of the proposed solution

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

While the arithmetic mean outperforms the proposed regression approach, the classification model works quite well. Only *Execution Stage 3_1* does not perform. For classification and for regression, additional attributes do not necessarily improve the predictive power. The accuracy for the classification is already precise for the 1st execution stage: A business user knows early in the process whether the instance will show exceptional behavior or not, i.e. if more than one repair iterations are needed.

Implications. The discussion above reveals practical and research related implications. With respect to practical applications of the procedure model, the modeling of the EPN could be simpler since the demonstration showed that a large number of attributes does not imply a high prediction quality. Classification seems to be a more promising use case for software vendors since it is accurate at an early stage and it is possible to predict different paths after control flow splits. The above discussion further suggests relying on the average if predicting durations, which includes avoiding additional modeling tasks. However, the issue with the duration prediction might also be caused by the use of synthetic log files and the underrepresented data exploration.

6 Discussion and Conclusion

A short reaction time to a business event is a critical factor in the success of an enterprise. Systems for predictive analytics and real-time monitoring of business processes have only been considered separately so far. In this paper we proposed to base predictions on current process data generated in a BPMS using an event-driven approach in an attempt to reduce data and analysis latency. A review of the extant literature revealed a lack of methodological guidance on how to integrate approaches of these two kinds. Based on a further requirements analysis, we elaborated a method which facilitates predictive edPA based on the Six Sigma DMAIC phases. The method exhibits actions required for a predictive edPA which formerly have been described in the separated publication contexts of predictive analytics and BAM. We have exemplified the method's capabilities with a simple repair process which we employed in order to demonstrate the method's applicability. A discussion of the implementation and its forecasting quality uncovered that the presented approach is likely to provide accurate predictions which would indeed reduce reaction time of decision makers.

Our work contributes to theory by integrating the to date separated areas of process analytics which analyzes the past, the future and which monitor the presence. In particular, we highlighted the intersections of the three dimensions. The procedure outlines, e.g., which data needs to be shared among monitoring components and prediction components in order to facilitate edPA. Also, with our approach we offer a new perspective on how process-aware information systems can work together with analytical systems in a near real-time manner. The architecture differs further from extant concepts through its loosely coupled nature. Hence, our approach facilitates, e.g., the connection of several BPMS and several analytical systems.

From a managerial perspective, we identify further implications: First, the presented procedure gives an overview on the various tasks to address and design decisions to take when integrating (event-driven) BAM and process analytics techniques. The procedure allows reducing the decision maker's action distance and thereby gaining additional business value through (pro-)active business process management. Second, the procedure is made available to practitioners in a prototypical implementation. The prototype was integrated in an internal release of a commercial BPMS.

We were not yet able to implement the full set of requirements. In particular, future work will have to elaborate on coupling event-driven BAM with persistent data stores and with adding data exploration components to the prototype. Further, the re-training of prediction models has not been addressed. Further work will be required to investigate how data mining algorithms, which allow for online training of the prediction models, can be integrated into the architecture. Also, we observed several shortcomings in the integration with BPMS. Practically speaking, these include but are not limited to the lack of a common business event format and possible a distinction of life cycle and business events, the lack of a standardized set of process log events, and the lack of standardized operations to trigger BPMS for automated insight-to-action.

References

1. Chaudhuri, S., Dayal, U., Narasayya, V.: An Overview of Business Intelligence Technology. Commun. ACM. 54 (8), 88–98 (2011)
2. Hackathorn, R.: Minimizing Action Distance. The Data Administration Newsletter. (2003)
3. zur Muehlen, M., Shapiro, R.: Business Process Analytics. In: vom Brocke, J., Rosemann, M. (eds.): Handbook on Business Process Management, Vol. 2, pp. 137–157. Springer, Berlin (2010)
4. Cheung, M., Hidders, J.: Round-trip Iterative Business Process Modelling Between BPA and BPMS Tools. Business Process Management Journal. 17 (3), 461–494 (2011)

5. van der Aalst, W.M.P., Reijers, H.A., Weijters, A.: Business Process Mining: An Industrial Application. Inform. Syst. 32 (5), 713–732 (2007)
6. Verbeek, H.M.W., Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: XES Tools. CAiSE 2010 Forum. (2010)
7. van Dongen, B., Crooy, R., van der Aalst, W.M.P.: Cycle Time Prediction: When Will This Case Finally Be Finished? In: Meersman, R., Tari, Z. (eds.): OTM 2008. LNCS, Vol. 5331, pp. 319–336. Springer, Berlin (2008)
8. Eder, J., Pichler, H.: Duration Histograms for Workflow Systems. In: Rolland, C., Brinkkemper, S., Saeki, M. (eds.): Proceedings of the IFIP TC8/ WG 8.1 Working Conference, pp. 239–253. Kluwer, Deventer (2002)
9. Castellanos, M., Casati, F., Dayal, U., Shan, M.C.: A Comprehensive and Automated Approach to Intelligent Business Processes Execution Analysis. Distrib. Parallel Dat. 16 (3), 239–273 (2004)
10. Castellanos, M. et al.: Predictive Business Operations Management. In: Bhalla, S. (ed.) DNIS 2005, LNCS, Vol. 3433, pp. 1–14. Springer, Berlin (2005)
11. Grigori, D., Casati, F., Castellanos, M., Dayal, U.: Business Process Intelligence. Computer in Industry 53 (3), 321–343 (2004)
12. Schiefer, J., List, B., Bruckner, R.M.: Process Data Store : A Real-Time Data Store for Monitoring Business Processes. In: Mařík, V., Retschitzegger, W., Štěpánková, O. (eds.): DEXA 2003. LNCS, Vol. 2736, pp. 760–770. Springer, Berlin (2003)
13. DeFee, J., Harmon, P.: Business Activity Monitoring and Simulation. In: Fischer, L. (ed.) Workflow Handbook. pp. 53–74. Future Strategies, Lighthouse Point, FL(2005)
14. Medeiros, A.K.A. de et al.: An Outlook on Semantic Business Process Mining and Monitoring. In: Meersman, R., Tari, Z., Herrero, P. (eds.): OTM 2007 Workshops. LNCS, Vol. 4806, pp. 1244–1255. Springer, Berlin (2007)
15. Etzion, O., Niblett, P.: Event Processing in Action. Manning, Greenwich, CT (2010)
16. Eckert, M.: Complex Event Processing with XChange EQ, München (2008)
17. Brinkkemper, S.: Method Engineering: Engineering of Information Systems Development Methods and Tools. Inform. Software Tech. 38 (4), 275–280 (1996)
18. Checkland, P., Holwell, S.: Information, Systems and Information Systems. Wiley, Chichester (1998)
19. Gupta, D., Prakash, N.: Engineering Methods from Method Requirements Specifications. Requirements Engineering 6 (3), 135–160 (2001)
20. van der Aalst, W.M.P., Pesic, M., Song, M.: Beyond Process Mining. In: Pernici, B. (ed.) CAiSE 2010. LNCS, Vol. 6051, pp. 38–52. Springer, Berlin (2010)
21. Conger, S.: Six Sigma and Business Process Management. In: v. Brocke, J., Rosemann, M. (eds.): Handbook on Business Process Management, Vol. 1, pp. 124–148. Springer, Berlin (2010)
22. Adams, C.W., Gupta, P., Wilson, C.E.: Six Sigma Deployment. Butterworth-Heinemann, Amsterdam (2003)
23. Shmueli, G., Koppius, O.R.: Predictive Analytics in Information Systems Research. MIS Quart. 35 (3), 553–572 (2011)
24. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York (2004)
25. 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)