Decision Point Analysis of Time Series Data in Process-Aware Information Systems*

Reinhold Dunkl, Stefanie Rinderle-Ma,
Wilfried Grossmann, Karl Anton Fröschl

University of Vienna, Faculty of Computer Science
{reinhold.dunkl, stefanie.rinderle-ma, wilfried.grossmann,
karl-anton.froeschl}@univie.ac.at

Abstract. The majority of process mining techniques focuses on control flow. Decision Point Analysis (DPA) exploits additional data attachments within log files to determine attributes decisive for branching of process paths within discovered process models. DPA considers only single attribute values. However, in many applications, the process environment provides additional data in form of consecutive measurement values such as blood pressure or container temperature. We introduce the $DPA^{TimeSeries}$ method as an iterative process for exploiting time series data by combining process mining and data mining techniques. The method also offers different approaches for incorporating time series data into log files in order to enable existing process mining techniques to be applied. Finally, we provide the simulation environment $DPA_{Sim}^{TimeSeries}$ to produce log files and time series data. The $DPA^{TimeSeries}$ method is evaluated based on an application scenario from the logistics domain.

Keywords: Process Mining, Decision Mining, Data Mining, Time Series Data

1 Introduction

Process mining aims at discovery and analysis of process models based on event logs. So far, process mining methodology emphasized the control flow, that is, restricting analysis to time-stamped event data (so-called log files) gathered from, or produced by, executed process instances. An extension towards the branching logic of processes is provided by Decision Point Analysis (DPA) [1]. DPA is based on enriching log file entries with additional information about process environments or other process-relevant data and aims at deriving decision rules at alternative branchings in process models. In a first step, the underlying process model is discovered. If the resulting process model contains decision points, the corresponding decision rules are analyzed using decision trees (data mining).

Fig. 1 depicts a container transportation example [2], where some temperature-sensitive cargo is transported and cargo temperature is measured repeatedly. On

* The work presented in this paper has been partly conducted within the EBMC^2 project funded by the University of Vienna and the Medical University of Vienna.

the left, the application of DPA [1] is illustrated: depending on the temperature value for each transport monitored, DPA concludes that for a temperature over 37, the vehicle has to return to its home base. Otherwise, it unloads the goods at the destination. As this example shows i) DPA takes into consideration single-valued attributes; ii) DPA is able to derive decision rules of type "x OP value" where x is the decision variable and OP is a comparison operator; iii) DPA relies on values that are stored within the event log of a process.

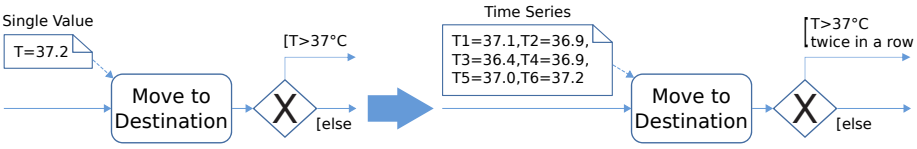

Fig. 1. Process Applications with Time Series Data

As the above characteristics show, DPA cannot adequately deal with real-world scenarios in which *time series data* are collected, e.g., in health care or container transportation. Based on time series data, *more complex* decision rules are conceivable, for example, "temperature exceeds a certain threshold for a time frame" (cf. right side in Fig. 1).

Hence, it would be desirable to process and analyze time series data by an extension of DPA. In this paper, we will present such an extension by means of method $DPA^{TimeSeries}$ that enables (a) a joint consideration of event log data and time series data, (b) iterative application of process and data/visual mining techniques, and (c) derivation of complex decision rules.

To do so, we distinguish two pertinent perspectives of this enhanced approach to process mining, viz. a method and a data perspective (Section 2). The ensuing process mining method is evaluated based on a real-world example of process analysis (Section 3). After reflecting our contribution against the state of the art in process and decision mining (Section 4), some concluding remarks (Section 5) finish this presentation.

2 Method and Data Perspective

The $DPA^{TimeSeries}$ method is illustrated in Fig. 2. As a first step it has to be decided how time series data is considered in connection with the event log data. For offering data structures within or outside the event logs that enable the application of $DPA^{TimeSeries}$, we identify the following options (cf. Fig. 2):

1. *Separation of Data:* We can prepare an analytical data set consisting of recurring measurements with sufficient temporally information to enable a matching with event data and provide this data separated from the log files.

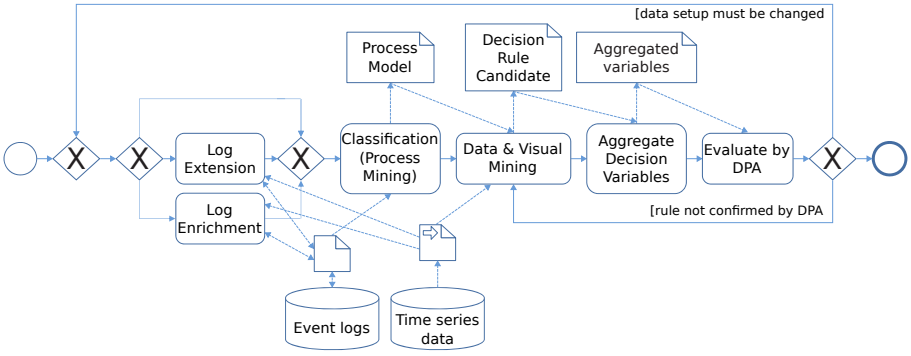

Fig. 2. $DPA^{TimeSeries}$ method (in BPMN notation)

2. *Log Enrichment:* This analytical data set can also be incorporated into the log by adding an attribute to the corresponding event within the log (e.g., a XES extension that allows such recurring measurement data structures).
3. *Log Extension:* Another approach is to dissemble the recurring measurement data and interlacing it into the log file as recurring events with single-valued attributes.

In the following, we discuss the pros and cons of these different options.

Separation of data does not modify the original event log data and therefore contributes to the maintenance of both data sets, an advantage if the event log data is used by other applications as well. The obvious disadvantage is that the connection between the event log data and the time series data is not explicitly stored and every analysis tool has to load and match the data by itself. Log enrichment and extension leads to an explication of this relation with the disadvantage of an additional preprocessing step to do so. Log enrichment does not change the number or kind of log entries as log extension does. Thus, process mining algorithms are not effected and, in turn, the resulting process models do not become more complex. Hence, the integration is in principle easier than for log extension. Log extension practically pushes the time series data into the event log what might be intended depending on the application and can therefore be an advantage as well as a disadvantage. This approach sure changes the log effectively but makes format extensions and extra files dispensable.

In summary, the choice of the approach is strongly dependent on the application. The case study presented in Sect. 3 features all three approaches.

As second step in the $DPA^{TimeSeries}$ method, process mining is used for *classifying* process execution paths along decisions made at runtime reflected by decision points in the resulting process models. In a third step we use data mining techniques such as CART, AdaBoost, Support Vector Machines as well as exploratory data mining including visual mining to explain the classification, i.e., derive the underlying decision rule. This more experimental mode of analysis, utilizing continuously improved understanding of (perhaps not yet) available

process and environment data is more appropriate at this stage of the method than a mechanical brute-force exploration.

Candidates for decision rules are transformed into aggregated variables in a fourth step. These variables can then be used to employ DPA [1] to evaluate the decision rule candidate as the last fifth step. Depending on the result, the inspection by both data and visual mining techniques has to be repeated. It is also possible that the way the time series data was reflected inside or outside the logs has to be modified.

Process Mining uses event logs that consist of a minimal data set of case ids, activity names and timestamps. It is also possible to store data values that were produced during process execution, e.g., the age of a patient. These single-valued attributes are exploited by, for example, DPA. However, existing event log formats do not offer straightforward means to store time series data.

3 Evaluation

We start our evaluation by simulating the process of a container transport example adapted from [2] with an exact knowledge of the (complex) decision rules. After that we analyzed the log by integrating recurring measurement data using the proposed $DPA^{TimeSeries}$ method. In each iteration of the $DPA^{TimeSeries}$ we can compare the found decision rules with the original ones.

For the generation of process log data as well as time series data produced by recurring events within the iterations we implemented the simulation environment $DPA^{TimeSeries}_{Sim}$. Using a programming language like Java instead of a model interpreting tool like CPN-Tools [3] for simulation purpose gives us the flexibility to implement more complex rules. The time series data was integrated into the event data in various ways and exported in the log file format MXML to be used in ProM 5.2. Additionally, the time series data were exported in a simple CSV file to be used for data mining independent of the ProM framework. We used various mining algorithms from the ProM 5.2 framework to mine the models we used as a basis for DPA.

The basic idea of the container transport example is that some temperature-sensitive cargo is moved, implying that there is some temperature threshold not to be exceeded during the handling; otherwise, if this threshold is violated for a certain duration, the carriage is interrupted, and the transporting vehicle returns to its home base. Apparently, the decision whether to continue or interrupt the carriage depends on the monitored cargo temperature, measured by some sensor, for instance every 10 minutes as long as the vehicle moves towards its destination.

We now start the first iteration of the $DPA^{TimeSeries}$ method – based on this description of the process – with a simple simulation to obtain a first data set. 100 process instances are generated synthetically with up to 12 temperature measurements, such that in 30% of the cases the preset temperature threshold of 38°C is exceeded at least twice consecutively – in which case the carriage has to interrupt – whereas in 20% of the cases the threshold value is exceeded once at a time only, and in the remaining 50% of the process instances the threshold

value is not overshot at all; that is, in 70% of the process instances the haulage continues until the destination is reached.

Using this data and the alpha algorithm of ProM 5.2[1] we develop the model shown in Fig. 3 (first model). We define a new analysis path for a better understanding of the decision of interrupting the carriage or not and identify that the temperature monitoring may be a useful candidate for a decision mining activity. Using the monitoring data as additional attribute and the approach of Log Enrichment (cf. Sect. 2) we attach the sequence of temperature observations to an "On the Way" event, after which the activities "Unload at Destination" (successful carriage) or "Return to Parking Lot" (interruption) commence.

A straight-forward application of the ProM 5.2 plug-in for DPA uses decision trees to identify attribute-value clauses underlying the branching of the process as shown in Fig. 3 (first model, shaded area). In order to apply this automatic procedure a data-preparing step is in place as the added time series in one attribute cannot be interpreted by the DPA. As the procedure would always refer to the latest (temperature) measurement available, an attribute indicating the most recent temperature observation at the time of branching was defined. In the following new iteration of the $DPA^{TimeSeries}$ method, DPA is able to classify all of the "Return to Parking Lot" instances correctly. However, due to the fact that the event of overshot temperature occurs at differing times DPA cannot infer a correct decision rule, because, for 20% of the instances taking the "Unload at Destination" branch, the overshot temperature condition is also met.

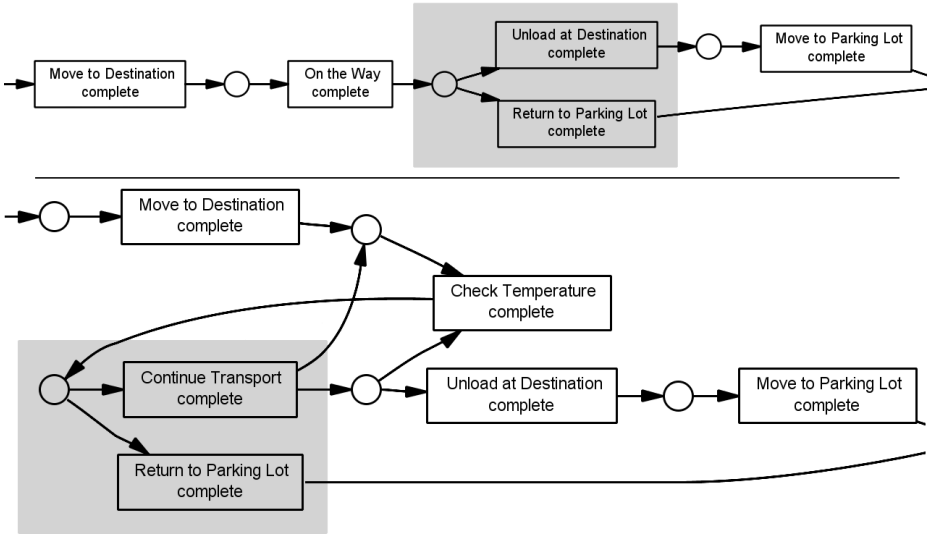

Fig. 3. Derived models, based on log enrichment and log extension (using ProM 5.2)

[1] `http://promtools.org/prom5/`

When starting a new iteration of the $DPA^{TimeSeries}$ method by adding all 12 (possible) measurements as individual process environment attributes (thus, however, losing the temporal *ordering* of the temperature information), DPA generates a fairly complex classification of cases able to classify 99 of the instances correctly anyway, but the tree is over-fitted to the data and fails to detect the proper decision criterion. The decision tree does not take into account the temporal ordering of the observations and is only applicable if all measurements are available. But even in that case it has only poor predictive power.

For the next iteration of the $DPA^{TimeSeries}$ method, we replaced the singular "On the Way" event of the process model, with all the temperature data attached, by a couple of recurrent activities, viz. "Check Transport" and "Continue Transport". This time, "Check Transport" events carry one temperature observation at a time, generating a recurrent measurement of the temperature attributes as defined above. This way we changed from Log Enrichment to Log Extension. As apparent from Fig. 3 (second model, shaded area), the entailed activity loop has been process-mined correctly.

Running DPA this time, for each of the attributes, the very same classification is obtained, but with entirely different evaluation output. First of all, amazingly, the number of process instances increases erroneously to 130; this happens because, within a process instance, only the first of recurring events is used for subsequent decision analysis [4]: hence, in all of the 100 instances, the decision after the first temperature measurement (that is, the first occurrence within the loop) branches to "Continue Transport", and just 30 instances – later in the process – "Return to Parking Lot" at all. A closer look at the log data unveils that, in 10 of the instances, the first temperature observation, respectively, exceeds the threshold value – which explains the 10 instances classified wrong.

We conclude, modeling the process in either approach cannot resolve the shortcoming of representing recurrent measurements (generated through process *loops*; [4]) of attributes for DPA, as there is no way to preserve the temporal structure of these measurements properly.

We now develop a new process view for the next iteration of the $DPA^{TimeSeries}$ method, which concentrates on the process instances and their decisions whether to return or not. For this view we use the monitoring data now as main source. This leads to an analysis model for classification of time series data. Because of regular structure of monitoring time we stick to Separation of Data and keep all 12 measurements as vector of attributes, but understand it as regular time series. Accordingly to the time series understanding we start with an analysis of the trend behavior and use parallel coordinate plots in R for the visualization of the groups. The results are shown in Fig. 4. While the critical plot (left side) only shows sharp single tops we can see clearly that the return plot (right side) has high plateaus leading immediately to the conjecture that the decision about return to parking lot depends on the duration of temperature above a threshold. From a more detailed investigation with visual data mining tools we can determine the rule: the critical event is that the temperature remain above threshold for two consecutive events.

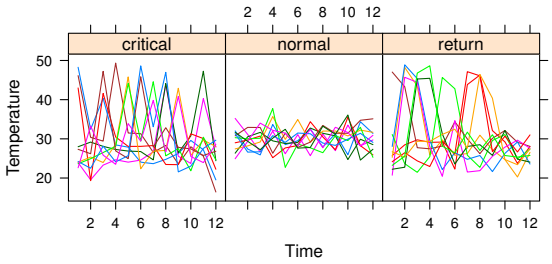

Fig. 4. Parallel Coordinate Plot of the all cases

One alternative now would be – using Log Extension again – to define a new event for the process which is defined as: "First occurrence of two consecutive measurements above the threshold". DPA would – that way – be able to identify this attribute as decisive.

We also applied different other classification methods for the data. It turned out that with Boosting and Support Vector Machines we obtained better results for the error rates in case of cross validation than with decision trees. But the results are not easy to interpret in application.

With that data understanding we now produce two aggregated data attributes: (i) a boolean attribute *temperatureThresholdViolation* indicating that the threshold we found using data mining was violated in two consecutive measurements; (ii) a numeric attribute *temperatureThresholdViolationCount* counting the number of these violations, bypassing the problem of loosing the temporal information. This way there is no need for the recurring events with single measurements and therefore we can again make use of the DPA by means of Log Enrichment using the two new aggregated attributes.

We start a new iteration of the $DPA^{TimeSeries}$ method with the augmented attribute set in the ProM environment and find with standard DPA 100% of cases are correctly classified.

4 Related Work

An integrated analysis of processes and data is provided by DPA [1], [4]. In [5], DPA was improved and generalized using algebraically-oriented procedures for finding complex decision rules with more than one variable. By contrast, the $DPA^{TimeSeries}$ method aims at finding new rules using statistically-oriented empirical methods, augmenting the space of possible decision functions with attributes through a data-driven search among empirical models. [6] overcomes other difficulties of DPA like invisible transitions and therefore certain kinds of loops within the process model or deviating behavior by control-flow alignment. Our approach differs from that in dealing with time series data and therefore recurring events that might not be found within existing log files. Our approach also resolves problems with loops through extending DPA with data mining tech-

niques to identify aggregation value attributes and defining new events within the business processes these attributes can be attached to. Another interesting approach is [7] that addresses the clustering of health care processes. The $DPA^{TimeSeries}$, by contrast, focuses on the classification of temporal data occuring in connection with processes.

Log preparation tools cover the extraction and integration of data from different sources as well as data quality improvement, e.g., [8, 9]. Log enrichment is one possibility to deal with the latter, e.g. in [10] it is proposed to make more complex time data usable.

5 Conclusions

In this paper, we proposed the $DPA^{TimeSeries}$ method for analyzing time series data and process logs by a combined and iterative application of process and data mining techniques. For equipping and analyzing the logs with time series data, we discussed the possibilities of log enrichment and extension as well as of keeping log and time series data in a separated way. The $DPA^{TimeSeries}$ method is implemented and evaluated based on use case from the logistics domain.

References

1. Rozinat, A., van der Aalst, W.: Decision mining in ProM. In: Business Process Management. (2006) 420–425
2. Rinderle, S., Bassil, S., Reichert, M.: A framework for semantic recovery strategies in case of process activity failures. In: ICEIS (1). (2006) 136–143
3. Jensen, K., Kristensen, L.M., Wells, L.: Coloured petri nets and CPN tools for modelling and validation of concurrent systems. Int. J. Softw. Tools Technol. Transf. **9**(3) (2007) 213–254
4. Rozinat, A., van der Aalst, W.: Decision mining in business processes. Technical report (2006)
5. de Leoni, M., Dumas, M., Garcia-Banuelos, L.: Discovering branching conditions from business process execution logs. In: Fundamental Approaches to Software Engineering. (2013) 114–129
6. de Leoni, M., van der Aalst, W.M.: Data-aware process mining: discovering decisions in processes using alignments. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, ACM (2013) 1454–1461
7. Rebuge, Á., Ferreira, D.R.: Business process analysis in healthcare environments: A methodology based on process mining. Information Systems **37**(2) (2012) 99–116
8. Rodriguez, C., Engel, R., Kostoska, G., Daniel, F., Casati, F., Aimar, M.: Eventifier: Extracting process execution logs from operational databases. In: BPM 2012 Demo Track. (2012)
9. Nooijen, E.H.J., Dongen, B.F.v., Fahland, D.: Automatic discovery of data-centric and artifact-centric processes. In: Business Process Management Workshops. (2013) 316–327
10. Dunkl, R.: Data improvement to enable process mining on integrated non-log data sources. In Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A., eds.: Computer Aided Systems Theory - EUROCAST 2013. Volume 8111 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 491–498