A Method for Analyzing Time Series Data in Process Mining: Application and Extension of Decision Point Analysis

Reinhold Dunkl(✉), Stefanie Rinderle-Ma, Wilfried Grossmann, and Karl Anton Fröschl

Faculty of Computer Science, University of Vienna, Vienna, Austria
{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 DPATS method as an iterative process for exploiting time series data by combining process and data mining techniques. The latter ranges from visual mining to temporal data mining techniques such as dynamic time warping and response feature analysis. 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 DPATSSim to produce log files and time series data. The DPATS method is evaluated based on application scenarios from the logistics and medical domain.

Keywords: Process mining · Decision mining · Data mining · Time series data

1 Introduction

The interest of research and practice in process mining has dramatically increased during the last years. Process mining has different objectives, ranging from discovering process models from event log data to comparing events logs and existing process models (conformance checking) [1]. Event logs can be described as time-stamped event data (so-called log files) gathered from or produced by process instances executed in some process environment. Example event logs might stem from higher education processes [2] or skin cancer treatment processes [3].

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.

© Springer International Publishing Switzerland 2015
S. Nurcan and E. Pimenidis (Eds.): CAiSE Forum 2014, LNBIP 204, pp. 68–84, 2015.
DOI: 10.1007/978-3-319-19270-3_5

This paper focuses on process discovery. So far, process discovery techniques have emphasized the control flow, i.e., discovering the process activities and the control structures of the process models from the event logs. The minimum information required for control flow discovery is information about the process task connected with the event (`WorkflowModelElement`) and a `Timestamp` as contained in the following event log fragment (in MXML format [4]).

```
<AuditTrailEntry>
   <WorkflowModelElement>Move to D</WorkflowModelElement>
   ...
   <Timestamp>2013-02-27T10:29:06.404+01:00</Timestamp>
</AuditTrailEntry>
```

An extension towards the branching logic of processes is provided by Decision Point Analysis (DPA) [5]. 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 branching in process models. Basically, DPA works as follows: in a first step, the underlying process model is discovered based on the event log entries. If the resulting process model contains decision points (and additionally data relevant for the decisions is present in the event logs), the corresponding decision rules are determined using decision trees. Assuming that task `Move to D` marks a decision point in the process, the following log fragment could be basis for DPA:

```
<AuditTrailEntry>
   <WorkflowModelElement>Move to D</WorkflowModelElement>
   <Data>
      Attribute name="ContainerTemperature">37.2</Attribute>
   </Data>
   <Timestamp>2013-02-27T10:29:06.404+01:00</Timestamp>
   ...
</AuditTrailEntry>
```

Figure 1 depicts the container transportation example associated with the two log fragments above. It is based on the real-world case provided in [6], where some temperature-sensitive cargo is transported and cargo temperature is measured repeatedly. On the left, the application of DPA [5] is illustrated: depending on the temperature value for each transport monitored (at task `Move to D`), DPA concludes that for a temperature over 37°C, the vehicle has to return to its home base. Otherwise, it unloads the goods at the destination. As this example illustrates, 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.

However, in many application domains, not only single values of data attributes are collected, but *time series data*. Examples comprise the logistics, health care, and the manufacturing domain where container temperature, blood pressure, or sensor data are measured in a continuous way. Such continuously updated data [7] can be stored as time series data. The main question of this

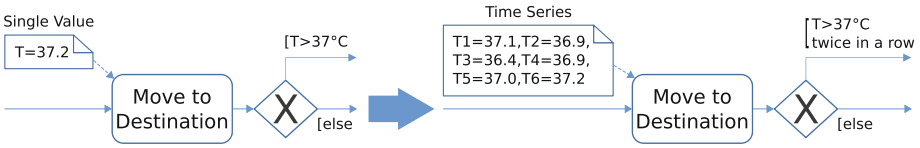

Fig. 1. Process applications with time series data

paper is whether it is possible to extend DPA from single value analysis to adequately incorporating time series data. By doing so, it shall also become possible to determine more complex decision rules than "x OP value", for example, "temperature exceeds a certain threshold for a given time frame" (cf. right side in Fig. 1).

In this paper, we will present such an extension of DPA by means of a novel method DPATS 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. Note that this paper is an extended version of [8]. The general method has been detailed and extended, e.g., incorporating further techniques such as time warping and a second evaluation example for multivariate data has been added.

Section 2 discusses different ways how time series attributes can affect the business process. This leads to two separable challenges within the problem setting, data preparation itself as well as data and visual mining aspects and how those integrate in the overall DPATS method (Sect. 3). The method uses as essential part approaches towards temporal data mining. The ensuing process mining method is evaluated based on a real-world examples of process analysis (Sect. 4). After reflecting our contribution against the state of the art in process, data, and visual mining (Sect. 5), some concluding remarks (Sect. 6) finish this presentation.

2 Business Processes and Time Series Attributes

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. In case of attributes defined as time series the situation is more complex because the measurement of the time series defines an additional process. This so called *measurement process* produces time series attributes which might cause effects on the execution of the business process of interest. The following two types of effects can be distinguished:

– *Separable effects* control the decision about activities in the business process. Depending on the results of the measurement process a decision about the execution of different activities in the business process is made. Take as example a treatment process in medical applications. The decision about different options for further therapy usually depends on the status of the patient

documented by time series for some medical parameters such as blood pressure recorded in the past. Looking at the development of these parameters, the doctor decides about future therapy, for example, choice of medication.
– *Intermingled effects* control the execution of activities in the business process. This means that the results of the measurement process influence the execution of the activities in business process. Take as example the transportation process of a container with temperature-sensitive cargo. During the transport, a measurement process continuously records the temperature of the container. In case of abnormal behavior of the temperature, the affected activity `transport` is interrupted (see e.g., [7]) and a new activity called the `return` starts.

Usage of time series data in modeling business processes depends not only on the above described types of effects, but also on the structure of the time series data. The simple case is based on the assumption that the effects of the time series data on the process depend only on the actual value of the time series. We call this effects *Markov-like effects*. The rationale behind this denotation is that the effect can be compared with the Markov property in processes, which states that all information about the process at a certain time is captured in the actual value. In the examples mentioned above, this would be the actual blood pressure of the patient or the actual temperature in the container.

In many applications such an assumption cannot be justified. For example, in case of container transport a short period of high temperature is not critical and return is only advisable in case of high temperature over a certain time period. Such cases need additional analysis about properties of the time series. We call such effects *non-Markov-like effects*.

The above elaboration makes clear that usage of time series data in DPA needs additional considerations. Basically, DPA is designed for applications with non temporal attributes. In case of temporal attributes, it can be applied, but only provided that the time series measurements generate separated effects and the effects depend only on the actual measurements, i.e., have a Markov like property. It is the aim of DPATS to develop an analysis framework which can handle also more complex cases. As explained in the following section, the framework is based on a combination of methods for process analysis, methods for temporal data mining, and appropriate representation of data.

3 DPATS Method

The DPATS method is illustrated in Fig. 2 and describes the process of analyzing time series data as basis for decision point analysis in business processes. The proposed method does not depend on a particular domain. The only precondition for its application is the existence of time series data collected at decision points in the process. In the following, we will motivate the design of the DPATS method by shortly explaining its steps. A more detailed discussion is following in the subsequent sections.

The DPATS method constitutes and extension of DPA which consists of the steps *classification* and *data mining* [5]. In order to be able to consider time series data, the DPATS method has to introduce a prior step of *data preparation* as the storage of time series data is not foreseen in existing log formats (cf. XES as standard log format [4]). The data preparation step is elaborated in Sect. 3.1.

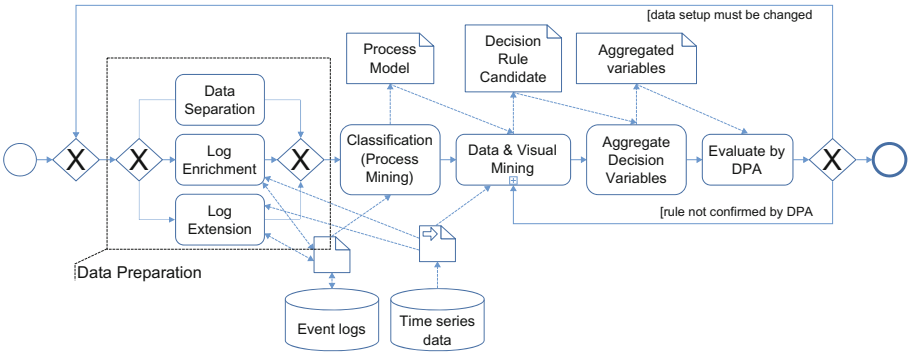

Fig. 2. DPATS method (in BPMN notation)

As we operate on time series data, various techniques for temporal data mining can be used in the *Data & Visual Mining* step (cf. Fig. 3 and Sect. 3.2) in order to explain the decisions in the observed process instances. This step follows the model set out by Keim et al. [9] which generates knowledge (in DPATS the decision rules) based on the data and a tightly integrated data visualization and mining approach. This more experimental mode of analysis, utilizing continuously improved understanding of (perhaps not yet) available process and environment data seems more appropriate at this stage of the analysis than a mechanical brute-force exploration.

As a result of the *Data & Visual Mining* step, candidates for decision rules can be identified and transformed into aggregated variables (*Aggregate Decision Variables*). These variables can then be used to employ DPA in order to evaluate the decision rule candidate (*Evaluate by DPA*). Depending on the result, the inspection by both, data and temporal data 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. For that reason as well as for the possible change of decision rule candidates, the *Evaluate by DPA* step can differ significantly from one iteration to the next one.

3.1 Data Preparation

As existing log files do not offer means to capture time series data, the question is how to provide such data for DPA. Intuitively, time series data can be offered within or outside the event logs. The first option is followed by *data separation*.

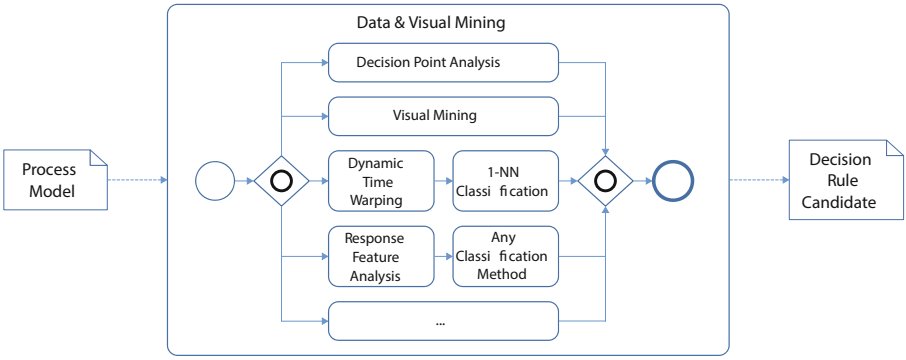

Fig. 3. Subprocess data and visual mining of the DPATS method

Integrating time series data within logs can be done in two ways: either by introducing a time series variable (enrichment) or simulating the production of time series data by extending the log with an (artificial) recurring measurement event. In detail:

1. *Data Separation:* 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.
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 first the pros and cons of each option from a technical point of view and afterwards the application in DPATS.

Data separation 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 affected 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 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 changes the log effectively, but makes format extensions and extra files dispensable.

In practical application, the choice between the options depends on the *Data & Visual Mining* techniques applied in DPATS (cf. Fig. 3), on the possible effects of the time series on the process, and on the structure of the time series. In the first step of DPATS, data separation is usually a good choice. After performing the *Classification (Process Mining)* step of DPATS, one can identify the decision points in the process and obtain a first understanding whether the time series has separable effects or intermingled effects (cf. Sect. 2).

In case of separable effects, the best choice for further analysis is log enrichment. Whether the log enrichment allows immediate application of standard DPA depends on the structure of the time series. If the effects of the time series are Markov-like (cf. Sect. 2), one can immediately use the actual values of the time series at the decision points for DPA in the *Data & Visual Mining* step. If the time series causes non-Markov-like effects the *Data & Visual Mining* step comprises different applications of temporal mining for finding candidates for the decision rule base on separated data.

In case of intermingled effects, a good starting point for the *Data & Visual Mining* step is what we call *post-mortem analysis*; i.e., we assume that all measured values of the observed process instances are known and an analysis after observation of the entire process instance is conducted. All values of the time series are considered as one entity and are input for extraction of decision rules candidates. By doing so the intermingled effect becomes a separable effect at the decision point at the price that the decision uses probably not only actual values of the time series, but also future values. The candidates for decision variables are usually derived by temporal mining techniques for separated data (step *Aggregate Decision Variables*). The rule can be evaluated afterwards. Note that a post-mortem analysis can only be used for decision rules of completed process instances, but not as decision rules for new process instances at runtime.

For development of decision rules at runtime, the best choice is usually to envisage a model with log extension. This means that whenever a new value of the time series occurs one has to make a decision about interruption of the involved activities. In best case, the decision variables for the different decision points can be derived in step *Aggregate Decision Variables* from the decision rules of the post-mortem analysis by applying the rule to the segment of the time series from the beginning up to the actual observation time. In more complicated cases new analyses for each possible decision point may be necessary.

3.2 Temporal and Visual Data Mining

Two frequently used approaches towards classification and clustering of temporal data are based on *dynamic time warping* and on *response feature analysis* [10].

The basic idea behind dynamic time warping is that observations of time series may have the same structural characteristics, for example, number of peaks and relation between peaks, but the position may be blurred due to external effects. In order to find the similarity between such time series, dynamic time warping stretches and compresses the time scale of the series in such a way that the distance between the time series is minimized. A side condition for these

transformations is that the order of the measurements in both series is preserved. The approach can be applied for time series of different length. Moreover, exact information of the time stamps in the observations is not necessary. Details may be found in for example in [11].

As a result of dynamic time warping, one obtains a matrix showing the similarity between different time series. This similarity matrix can be used later on for classification or clustering. In case of clustering, any method based on distances can be applied, in case of classification the straightforward approach is using the similarity matrix with 1-NN classification. As reported in [10], this rather simple classification method has been shown successful in many applications.

Response feature analysis reduces the problem of clustering and classification of time series to problems for non-temporal data. Response features can be obtained in different ways, depending on the problem. In business applications, typical response features can be based on defining regression or time series models for each observed time series. As a result, one obtains a number of time-independent parameters. Another approach is to look at characteristic of the frequency distribution of the individual time series, for example, means, variances, or quantiles of the values of the time series. As a third method, one can find structural properties of the time series, for example, change points in the behavior of the time series. The response features can be used afterwards as input for classical classification and clustering algorithms.

Dynamic time warping and response feature analysis transform the problem of classification and clustering of time series into problems without temporal structure. Another interesting and more experimental approach is *visual data mining*. By plotting different time series it may become possible to detect interesting features of the time series which allow also interpretation in terms of the problem. In Sect. 4, we will show applications of all three approaches.

4 Evaluation

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 DPATSSim. Using a programming language like Java instead of a model interpreting tool like CPN-Tools [12] 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 process models as a basis for DPATS. To avoid misinterpretation of event names the keyword “complete” – signifying a point in time event in ProM – was removed from the screen shots (Figs. 4 and 6). Both Figues show the basic mined modells depicting the scenarios as well as the decsison points found, with gray background the decision point that was selected while the screenshot was taken.

After that we analyzed the log by integrating recurring measurement data using the proposed DPATS method and compared the found decision rules with the original ones.

The rationale behind the selection of the scenarios is to illustrate the analysis of time series data. The first example was chosen simple with one variable as this is sufficient to explain the challenges of time series data. The second example illustrates the more complex case, i.e., the multivariate case. The choice is independent of the application domain.

4.1 Univariate Scenario: Container Transportation

We start our evaluation by simulating the process of a container transport example adapted from [6] with exact knowledge of the (complex) decision rules. The basic idea of the container transport example is that a 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 min as long as the vehicle moves towards its destination. 100 process instances are generated synthetically with up to 12 temperature measurements, such that in 30 % of the cases the pre-set temperature threshold of 37°C is exceeded at least twice consecutively – in which case the carriage has to interrupt – whereas in 20 % of the cases the threshold value exceeds 37°C only at one time point. In the remaining 50 % of the process instances the threshold value is not overshot at all. Hence, in 70 % of the process instances the haulage continues until the destination is reached. For the instances where the transport was interrupted the time series are usually shorter taking into account only time for reaching the parking lot.

In the first step we decide to start with separated data and use in the second step for finding the first process model only the transportation events without consideration of the time series data. Using the alpha algorithm of ProM 5.2[1] we develop the model shown in Fig. 4 (first model). Obviously the decision point generates an interruption of the transportation activity. We identify the measurement process of temperature as useful candidate for the decision mining activity. This process has intermingled effects on the transportation process. Hence, we decide to use in the third step post-mortem analysis for learning the effect of the time series. Because we have to consider her the complete time series it is not reasonable to think about a Markov-like effect on the transport process and we use again separated data for temporal data mining. The time series data are augmented with the observed decisions “normal” and “return”.

We start the temporal mining activity with dynamic time warping for the observed time series. Using 1-NN classification for the time series we obtain a correct classification of all cases which return to the parking lot but 11 cases

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

which completed the transport were wrongly classified as cases which had to return to the parking lot. In order to understand the reasons for misclassification we decide to use visual mining using plots of the time series. The plots of correctly classified and misclassified cases are shown in Fig. 5. The misclassified cases are labelled as "normal-critical". From visual inspection it is quite obvious that the normal-critical cases are characterized by isolated spikes with high temperature, whereas all return cases show high temperature for at least two consecutive measurements. Hence, we conclude that a decision rule candidate could be: measurement at two consecutive measurement time point is above 37°C. Using this rule in the decision point of the post-mortem analysis we obtain 100 % correctly classified instances.

Now we start with a new analysis round and decide to use the time series data with log extension. In that case the decision rule from post-mortem analysis can be immediately transferred into a rule at runtime by the attribute "minimum of the actual and previous measurement". The decision rule is whether the attribute is above 37°C or not. Evaluating this rule in the process model with log extension shown in the lower part of Fig. 4 gives us a correct classification.

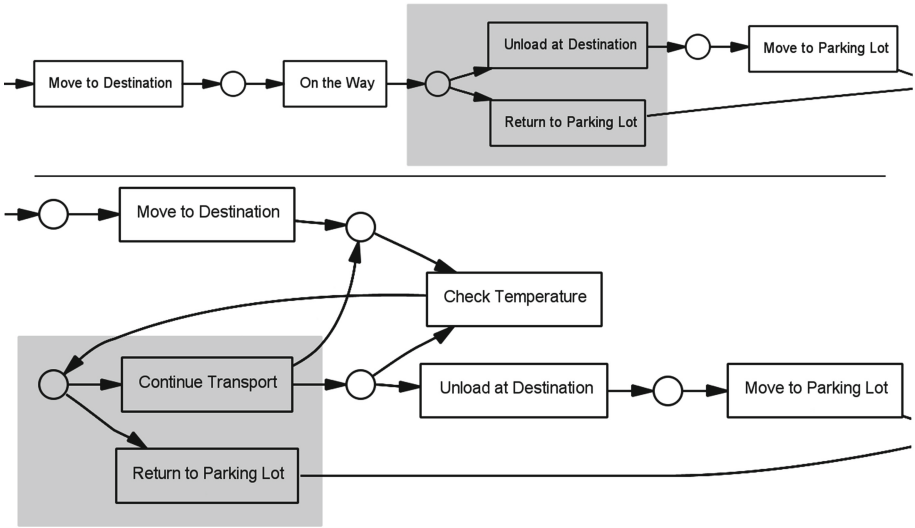

Fig. 4. Univariate scenario: derived models, based on log enrichment and log extension (using ProM 5.2)

4.2 Multivariate Scenario: Hypertension in Pregnancy

The second example we want to use for evaluating the DPATS method is from the medical field where certain diagnostic values are measured recurrently. Elevated blood pressure during the pregnancy is an important sign for illnesses like preeclampsia or general hypertension. Additionally, the weight and the proteinuria are measured as second indicators. In this case blood pressure and

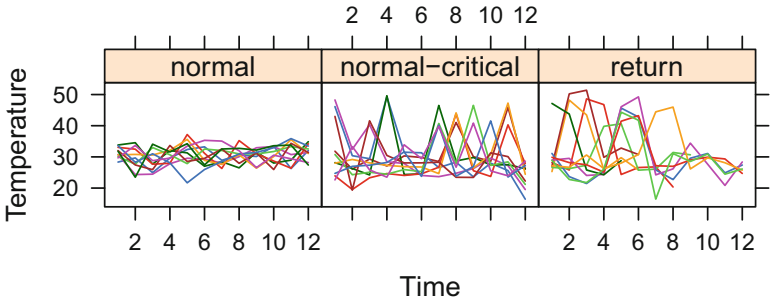

Fig. 5. Univariate scenario: correctly and misclassified time series

weight are measured by the patient herself, additionally to that proteinuria at the patient's regular visit at the doctor. We have to consider four data elements *BodyWeight*, *SystolicBloodPressure*, *DiastolicBloodPressure* and *Proteinuria* during the pregnancy. If the *SystolicBloodPressure* reaches more that 160 mmHg or the *DiastolicBloodPressure* more than 100 mmHg, the patient has to be admitted to a hospital. Additional criteria are a *SystolicBloodPressure* over 140 mmHg or a *DiastolicBloodPressure* over 90 mmHg in combination with *Proteinuria* over 0.3 g/l or a more than 1 kg weight gain in a week. Epiphenomenon like sight disorder, cerebral symptoms and pain in the epigastrum are also criteria that leads to admit the patient to a hospital but are skipped to simplify the example.

Like in the container scenario, we produced synthetic data using a simulation tool. 300 cases were generated where in 27 cases the patient had to be admitted to a hospital. The recurring measurements start after the 20th week of gestation until the patient is hospitalized because of the violation of one of the three rules or giving birth. The measurements are recurring but, opposed to the container scenario, missing values can occur. Randomly distributed, some patients are more often measuring than others, some patients weigh themselves every day while measuring their blood pressure much less frequently and the other way round. If the blood pressure is elevated (higher than 140/90) through four days, the regular check for proteinuria is brought forward. After a proteinuria check, weight and blood pressure is always checked, too. Each check is represented by an event with attributes attached. We started with data using recurrent measurement events.

In a first attempt of process mining using the alpha and heuristic miner no fitting model (according to our understanding) was identified. Parallel checking of three different values in a loop overcharges these algorithms. We changed to a genetic mining algorithm [13] and got a fitting model (cf. Fig. 6 as Petri Net) that can be used for DPA. DPA was not able to find any rules for any of the three relevant decision points.

For a new iteration of the DPATS method, we inserted a decision event comprising all four attributes thus consolidating the model's decision point we

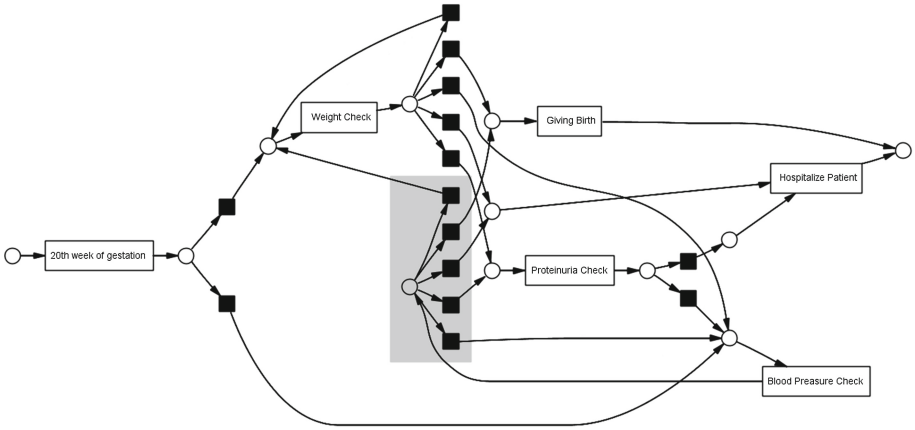

Fig. 6. Multivariate scenario: derived petri net model as basis for DPA

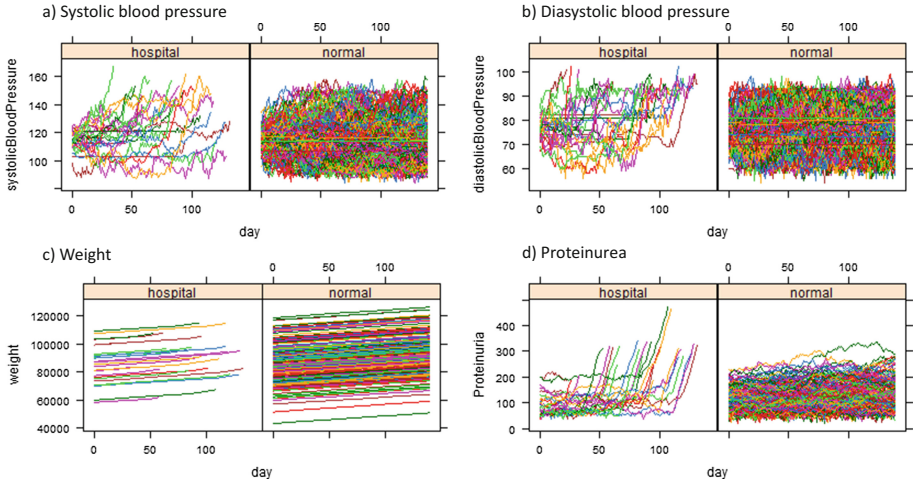

Fig. 7. Multivariate scenario: evaluation of variables a) - d)

want to analyze. DPA was not able to find any rule, most likely due to the already stated fact that loops present a challenge to DPA.

After that we turned to data mining to improve our data understanding. For the next iteration of the DPATS method we defined a second analysis round with a data mining goal for learning the decision. Because the time series have different length and probably also monitoring times we treat them as vector of time series data, i.e. four measurements at each time point, and group the data according to the case ID characterizing the subject and classify the entire series as normal or hospital. We started with trend analysis of the series and produce plots in the four variables as shown in Fig. 7(a) – (d). Plots (a) – (c)

lead to the conjecture that going to hospital may be explained by increase in at least one of the attributes systolic, diastolic and proteinuria. At least for proteinuria the results are striking. The plots indicate that there is a change point in the proteinuria series in case of hospitalization. With respect to weight such a conjecture is not so obvious. Hence we have to think about more complex models. One option is modeling the intercept of weight increase as a personal parameter and the slope as a group specific parameter. Another option is to transform the time series.

Using the second approach we learn that first differences are not sufficient, mainly because of random fluctuation of the measurements. Hence, we look at weight increase in longer time periods and find that a period of seven days could be a good candidate for discrimination of the two groups from a practical point of view as decision rule for the patients. The behavior of the two groups with respect to this new attribute is shown in Fig. 8. Also in this case a change point in the behavior of the series seems to be a good indicator for the groups.

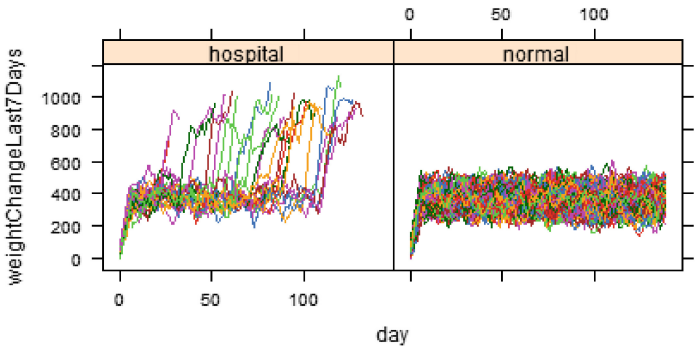

Fig. 8. Multivariate scenario: weight change of the last 7 days

Putting all these results together, we attempt to formulate a classification model with the following response features attributes (i) 0.95 quantiles for weight, proteinuria, systolic, and diastolic blood pressure; (ii) change points for weight change in the last seven days, and proteinuria; (iii) slopes of the weight series. We used not the maximum but the 0.95 quantile for getting more robustness against single extreme values. For these variables we applied tree classification, support vector machines and AdaBoost. All classifications were done with R and 10 fold cross validation was used. Classification trees with 10 fold cross validation generated a simple decision tree using only the change points of the variable `weightChangeLast7Days`. Alternatives for the splits were the slopes of the weight variable. All patients with hospitalization were correctly classified. From the 275 time series of persons with normal pregnancy three were misclassified. An application of boosting and support vector machines produced better results. However, explanation of the decision from practical point of view is difficult. Note that in this case we are not in position to learn the classification

rule applied in data generation. In particular, the influence of blood pressure measurements on the decision is not visible, mainly because there are no cases with only blood pressure effects in the data. Also missing values in blood pressure hamper the use of that series in the analysis.

As a last iteration we now can use all the gathered information from our data mining approaches and translate that into aggregated variables to be evaluated with DPA. For this multivariate application it is possible – but was not necessary – to aggregate all variables into one decision variable for evaluation like in the transportation example. But in this case it is advantageous to keep variables separated as different rules are dependent on combinations of different variables. As shown before not all rules could be found with the data set at hand, which was also reflected by the last evaluation step with DPA.

5 Related Work

The DPATS method can be located at the interface of three areas, i.e., process mining (e.g., [1]), data mining (e.g., [10]), and visual mining (e.g., [9]) as depicted in Fig. 9. Clearly, for all these areas several approaches exist, also at the interfaces between the different areas. A combination of process mining and visual mining is proposed by [14] where mined process models can be compared using difference graphs as a visual means. DPATS is to the best of our knowledge the first method to combine all three areas in order to be able to tackle the analysis of time series data in the context of process mining.

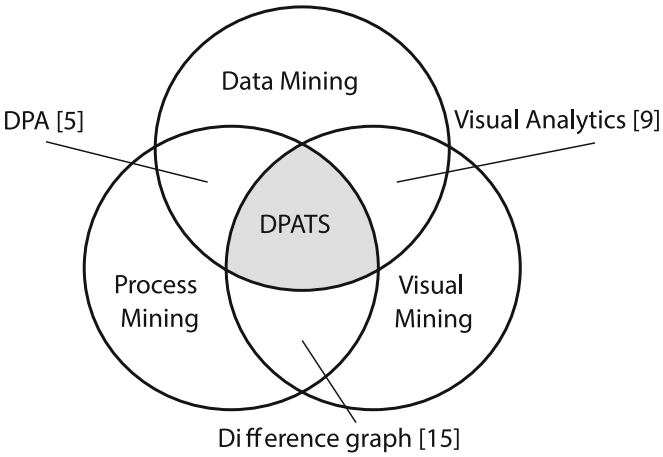

Fig. 9. Related research areas

DPA [5] is based on the combined application of process and data mining, more precisely, decision trees. In [15], DPA was improved and generalized using

algebraically-oriented procedures for finding complex decision rules with more than one variable. By contrast, the DPATS 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. [16] 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 techniques to identify aggregation value attributes and defining new events within the business processes these attributes can be attached to. Another interesting approach is [17] that addresses the clustering of health care processes. The DPATS method, by contrast, focuses on the classification of temporal data occurring 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., [18,19]. Log enrichment is one possibility to deal with the latter, e.g. in [20] it is proposed to make more complex time data usable.

6 Conclusions

In this paper, we proposed the DPATS 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. Log preparation might be more challenging with real world data, particularly at the presence of complex logs that are further extended by recurring measurement events reflecting the production of time series data. Then the aspect of analyzing expressive constructs, i.e., time series data, has to be balanced with complexity of the analysis. We will expand our studies in this direction by applying DPATS in real-world settings. Candidates are the EBMC^2 project (`ebmc2.univie.ac,at`) on patient treatment or FP7 project ADVENTURE (http://www.fp7-adventure.eu/) from the manufacturing domain.

The DPATS method features data and visual mining techniques such as dynamic time warping as main analysis step and is implemented and evaluated based on use cases from the logistic and medical domain. Several future research directions such as the inclusion of time sequences and application of the DPATS method for monitoring process execution during runtime have been discussed. We will follow up along this line of research.

References

1. van der Aalst, W.M.P.: Process Mining – Discovery, Conformance and Enhancement of Business Processes. Springer, Ber (2011)

2. Ly, L.T., Indiono, C., Mangler, J., Rinderle-Ma, S.: Data transformation and semantic log purging for process mining. In: Ralyté, J., Franch, X., Brinkkemper, S., Wrycza, S. (eds.) CAiSE 2012. LNCS, vol. 7328, pp. 238–253. Springer, Heidelberg (2012)
3. Binder, M., Dorda, W., Duftschmid, G., Dunkl, R., Fröschl, K.A., Gall, W., Grossmann, W., Harmankaya, K., Hronsky, M., Rinderle-Ma, S., Rinner, C., Weber, S.: On analyzing process compliance in skin cancer treatment: an experience report from the evidence-based medical compliance cluster (EBMC^2). In: Brinkkemper, S., Franch, X., Ralyté, J., Wrycza, S. (eds.) CAiSE 2012. LNCS, vol. 7328, pp. 398–413. Springer, Heidelberg (2012)
4. Verbeek, H.M.W., Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: XES, XESame, and ProM 6. In: Soffer, P., Proper, E. (eds.) CAiSE Forum 2010. LNBIP, vol. 72, pp. 60–75. Springer, Heidelberg (2011)
5. Rozinat, A., van der Aalst, W.M.P.: Decision mining in ProM. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 420–425. Springer, Heidelberg (2006)
6. Rinderle, S., Bassil, S., Reichert, M.: A framework for semantic recovery strategies in case of process activity failures. In: ICEIS 2006, vol. 1, pp. 136–143 (2006)
7. Bassil, S., Rinderle, S., Keller, R., Kropf, P., Reichert, M.: Preserving the context of interrupted business process activities. In: Chen, C.-S., Filipe, J., Seruca, I., Cordeiro, J. (eds.) Enterprise Information Systems VII, pp. 149–156. Springer, The Netherlands (2006)
8. Dunkl, R., Rinderle-Ma, S., Grossmann, W., Fröschl, K.A.: Decision point analysis of time series data in process-aware information systems. In: CAISE Forum 2014, pp. 33–40. ceur-ws.org (2014)
9. Keim, D.A., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., Melançon, G.: Visual analytics: definition, process, and challenges. In: Kerren, A., Stasko, J.T., Fekete, J.-D., North, C. (eds.) Information Visualization. LNCS, vol. 4950, pp. 154–175. Springer, Heidelberg (2008)
10. Mitsa, T.: Temporal Data Mining. CRC Press, Boca Raton (2010)
11. Giorgino, T.: Computing and visualizing dynamic time warping alignments in R: The dtw package. J. Stat. Softw. **31**(7), 1–24 (2009)
12. 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. Tran sf. **9**(3), 213–254 (2007)
13. van der Aalst, W.M.P., de Medellin, A.K.A., Weijters, A.J.M.M.T.: Genetic process mining. In: Cardozo, G., Darondeau, P. (eds.) ICAHN 2005. LNCS, vol. 3536, pp. 48–69. Springer, Heidelberg (2005)
14. Kriglstein, S., Wallner, G., Rinderle-Ma, S.: A visualization approach for difference analysis of process models and instance traffic. In: Daniel, F., Wang, J., Weber, B. (eds.) BPM 2013. LNCS, vol. 8094, pp. 219–226. Springer, Heidelberg (2013)
15. de Leoni, M., Dumas, M., García-Bañuelos, L.: Discovering branching conditions from business process execution logs. In: Cortellessa, V., Varró, D. (eds.) FASE 2013 (ETAPS 2013). LNCS, vol. 7793, pp. 114–129. Springer, Heidelberg (2013)
16. 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, pp. 1454–1461. ACM (2013)
17. Rebuge, Á., Ferreira, D.R.: Business process analysis in healthcare environments: a methodology based on process mining. Inf. Syst. **37**(2), 99–116 (2012)

18. 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)
19. Nooijen, E.H.J., van Dongen, B.F., Fahland, D.: Automatic discovery of data-centric and artifact-centric processes. In: La Rosa, M., Soffer, P. (eds.) BPM Workshops 2012. LNBIP, vol. 132, pp. 316–327. Springer, Heidelberg (2013)
20. Dunkl, R.: Data improvement to enable process mining on integrated non-log data sources. In: Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A. (eds.) EUROCAST. LNCS, vol. 8111, pp. 491–498. Springer, Heidelberg (2013)