Using Regression Trees to Analyze Patterns in Business Processes

J. Nakatumba[1], J. A. Quinn[2], and W.M.P. van der Aalst[1]

[1] Eindhoven University of Technology
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands,
{jnakatum,w.m.p.v.d.aalst}@tue.nl
[2] Makerere University Kampala
P.O. Box 7062, Kampala, Uganda,
jquinn@cit.ac.ug

Abstract. Process-aware information systems typically log events related to a particular business process in the so-called event logs. The proper analysis of the resource perspective of these logs is very important because this information can be used as a basis for operational decision making. Given an event log, *process mining* techniques can be used to obtain information characterizing resource behavior. In this paper, we refer to this information as "resource patterns". The main aim of this paper is to use classical machine learning techniques to find such resource patterns. In particular, we use regression trees to analyze the effect of these patterns on performance. This is implemented as a *Regression Miner* plug-in in the ProM framework. By applying machine learning techniques in the context of business processes we can gain insights into the behavior of resources. Moreover, the analysis of these patterns can be used to improve resource allocation in process-aware information systems.

Key words: Process Mining, Regression Trees, Resource Patterns.

1 Introduction

Currently, more and more business processes are being supported by software systems that manage and execute these processes. These software systems are referred to as Process-Aware Information Systems (PAISs) [7]. PAISs record information about the different processes they support in event logs or audit trails. An event log is simply a set of traces and each trace consists of activities from start to end of the trace execution. These logs provide an excellent source of information useful for *process mining* [2]. Process Mining is a technique used for the extraction of information from event logs recorded by a PAIS. For example, based on the timestamp field of the event logs, we can extract information about how long it takes to execute activities (i.e.,service times).

In business processes today, there are a number of factors that influence the way people actually work. In our previous work [1, 3, 12], we observed that many of these factors though important, were typically ignored when building simulation models and making resource allocation decisions in PAISs. Hence there are surprising differences when comparing the behavior of humans observed when using process mining techniques and the behavior of humans assumed in simulation tools and when making resource allocation decisions.

Currently, most of the PAISs assume a very stable process and organization, and that neither of these can change over time. The resource roles defined in these systems are static and cannot vary based on e.g. workload, the time of day, day of the week. Further on, the current resource allocation rules do not consider seasonal effects or particular times of the day when work is started and yet these have an effect on resource performance. In [4] it is shown that such "second order dynamics" heavily influence performance. These are important aspects and yet are currently not considered when making resource allocation decisions.

Using process mining techniques we can discover useful information describing resource behavior from event logs. For example, in [12] we characterized the behavior of people based on the concept of *workload dependent processing speeds.* Using linear regression analysis, we learned from event logs that there is a relationship between workload and the speed at which resources work. With these and other insights into resource behavior, we can be able to provide a better analysis of resource behavior useful for building simulation models and for resource allocation in PAISs.

We introduce the concept of resource patterns and *investigate what the effect of these resource patterns is on the service times.* The patterns discussed are aspects relating to when resources start executing activities and we use *regression trees* for this analysis. Regression trees are a technique that explain and predict values for continuous numeric attributes from a set of explanatory variables [5].

Regression trees were chosen for this analysis because a) we are interested in predicting service time which is a numeric attribute, hence regression trees which are an extension of decision trees with numeric values at the leaf nodes can be easily used to represent the values for the service time. b) the different resource patterns analyzed in this paper, do not necessarily have linear relationships between them. They are aspects that influence the way people work. We are not interested in investigating the relationships between these different patterns but rather what the effect of resource patterns is on the service times. Hence, regression trees were selected for the analysis of different resource patterns and their effect on the service times. To support this analysis, we have implemented a *Regression Miner* plug-in in the context of the ProM framework [1]. The analysis and mining of the different resource patterns observed in the event logs is important for two main reasons, i.e., *analysis* and *resource allocation* purposes.

The users of the PAISs may be interested in how long it takes to execute activities given certain resource patterns. From the analysis of the information about resource behavior recorded in the logs, this can be used as a basis for providing feedback. For example, a prediction can be made about how long

[1] The approach presented in this paper is implemented in the pre-release version of the ProM-framework 2008, which can be obtained from www.processmining.org

activities will take to be executed given specific patterns. Hence the insights obtained based on the analysis of resource behavior can be used for analysis purposes.

One of the main roles of a PAIS is to facilitate the allocation and distribution of work amongst a group of resources. However, most of the allocation decisions made in the current PAISs suffer from a limited understanding of resource behavior. Based on the insights obtained through the analysis of resource patterns, this can act a basis for the decisions about which resources to allocate work in enactment software.

The remainder of the paper is organized as follows. First, we provide an overview of event logs in Section 2. Section 3 introduces regression trees and discusses their application in a simulated model. In Section 4, we describe the application of our approach to a case study. Here, we use real-life logs to validate our approach and also discuss how the information we obtain can be used for resource allocation in PAISs. Section 5 discusses related work and finally Section 6 concludes the paper.

2 Process Mining and Event logs

Most information systems (e.g. WFM and BPM systems) provide some kind of *event log* also referred to as audit trail entry or workflow log [2]. An event log is a set of events and each event is linked to a particular trace and is globally unique, i.e., the same event cannot occur twice in a log. A trace represents a particular process instance and for each trace time should be non-decreasing.

Definition 1 (Event Log and Trace) Let $\mathcal{E}$ be the event universe, i.e., the set of all possible events. For any event $e \in \mathcal{E}$, we define $\overline{e}$ as a shorthand for the time of occurrence of event e (i.e., timestamp). A trace is a sequence of events $\sigma \in \mathcal{E}^*$ such that each event appears only once and time is non-decreasing, i.e., for $1 \leq \mathrm{i} < \mathrm{j} \leq |\sigma| : \sigma(\mathrm{i}) \neq \sigma(\mathrm{j})$ and $\overline{\sigma(i)} \leq \overline{\sigma(j)}$. $\mathcal{C}$ is the set of all possible traces (including partial traces). An event log is a set of traces $L \subseteq \mathcal{C}$ such that each event appears at most once in the entire log, i.e., for any $\sigma_1, \sigma_2 \in L$: $\forall_{e_1 \in \sigma_1} \forall_{e_2 \in \sigma_2}\ e_1 \neq e_2$ or $\sigma_1 = \sigma_2$.

Process mining aims at the extraction of information from a set of real executions (event logs). Before any process mining technique can be applied to an event log, information can directly be obtained from the log through the *pre-processing* step. This information can include the number of traces and events in the log, the resources and activities, and the frequency of their occurrences in the log. Based on this information *log filtering* can be done, for example, to remove the resources and activities with infrequent occurrence in the log.

After this step, different process mining techniques can then be applied to the log to discover different perspectives, i.e., process perspective (example is the output from the Fuzzy Miner shown in Figure 4), organizational perspective, and case perspective. Orthogonal to these three perspectives, the performance perspective can also be discovered (example is the output from the Dotted Chart analysis shown in Figure 3). In this paper, we focus on the performance perspective and analyse what the effect of different resource patterns is, on the performance of people (denoted by the service times).

3 Use of Regression Trees for Analyzing Resource Patterns

In this section, we provide a brief description of regression tree and how they are used to analyse patterns influencing resource behavior.

3.1 The REPTree Algorithm for Building Regression Trees

Machine learning algorithms have become a widely adopted means for the extraction of knowledge from large amounts of data [11, 16]. Regression trees are constructed based on a recursive partitioning procedure and the starting point (complete dataset) is referred to as the *root* and each split in this dataset is called a branch. The data subset resulting from the splitting is called a *node* and the ending nodes are referred to as the *leaf nodes* of the tree. At each leaf node, the regression tree stores a class value that represents the average value of instances that reach this node [5, 16].

In Weka software package, regression trees are implemented in the REP (Reduced Error Pruning) Tree algorithm [16]. REPTree are used to generate the rules useful for constructing the regression trees. The REPTree algorithm builds the tree using *information gain* as the splitting criterion, and the *reduced-error pruning* technique for pruning the tree. The algorithm splits the data into two sets, i.e., the building and the pruning sets. The optimal tree size is determined by using the stratified 10-fold cross-validation procedure that maximizes the prediction accuracy on pruning data [13]. Under the 10-fold cross validation, the data is randomly divided into 10 pairs and each part is held out in turn and the built model is trained on the remaining nine-tenths of the data, and the error rate is calculated on the hold out set. This procedure is repeated based on the number of pairs. The error estimates are then averaged to get the overall error estimates which are calculated for both the building and pruning sets[13]. In the next section, we discuss the resource patterns that were defined and analyzed in this paper.

3.2 Resource Patterns

As stated already, event logs originate from processes that are supported by PAISs. These logs can be analyzed to obtain information characterizing resource behavior. In this section, we introduce the concept of "resource patterns" as a way of characterizing the behavior of resources. Figure 1 gives an overview of the approach taken in this paper to describe the analysis of resource patterns and their effect on the service times. From the event logs represented in the MXML (Mining XML) format [6], information is obtained and translated into tabular format showing per event in the log, the different patterns associated with it. We identify patterns based on when resources start executing tasks and the following patterns are considered in this paper:

a) The time of the day when activities are started, i.e., morning period (if started before 12 hours), afternoon period (if started between 12 and 15 hours), and evening period (if started after 15 hours).
b) The day of the week when activities are started, i.e., weekday and weekend.
c) The particular hour when activities are started, for example, at 8:20hrs, 16:45hrs.
d) The resource and activity name combinations.

When building the regression tree, these patterns are considered as the explanatory variables in the regression model.

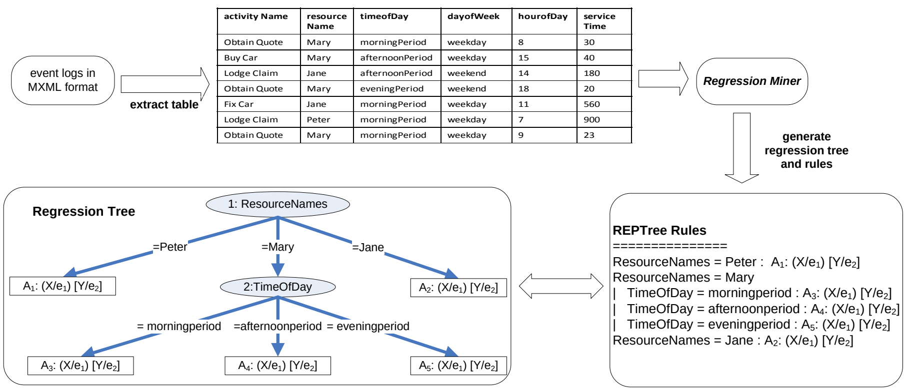


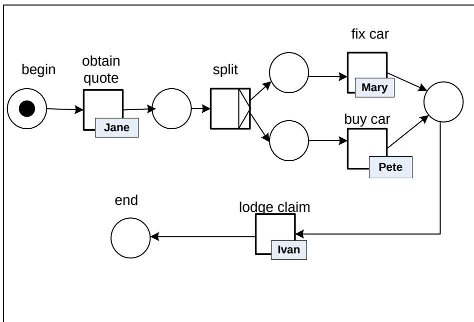
Fig. 1. Overview of the approach taken in this paper. From the logs represented in MXML format, we extract a table with the different resource patterns (a) activity names, (b) resource names, (c) *timeofDay*, (d) *dayofWeek*, (e) *hourofDay* and (f) service times (difference between the completion and start times measured in minutes).

Given the table in Figure 1, the *Regression Miner* generates a regression tree (on left-hand side) and rules that produce this tree (on right-hand side). These results depict how the resource patterns influence the service times. The patterns can be analyzed for events executed based on three different perspectives, i.e., resource, activity and resource/activity combinations. The regression tree shown in this figure is constructed based on the resource perspective with a root node as *resourceNames*, which is also given as the first line in the REPTree rules. This node is then subdivided into three other nodes based on the three different resources also seen in the REPTree rules, i.e., *Pete*, *Mary* and *Jane*. For the resources *Pete* and *Jane*, there are no outgoing branches from these nodes hence the leaf nodes (identified as the node given by $A : (X/e_1)[Y/e_2])$ whereas for resource *Mary*, we have an internal node, i.e., *timeofDay* which is subdivided into three branches also resulting into the leaf nodes. The REPTree rules obtained can be translated into the regression tree by taking the values such as *resourceNames* and *timeofDay* (on the left-hand side before the equals sign), as the nodes in the tree and their output (on the right-hand side of the equals sign) as the branches of the tree.

As stated already, when building the regression tree, the REPTree algorithm splits the data into two sets, i.e., building and training sets. These sets explain the values for the output shown in the leaf nodes, i.e., $A : (X/e_1)[Y/e_2]$ and are explained as: (a) A: average value of the instances that reach this node, (b) X: the number of instances used in the training set, (c) e_1: the value of the errors estimates in the training set, (d) Y: the number of instances in the pruning set, and (e) e_2: the value of the errors estimates in the pruning set. From these values explained, it is interesting to compare the average values for each leaf nodes in the tree and this is the main basis of discussion in this paper.

3.3 Car Repair Example

In this section, we introduce a running example developed in CPN tools [9] used for evaluating the idea discussed in this paper. The main focus here is on the results obtained from the event logs generated from the simulation model and not the CPN model. Figure 2(a) shows a simplified example of a process involving a car accident. The first step taken in this process is to obtain a quote for the costs involved in dealing with the car damage. After this step, a choice is made whether the car is going to be fixed or a new one is to be bought. After this decision has been made, then an insurance claim is lodged.



(a) WF-Model of the car accident example

```
REPTree Results
===============
ResourceNames = Ivan
|   TimeOfDay = morningperiod : 844.13 (77/1091127) [40/468625.59]
|   TimeOfDay = afternoonperiod
|   |   HourOfDay < 12.5 : 1135.39 (7/1065338.78) [3/543270.75]
|   |   HourOfDay >= 12.5 : 772 (13/545898.22) [9/579048.78]
|   TimeOfDay = eveningperiod: 50.98 (159/1146.58)[92/7769.4]
ResourceNames = Mary
|   TimeOfDay = morningperiod : 690.19 (52/543548.52) [26/425004.14]
|   TimeOfDay = afternoonperiod
|   |   DayOfWeek = Weekday : 1222.79 (5/790240) [8/1079750]
|   |   DayOfWeek = Weekend : 400 (3/13866.67) [0/0]
|   TimeOfDay = eveningperiod
|   |   HourOfDay < 20.5
|   |   |   DayOfWeek = Weekday : 52.68 (17/1951.7) [11/953.35]
|   |   |   DayOfWeek = Weekend : 67 (12/3701.75) [4/4964.75]
|   |   HourOfDay >= 20.5 : 39.23 (52/1674.63) [16/979.55]
ResourceNames = Pete : 37.81 (130/2249.65) [64/1136.88]
```

(b) REPTree results of event logs from the CPN model

Fig. 2. Workflow model of a car accident process and the REPTree analysis results. The REPTree rules can be translated into a regression tree (see Section 3.2).

In the CPN model, we incorporated the information about resource patterns based on the time when activities were started. The information was added as a factor on the execution times of the activities. For activities *obtainquote* (executed by *Jane*) and *buycar* (executed by *Pete*), the service time were entirely dependent on the case durations, whereas for activities *fixcar* (executed by *Mary*) and *lodgeclaim* (executed by *Ivan*) the service times were dependent on the both the activity durations and the resource patterns. If an activity was started in the morning and afternoon period, then their service times are dependent on these patterns (in the model, we added a factor of 10 to the service times) but if it was started in the evening period, then the service times are entirely dependent on the generated activity durations.

From the REPTree results shown in Figure 2(b)[2], the important resource patterns are: *timeofDay*, *dayofWeek* and the *hourofDay*. These patterns were significant for resources *Ivan* and *Mary* and the activities they executed in the morning and afternoon periods, took a much longer time on average as compared to the activities executed in the evening period. For example, for *Ivan* the average service time for events executed in the afternoon period took 1135 minutes while in the evening period, the average was 50 minutes. For *Pete*, it did not matter when he started activities hence none of the resource patterns applied. The average execution time for *Pete* is low (37 minutes) in comparison to, for example, *Mary* whose average execution time for activities in the morning period is 690 minutes.

From these results obtained based on the logs generated from the CPN model, we can indeed show that the resource patterns have an effect resource performance. Such information can be obtained from the event logs and used as a basis for resource allocation in PAISs.

4 Experiments

We tested the plug-in on a real case study involving a multinational company that develops and maintains photocopier machines.

4.1 Case Study

The case study was conducted on event logs obtained from the Dutch branch of a company that develops and sells photocopiers. The company also provides maintenance and repair activities. For this case study, we analyzed the service process of the Dutch branch that deals with the maintenance and repair activities. Once a photocopy machine breaks down, a complaint is made to the company and this is handled by the *helpdesk* resource. Then a field staff is sent out to look at the photocopy machine and either the machine is repaired or if more information is needed then the *helpdesk* resource handles this request.

Figure 3 shows the result obtained from the Dotted Chart Analysis in ProM [15] which shows a "helicopter view" of the spread of events in the log over time and it plots a dot for each event in the log hence we use it to gain some insight in the photocopier process. The x-axis shows the time when events occurred and

[2] In the discussion of the results for the *hourofDay* as a leaf node, the value used for branching, for example, 20.5 is interpreted as 20:30 hours.

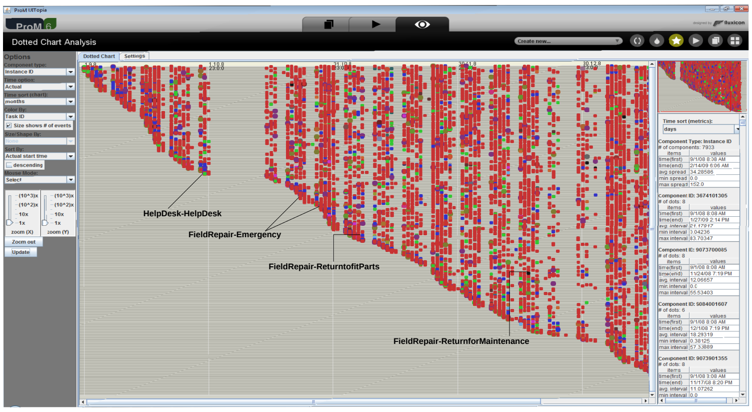


Fig. 3. The dotted chart for the event log containing 33912 events relating to 7933 photocopier machines.

here the actual time is used. The y-axis shows the component types which in this case are the instances representing the photocopier machines. The instances are sorted based on their start times.

The metrics on the left-hand side of Figure 3 show that the average case duration was about 34 days and the longest case took 152 days. The log contains information about 7933 cases, 33912 events, and 15 activities. The start date of the log is “2008-09-01” and the end date is “2009-02-14” and the log obtained contained 16616 events. From the results, we can see that the events for the complaints that arrive at the beginning take a much longer time in comparison to the events for the complaints that arrive later on. It is easy to see the actual times when work is actually done in the process, for example, from the “2008-10-01” - “2008-10-15”, there were no photocopier machines repaired.

Based on the activity perspective from the dotted chart, we observe that the activities in the event log have sub-types, for example, from Figure 3 we observe that activity *fieldrepair* has a number of sub-types, e.g., *emergency*, *courtesy* and *returntofitparts*. Further on, the resource names in the log were derived based on the main activities as seen in Figure 5(b), i.e., *fieldrepair* and *helpdesk*.

The Fuzzy Miner in ProM was used to discover the process model as shown in Figure 4. The *fuzzy model* focusses on the most significant behavior in the log and the flow from one activity to another. The main activities in the process are highlighted (with rectangular shape). For example, the main activities shown are: *emergency*, *timeDispatch*, *returntofitparts*, *maintenance* and *helpdesk*. The insignificant activities are clustered together (with oval shape).

Based on the analysis based on the dotted chart (gives a view on the log) and the fuzzy model (gives a view on the process), we are able to gain insight into the photocopier repair process and the significant activities in this process. We applied the regression miner to the event log, to obtain the analysis of different resource patterns and their effect on the service times. The analysis was based

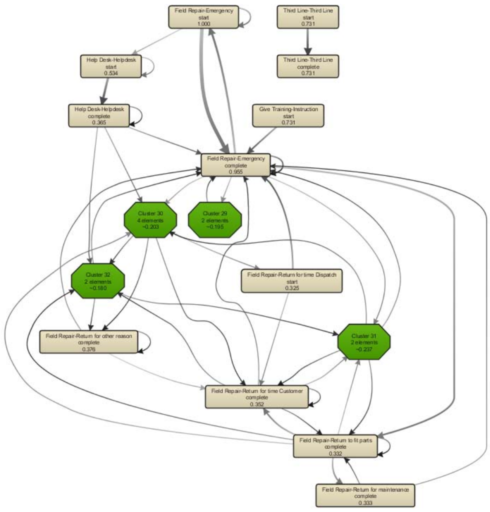


Fig. 4. The Fuzzy Model showing the main steps in the process (Fuzzy Miner [8]).

on events for the activity perspective (shown in Figure 5(a)) and the resource perspective (shown in 5(b)) [3]. The results shown on the leaf nodes reflect the average service times obtained based on the instances that were evaluated in each node and are measured in minutes.

REPTree Results

```
===============
ActivityNames = FieldRepair-Emergency
|   TimeOfDay = morningperiod : 592.3 (4956/481965.3) [2409/247493.8]
|   TimeOfDay = afternoonperiod : 548.3 (3631/187155.9) [1890/174013.7]
|   TimeOfDay = eveningperiod : 547.2 (732/225887.2) [360/177470.5]
ActivityNames = FieldRepair-Return for knowledge reason : 118.67 (71/7394.7) [28/6091.6]
ActivityNames = FieldRepair-Return for maintenance
|   TimeOfDay = morningperiod : 91.5 (5/893.04) [1/1339.5]
|   TimeOfDay = afternoonperiod : 92 (4/2790.75) [5/930.2]
|   TimeOfDay = eveningperiod : 165.9 (8/6918.73) [4/5174.5]
ActivityNames = FieldRepair-Return for other reason : 103.2 (41/8324.5) [24/6706.8]
ActivityNames = FieldRepair-Return for skill match reason : 119.9 (13/8498.85) [11/4589.7]
ActivityNames = FieldRepair-Return for time Customer
|   TimeOfDay = morningperiod
|   |   HourOfDay < 5.5 : 142.3 (3/5788.2) [0/0]
|   |   HourOfDay >= 5.5
|   |   |   HourOfDay < 9.5 : 89.5 (16/4186.1) [10/2418.6]
|   |   |   HourOfDay >= 9.5 : 51.9 (15/1719.9) [10/1703.3]
|   TimeOfDay = afternoonperiod : 100.8 (29/3442.2) [9/1561.8]
|   TimeOfDay = eveningperiod
|   |   DayOfWeek = Weekday : 90.9 (17/2858.3) [10/3467.3]
|   |   DayOfWeek = Weekend : 131.6 (6/5891.5) [8/7442]
ActivityNames = FieldRepair-Return for time Dispatch : 103.7 (11/3567.2) [6/1901.5]
ActivityNames = FieldRepair-Return to fit parts : 96.4 (771/5982.4) [374/5253.7]
ActivityNames = HelpDesk-Helpdesk : 45.5 (697/220025.6) [344/120244.7]
ActivityNames = ThirdLine-Thirdline : 0 (1/0) [0/0]
```

(a) Activity Dimension

REPTree Results

```
===============
ResourceNames = FieldRepair
|   HourOfDay < 11.5
|   |   HourOfDay < 7.5 : 124.26 (22/11324.4) [14/21850.4]
|   |   HourOfDay >= 7.5 : 560.95 (5272/467584.5) [2584/246711.3]
|   HourOfDay >= 11.5
|   |   HourOfDay < 17.5
|   |   |   DayOfWeek = Weekday
|   |   |   |   HourOfDay < 15.5 : 482.91 (2298/174955.7) [1226/172003.4]
|   |   |   |   HourOfDay >= 15.5
|   |   |   |   |   HourOfDay < 16.5 : 438.16 (440/191127.5) [202/175624.4]
|   |   |   |   |   HourOfDay >= 16.5 : 381.74 (57/169973.6) [33/105039.1]
|   |   |   DayOfWeek = Weekend
|   |   |   |   HourOfDay < 15.5
|   |   |   |   |   HourOfDay < 12.5 : 493.38 (410/170644.3) [182/175494.8]
|   |   |   |   |   HourOfDay >= 12.5
|   |   |   |   |   |   HourOfDay < 14.5
|   |   |   |   |   |   |   HourOfDay < 13.5 : 548.88 (474/212993.9) [232/211221.4]
|   |   |   |   |   |   |   HourOfDay >= 13.5 : 521.08 (471/225604.9) [240/186463]
|   |   |   |   |   |   HourOfDay >= 14.5 : 492.81 (478/184539.7) [235/149184.6]
|   |   |   |   HourOfDay >= 15.5
|   |   |   |   |   HourOfDay < 16.5 : 475.06 (376/147463.5) [202/188098.2]
|   |   |   |   |   HourOfDay >= 16.5 : 450.22 (58/817563.7) [24/121423.6]
|   |   HourOfDay >= 17.5 : 109.27 (23/11130.9) [20/3820.0]
ResourceNames = HelpDesk : 45.59 (697/220025.1) [344/120244.8]
```

(b) Resource Dimension

Fig. 5. Results from the Regression Miner based on the activity and resource perspective.

From Figure 5(a), we see that the events executed for activity *fieldrepair-emergency* in the morning period took up the longest time (592 minutes) in

[3] Figure 5 shows the REPTree rules as the output from the analysis, but the regression tree can be obtained from these rules as explained in Section 3.2.

comparison to the other activities. For the activities *fieldrepair-emergency, returnfortimecustomer,* and *returnformaintenance*, the *timeofDay* when activities were executed was a significant resource pattern. For example, for activity *returnformaintenance*, activities executed in the evening period took up a much longer time (165 minutes) compared to the average execution times of activities in the morning period (91 minutes). Further on, for activity *fieldrepair-emergency*, the events executed in the *morningperiod* took a longer time in comparison to the other times of the day, i.e., 592 compared to 548 and 547 minutes. For activity *returnfortimecustomer*, three patters were significant. For example, events executed in the *morningperiod*, i.e., before 9 hours took a longer time (142 minutes) in comparison to the events executed later on. Also, the events executed on the weekend (131 minutes) on took a longer time in comparison to the ones executed on the weekday (90 minutes) for this activity.

Figure 5(b) shows the results based on the resource perspective and the resource *fieldrepair* covers all the resources that executed events for the main activity *fieldrepair*. From this perspective, the important resource patterns that influence the service times are: *hourofday* and *dayofweek*. As mentioned earlier, the events for *fieldrepair* activity take up the longest execution time especially for events executed in the *morningperiod* (560 minutes from *hourofday*>=7:30 hours). Based on these results, we can see the specific hours of the day when the activities take up a much longer time, this was between 7:30 and 11:30 hours hence the morning period. The *dayofweek* is a significant pattern for activities executed between 15:30 and 17:30 hours. The activities executed on the weekend took a much longer time (548 minutes from *hourofday*<13:30 hours) in comparison to those executed on the weekday (482 minutes from *hourofday*<15:30 hours). From the results discussed we can be able to identify resource patterns that are significant and influence the performance of resources. For example, we can observe the particular hour when activities are executed and those periods where activities take a much longer time to execute activities.

4.2 Using Resource Behavior Characterization for Operational Support

In most of the PAISs, resource roles are usually defined statically and are based on privileges and capabilities of these resources. Further on, the use of resource history (the information about preceding work items in the previous cases recorded in event logs) is limited and many aspects about the way people actually work are ignored when making resource allocation decisions. With the resource characterization in this paper and our earlier work [3, 12], such information which is available in the event logs can be used as a basis when making resource allocation decisions.

We propose a new work allocation strategy also referred to as *history-dependent* allocation strategy, which uses resource characterization information as a basis for making the allocation decisions. Based on the properties of a concrete and running case in PAIS, information from the logs can be analyzed characterizing resource behavior to form a ranking strategy for resources. For example, based on an work item for a current case and group of resources capable of executing this work item, information about resource behavior, i.e., availability, time of day, workload can be learned from the logs and a recommendation made about which resource to allocate work to based on this ranking strategy. The characterization of resource information therefore, can provide a very useful means for selecting the most suitable resource to allocate work, and many allocation strategies can be enhanced based on this information.

5 Related Work

The work presented in this paper is related to work that has been carried out in the area of application of machine learning techniques in business processes. Machine learning algorithms have become a widely adopted means for the extraction of knowledge from large amounts of data [11, 16] and have been applied in a number of fields. The authors in [5] provide an overview of classification and regression trees in more detailed and formalized way.

In terms of the application of data mining techniques in business processes, we point out the work reported in [10] and [14]. In [14], the authors present the concept of decision mining which aims at the detection of data dependencies affecting the routing of cases. Starting from a process model, they analyze how data attributes influence the choices made in the process based on past process executions and they use decision trees for this purpose. Li et.al. [10] also use decision trees learning approach for the mining of staff assignment rules from process logs and an explicitly given organizational model.

The related work mentioned mainly focusses on the use of decision trees and their application in business processes. The main focus of the work presented in this paper is the use of regression trees for the analysis of different resource patterns and their effect on the service times. This approach is implemented as a plug-in in the ProM framework.

6 Conclusion

Using process mining techniques, we can generate insights into the way people actually work based on the information recorded in the event logs. In this paper, we have presented a *Regression Miner* plug-in implemented in the ProM framework that analyzes the effect of different resource behavior characteristics on performance. The analysis provided in this paper is based on the REPTree algorithm in Weka which is an implementation of regression trees. The mining and analysis of resource patterns is important and this information can be used as a basis for resource allocation decisions in PAISs.

Besides the resource patterns discussed in this paper, we take into account other aspects of resource behavior that influence resource behavior. Experiments show that these factors really influence performance [3] and can also be discovered from event logs [12]. In this paper, we have also discussed how such information can be used for operational decision making especially in resource allocation. In our future work, we aim at making this operational with implementation in ProM and based on the characteristics of a concrete case in PAIS, provide a recommendation about which resource to allocate work to based on the resource analysis characterization.

References

1. van der Aalst, W.M.P.: Process-Aware Information Systems: Lessons to Be Learned from Process Mining. In van der Aalst, W.M.P., Jensen, K.(eds) ToPNoC II, LNCS 5460, pp.1-26 (2009)
2. van der Aalst, W.M.P., van Dongen, B.F., Herbst, J., Maruster, L., Schimm, G., Weijters, A.J.M.M.: Workflow Mining: A Survey of Issues and Approaches. Data and Knowledge Engineering 47(2), 237-267 (2003)
3. van der Aalst, W.M.P., Nakatumba, J., Rozinat, A., Russell, N.: Business Process Simulation: How to get it Right? BPM Center Report BPM-08-07 (2008)
4. van der Aalst, W.M.P., Rosemann, M., Dumas, M.: Deadline-based Escalation in Process-Aware Information Systems. Decision Support Systems, 43(2), pp. 492-511 (2007)
5. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth, Pacific Grove, CA. (1984)
6. van Dongen, B.F., van der Aalst, W.M.P.: A Meta Model for Process Mining Data. In Casto, J., Teniente, E. (eds.) Proceedings of the CAiSE Workshops (EMOI-INTEROP Workshop) vol. 2, pp. 309-320 (2005)
7. Dumas, M., van der Aalst, W.M.P., ter Hofstede A.H.M.: Process-Aware Information Systems: Bridging People and Software through Process Technology. Wiley & Sons (2005)
8. Günther, C.W., van der Aalst, W.M.P.: Fuzzy Mining: Adaptive Process Simplification Based on Multi-perspective Metrics. In Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 328-343. Springer-Verlag, Berlin (2007)
9. Jensen, K., Kristensen, L.M., Wells, L. : Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. International Journal on Software Tools for Technology Transfer, vol. 9, pp. 213-254 (2007)
10. Ly, L.T., Rinderle, S., Dadam, P., Reichert, M.: Mining Staff Assignment Rules from Event-Based Data. In Bussler, C., and Haller, A. (eds) BPM 2005 Workshops, LNCS, vol. 3812 pp. 177-190. Springer-Verlag, Berlin (2006)
11. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
12. Nakatumba, J., van der Aalst, W.M.P.: Analyzing Resource Behaviour Using Process Mining. In S. Rinderle-Ma, S., Sadiq, S. Leymann, F.(eds.) BPM 2009 Workshops LNBIP, vol. 43, pp. 69-80 Springer, Berlin (2010)
13. Quinlan, J. R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA. (1993)
14. Rozinat, A., van der Aalst, W.M.P.: Decision Mining in ProM. In Dustdar, S., Faideiro, J.L., Sheth, A.(eds.) BPM 2006. LNCS, vol. 4102, pp. 420-425. Springer, Berlin (2006)
15. Song, M., van der Aalst, W.M.P.: Supporting Process Mining by Showing Events at a Glance. In Chari, K., Kumar, A. (eds) WITS 2007, pp. 139-145. (2007)
16. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques (Second Edition). Morgan Kaufmann (2005)