This is the author's version of a work that was submitted/accepted for publication in the following source:

An Extensible Framework for Analysing Resource Behaviour Using Event Logs

A. Pika[1], M. T. Wynn[1], C. J. Fidge[1], A. H. M. ter Hofstede[1,2], M. Leyer[3], and
W. M. P. van der Aalst[2,1]

[1] Queensland University of Technology, Brisbane, Australia
{a.pika,m.wynn,c.fidge,a.terhofstede}@qut.edu.au
[2] Eindhoven University of Technology, Eindhoven, The Netherlands
{w.m.p.v.d.aalst}@tue.nl
[3] Frankfurt School of Finance and Management, Frankfurt am Main, Germany
{m.leyer}@fs.de

Abstract. Business processes depend on human resources and managers must regularly evaluate the performance of their employees based on a number of measures, some of which are subjective in nature. As modern organisations use information systems to automate their business processes and record information about processes' executions in event logs, it now becomes possible to get objective information about resource behaviours by analysing data recorded in event logs. We present an extensible framework for extracting knowledge from event logs about the behaviour of a human resource and for analysing the dynamics of this behaviour over time. The framework is fully automated and implements a predefined set of behavioural indicators for human resources. It also provides a means for organisations to define their own behavioural indicators, using the conventional Structured Query Language, and a means to analyse the dynamics of these indicators. The framework's applicability is demonstrated using an event log from a German bank.

Keywords: Process mining, resource behaviour indicators, employee performance measurements

1 Introduction

Human resource management is an important function in any organisation. Human behaviour is considered to be "unequivocally the single most important element that can affect project success" [17]. Hence, having accurate information about how an employee performs their required tasks can be very valuable in evaluating that employee's performance. Knowledge about past actions of resources can also assist in forecasting how a resource might perform in the future.

Modern companies use information systems to automate their business operations. Details about such business operations (i.e., business processes) are usually recorded in logs, including information about the activity being performed, the time at which it was performed and the person responsible for carrying it out.
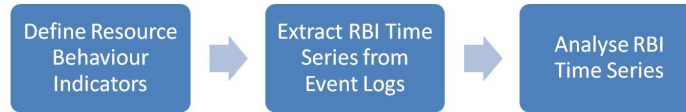
Fig. 1. Our three-step approach for resource behaviour analysis

Our research aims to make use of information recorded in event logs to extract knowledge about the behaviour of a resource over time.

Process mining focuses on analysing business processes using data available in event logs [20]. Recently, ways have been proposed for analysing certain aspects of resource behaviour via such logs, e.g., to extract social networks [16, 21] or organisational models [16]. However, these methods usually look at resource behaviour from an organisational, rather than individual, perspective and they do not consider the fact that a resource's behaviour can change over time.

Here we present an extensible software framework that allows organisations to extract knowledge about the behaviour of their employees from event logs, including how their behaviour has changed over time. Our approach consists of three steps, shown in Figure 1. (1) For various dimensions of resource behaviour, such as skills, utilisation, preferences, productivity and collaboration, we define a set of Resource Behaviour Indicators (RBIs) which are discoverable from event logs. The framework enables the definition of new RBIs as necessary via Structured Query Language (SQL) [5] statements. (2) We extract from an event log a time series that reflects the evolution of each RBI over time. (3) We analyse the time series using established methods such as time series charts with trends, outlier detection, change point detection, and time series comparison.

Knowledge extracted using the framework can provide valuable insights for resource performance evaluation, e.g., for rewards and recognition as well as for risk management. For example, a manager could see that an employee is getting involved in more complex tasks or is getting faster when executing certain tasks. A manager can also discover that an employee's workload is abnormally high at the end of each quarter, which could result in delays or low-quality outputs. The framework has been implemented as a plug-in within the process mining framework, ProM, and evaluated using data from a real-life event log.

2   Related and Previous Work

The performance of human resources with respect to business processes can be seen from two perspectives [14]. (1) Descriptive measures, i.e., how the performance of a resource is actually observed. (2) Normative measures, i.e., what the performance of a resource should be. Evaluation of an individual employee's performance is an important factor for companies because individuals are the smallest entity in the company who perform the work. However, performance indicators are typically defined on an aggregate level, e.g., company, department or process teams [11]. There is little literature on individual measures and only a few measures have been described [10, 18]. Our approach also uses indicators

to measure human resource performance, but we focus on individual resources and present an extensible framework which allows new measures to be defined.

Some techniques that can extract knowledge about certain aspects of resource behaviour from event logs have been proposed in the process mining area: Van der Aalst et al. proposed a method for extracting social networks [16, 21]; Song et al. proposed techniques for mining organisational models [16]; Nakatumba et al. investigated the effect of resource workload on service times [9]; and Huang et al. proposed a few measures for resource preference, availability, competence and cooperation and demonstrated how these measures can be used [4]. In our own earlier work, we showed how process risk indicators can be used to detect the risk of case delays [12, 13]. Some of these previous risk indicators were related to the behaviour of resources. We showed that the involvement of certain resources in a case or a high resource workload can result in case delays. Bose et al. proposed a generic framework for detecting concept drifts in business processes and defined features that characterise control-flow perspective [1]. By contrast, we focus here on understanding the evolving behaviour of individual resources over time and develop an extensible framework that can identify trends, anomalies and changes in resource behaviours.

3   Approach

To extract knowledge about resource behaviour we follow the three major steps depicted in Figure 1: defining RBIs (Section 3.1); extracting RBI time series from event logs (Section 3.2); and analysing RBI time series (Section 3.3). Our approach is based on the analysis of event log data, hence the quality of the data is crucial for getting meaningful results.

3.1   Defining Resource Behaviour Indicators

Our first challenge concerns the kinds of resource behaviour we wish to measure. This will vary depending of the reasons for the analysis. For example, if we wish to gain insights into the performance of an employee, we may look at their workload, their average duration to execute tasks, the number of interactions with other employees, etc. Thus, relevant employee behaviour depends on a particular context. In a specific situation there may be many indicators of interest [10]. Based on the literature we propose here the following general categories.

1. Skills [18]: *What can a resource do?*
2. Utilisation [10]: *What is a resource actually doing?*
3. Preferences [4, 19]: *What working behaviour does a resource often demonstrate?*
4. Productivity [8]: *How good is a resource at what it does?*
5. Collaboration [4, 21]: *How does a resource work with other employees?*

Within these five categories, we present a set of RBIs which can be drawn from typical event logs (depicted in Figure 2). Some predefined RBIs are generic
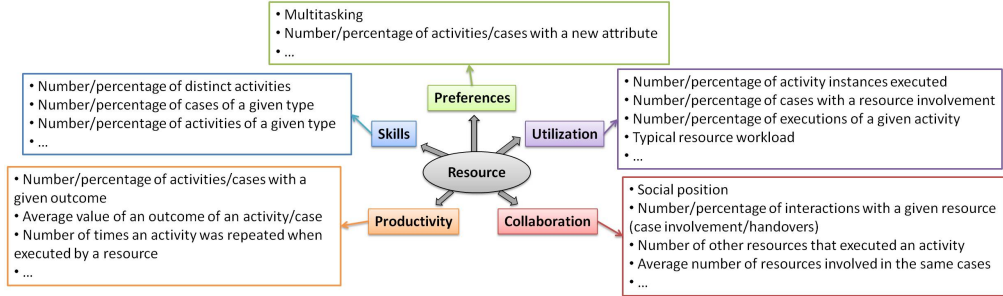
Fig. 2. Categories of resource behaviour and RBIs

while others are only relevant in specific contexts. Though these pre-defined RBIs can provide useful insights about resource behaviour, we recognise that managers may be interested in other indicators that reflect the specifics of their organisations. Therefore, the framework supports the definition of new RBIs.

Definitions An *event log EL* consists of a set of events $e \in \mathcal{E}$. Events are characterised by different attributes. Let $\{caseid, task, type, time, resource\}$ be *basic* event attributes, i.e., we assume that we can retrieve the values of these attributes for each event. We assume that *start* and *complete* event types are recorded for each activity. Such an event log can be treated as a relation whose relation scheme is specified by the set of event attributes. Similarly, a *case log CL* consists of a set of cases $c \in \mathcal{C}$ that are characterised by attributes. A case is uniquely identifiable by the case attribute *caseid*. A case log can also be treated as a relation whose relation scheme is specified by the set of case attributes. Events and cases can have other attributes (e.g., *outcome*, *cost*, etc.). Most of our pre-defined RBIs only require basic event and case attributes. We also show examples of RBIs that require richer data to be recorded. The value of attribute a of event e is denoted as e_a and the value of attribute a of case c is denoted as c_a. We derive from the basic event log case attribute *case_duration* (the time difference between the timestamps of the last and the first event in a case). The following event attributes are also derived from a basic event log: *task_duration* (the time difference between corresponding activity *complete* and *start* events), *workload* (the number of work items that are assigned to or started but not completed by a resource executing an event at the moment of event execution) and *workload_duration* (the time period during which the resource's workload has not changed). If a resource is involved in multiple processes that are recorded in separate logs, we assume that preprocessing is performed and logs are merged ensuring that case identifiers in the combined log are unique.

Let R be a set of resources, A be a set of activities, t_1 and t_2 be the beginning and the end of a given time slot and r be a given resource. We define the following functions that are later used in RBI definitions.

Events completed during a given timeslot [t1,t2]:
$$E_{CT}(t_1, t_2) \triangleq \{e \in EL \mid e_{time} \geq t_1 \wedge e_{time} < t_2 \wedge e_{type} = \text{`complete'}\}$$
Events executed by a given resource during a given timeslot:
$$E_{TR}(t_1, t_2, r) \triangleq \{e \in EL \mid e_{time} \geq t_1 \wedge e_{time} < t_2 \wedge e_{resource} = r\}$$
Events completed by a given resource during a given timeslot:
$$E_{CTR}(t_1, t_2, r) \triangleq E_{CT}(t_1, t_2) \cap E_{TR}(t_1, t_2, r)$$
Cases completed during a given timeslot:
$$C_{CT}(t_1, t_2) \triangleq \{c \in \mathcal{C} \mid \exists e \in E_{CT}(t_1, t_2)[e_{caseid} = c_{caseid}] \wedge$$
$$\nexists e1 \in EL[e1_{caseid} = c_{caseid} \wedge e1_{time} > t_2]\}$$
Cases in which a given resource was involved:
$$C_R(r) \triangleq \{c \in \mathcal{C} \mid \exists e \in EL[e_{caseid} = c_{caseid} \wedge e_{resource} = r]\}$$
Cases in which a given resource was involved during a given timeslot:
$$C_{TR}(t_1, t_2, r) \triangleq \{c \in \mathcal{C} \mid \exists e \in E_{TR}(t_1, t_2, r)[e_{caseid} = c_{caseid}]\}$$

Below we discuss RBIs in each of the categories of resource behaviour. Due to space limitations we provide formal definitions for only a small selection of pre-defined RBIs, the rest are briefly described in text.

1. Skills: What can a resource do? Resources within an organisation have different capabilities and they tend to acquire new skills in different paces. Knowledge about resource capabilities is needed for more precise resource scheduling [4, 18], resource performance evaluation and for resource development planning. For the RBIs in this category we assume that a resource is capable of performing those activities it has performed in the past. Hence, we can find out from an event log how a resource's skills are developing over time. These RBIs reflect only "demonstrated" skills, i.e., we cannot see from a log that a resource is capable of performing some activity if it has never performed it. RBI "Distinct activities" is relevant in the working environments where new employees are involved in few tasks, they learn new skills and are involved in more tasks over time (e.g., in fast food restaurants). Looking at the types of cases executed by a resource one can find out, for example, that the resource is getting involved in more complex cases over time.

Distinct Activities: The number of distinct activities completed by a given resource, r, during a given timeslot, t_1 to t_2.
$$Distinct\_Activities(t_1, t_2, r) \triangleq |\{task \in A \mid$$
$$\exists e \in E_{CTR}(t_1, t_2, r)[e_{task} = task]\}|$$

Case types: The number of cases with a given property (e.g., *complexity*) completed during a given timeslot in which a given resource was involved (requires case type attributes to be recorded in a log).
$$Complex\_Cases(t_1, t_2, r) \triangleq |\{c \in C_{CT}(t_1, t_2) \cap C_R(r) \mid$$
$$c_{complexity} = \text{`complex'}\}|$$

Other RBIs defined in this category include: the percentage of distinct activities completed, the number of activity completions with a given property, and the percentage of activity or case completions with a given property.

2. Utilisation: *What is a resource actually doing?* Utilisation RBIs reflect how active a resource is without considering the quality of its outputs. They are inspired by manufacturing measures, e.g., the number of units produced [10].

Activity completions: The number of activity instances completed by a given resource during a given time slot.
$$Activity\_Completions(t_1, t_2, r) \triangleq |E_{CTR}(t_1, t_2, r)|$$

Case completions: number of cases completed during a given time slot in which a given resource was involved.
$$Case\_Completions(t_1, t_2, r) \triangleq |C_{CT}(t_1, t_2) \cap C_R(r)|$$

We have also defined RBIs for: the percentage of activity instances completed, number and percentage of completions of a given activity, percentage of completed cases in which a resource was involved, and typical resource workload.

3. Preferences *What working behaviour does a resource often demonstrate?* Resources have different working preferences and styles that may affect their performance and overall process outcomes [19]. For example, they may prefer execution of certain activities [4], working on multiple tasks, executing similar tasks or taking risks.

Multitasking: The percentage of active time when a given resource is involved in more than one activity.
$$Multitasking(t_1, t_2, r) \triangleq \sum\nolimits_{\{e \in E_{TR}(t_1, t_2, r) | e_{workload} > 1\}} e_{workload\_duration} \Big/ \sum\nolimits_{\{e \in E_{TR}(t_1, t_2, r) | e_{workload} > 0\}} e_{workload\_duration}$$

New Attributes: The number of times a resource completed a task during a given timeslot with a new value of a given attribute, e.g., *sum* (reflects propensity to execute new hence risky tasks).
$$New\_Attributes(t_1, t_2, r, sum) \triangleq |\{e \in E_{CTR}(t_1, t_2, r) \mid \nexists e1 \in EL$$
$$[e1_{sum} = e_{sum} \wedge e1_{time} < e_{time}]\}|$$

4. Productivity *How good is a resource at what it does?* RBIs in this category reflect a resource's results, e.g., in terms of timeliness, costs or quality of outputs (assuming cost and quality information is recorded in the event log). We defined here RBIs for: the number/the percentage of activities/cases completed with a given outcome in which a resource was involved, average value of a given outcome for a given activity or a case completed by a resource, and the number of times when a given activity was repeated when completed by a resource.

Activity Outcomes: The percentage of activities completed during a given time slot by a given resource with a given outcome (e.g., *duration*).
$$In\_Time\_Activities(t_1, t_2, r, dur) \triangleq$$
$$|\{e \in E_{CTR}(t_1, t_2, r) \mid e_{task\_duration} < dur\}| \ / \ |\{e \in E_{CTR}(t_1, t_2, r)\}|$$

Case outcomes: The percentage of cases completed during a given time slot with a given outcome (e.g., *cost*) in which a given resource was involved (requires case attribute *cost* to be recorded).

$Overbudget\_Cases(t_1, t_2, r, cost) \triangleq$
$|\{c \in C_{CT}(t_1, t_2) \cap C_R(r) \mid c_{cost} > cost\}| \ / \ |C_{CT}(t_1, t_2) \cap C_R(r)|$

5. Collaboration *How does a resource work with other employees?* Collaborative aspects of resource behaviour are important as people are more often involved in teamwork nowadays. RBIs in this category can help us to learn about resource's collaboration patterns with a particular resource or to get insights about his overall social position within an organisation.

Social position: The percentage of resources involved in the same cases with a given resource during a given time slot.

$Social\_Position(t_1, t_2, r) \triangleq$
$|\{r_1 \in R \mid \exists c \in C_{TR}(r_1) \cap C_{TR}(r)\}| \ / \ |\{r_1 \in R \mid \exists c \in C_{TR}(r_1)\}|$

Interactions with a given resource: The number of cases completed during a given time slot in which two given resources were involved.

$Interactions\_With\_A\_Resource(t_1, t_2, r_1, r_2) \triangleq$
$|C_{CT}(t_1, t_2) \cap C_R(r_1) \cap C_R(r_2)|$

We have also defined collaboration RBIs for: the number of other resources that executed a given activity, the average number of resources involved in the same cases with a given resource, and the number of handovers with a given resource.

3.2 Extracting RBI Time Series from Event Log

The value of an RBI at a particular point in time is not very useful unless it is being compared with some other values. For such comparisons, we extract RBI time series as the second step. This consists of RBI values extracted for a given period of time, e.g., per day, week, month, etc, for a particular resource. The user selects a starting time point, time series sampling rate and a number of time slots. Selection of the time series sampling rate is an important step that can affect the analysis results. It is a well-known problem often discussed in the literature [6]. When choosing sampling rate for RBI time series one should consider process characteristics (e.g., process granularity) and the type of analysis one is interested in. For example, if a manager is interested in checking whether or not an employee is less productive on Mondays, daily RBI values are needed, rather than weekly or monthly.

Let $RBI_n(t_1, t_2)$ denote the value of an RBI n during a time slot (t_1, t_2); TS_{start} be the starting time point; $TS_{slotsize}$ be the sampling rate; TS_{size} be the number of time slots; $Start(t)$ and $End(t)$ are functions that return the beginning and the end of a timeslot for a given time t correspondingly. Then an RBI time series can be defined as:

$$TS_{RBI_n} \triangleq \{(RBI_n(Start(t), End(t)), t) \mid$$
$$t \in \{TS_{start} + i * TS_{slotsize} \mid i \in \{0, 1, \ldots, TS_{size} - 1\}\}\}$$

We use here the following pre-defined functions for the beginning and the end of a time slot: $Start(t) = t$ and $End(t) = t + TS_{slotsize}$. We also provide an interface for users to define their own functions. This gives flexibility to use different time series sampling methods, e.g., defining overlapping time slots.

3.3 Analysing RBI Time Series

During the third step, we analyse the extracted RBI time series and visualise the results. Our framework generates time series charts accompanied by trend lines. It can also automatically detect change points and outliers and provides a means for time series comparisons. Hence, one can compare RBI values for different resources or compare RBI values against benchmark values.

Time series charts alone accompanied by trend lines can give many interesting insights about the dynamics of resource behaviour. For example, a manager can see when the number of cases handled by a resource has increased or decreased significantly or a resource is handling an abnormally higher or lower number of tasks during certain periods. While charts are simple and powerful tools for time series analysis, they are not very convenient when the amount of available data is large. If a manager is doing periodic performance reviews for multiple employees and is interested in multiple RBIs it may be necessary to check hundreds of charts. To facilitate this, we enable the use of automatic techniques for time series analysis such as the detection of change points [15], outliers detection [22] and time series comparison [7].

One way to make time series analysis efficient is automatic detection of significant changes. One would like to know if such changes have occurred and the points in time when they did so. The problem of change point detection has received significant attention and many methods have been developed. Most of the existing change point detection techniques make certain assumptions about the distribution of the data, however recently a few non-parametric methods have been proposed [3, 15]. As we do not know the distributions of RBIs in advance, we advocate using such methods. A user can choose non-parametric tests to detect changes in location (Mann-Whitney), in scale (Mood) or to detect arbitrary distributional changes (Kolmogorov-Smirnov, Cramer-von-Mises) [15]. Alternative approach would require learning the distribution of the data and using an appropriate change point detection test.

Detecting outliers in RBI time series (i.e., points in time when RBI values significantly deviate from typical values) can be helpful in problem investigations. For example, a high resource workload during specific points in time may explain case delays. Many outlier detection techniques have been proposed in different research fields. We use an outlier detection method that fits the distribution of the observations and detects observations which are unlikely to be generated by this distribution [22].

The time series analysis methods described above allow the analysis of how the behaviour of a resource has evolved over time. Another way to evaluate performance of a resource using RBIs is to compare its behaviour with the behaviour of other resources. This allows us to quickly identify those resources
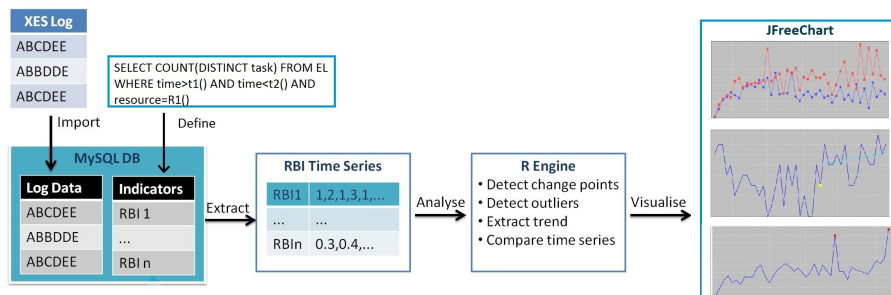
Fig. 3. Overview of the plug-in for analysing resource behaviour

whose behaviour significantly differs from others (e.g., over-performing or under-performing employees). To compare RBI time series we recommend using the non-parametric Mann-Whitney U-test [7].

However, typical event logs do not contain complete information about the behaviour of a resource, hence the meaning of an RBI can be subject to interpretation. In addition, the existence of certain change points may not provide an explanation regarding the root causes behind such changes. Let's consider as an example RBI "typical duration of a given activity completed by a given resource". If we detect that a resource is getting faster when executing a particular activity, this may mean that the resource is getting more efficient as it becomes more experienced in carrying out such an activity or it could mean that the resource is doing a hasty and possibly poor job.

4   Validation

This section first describes the implementation of our approach as a plug-in of the process mining framework ProM[4] and then presents the practical insights gained from testing the framework with an event log from a German bank.

4.1   Implementation

Figure 3 depicts the main functionality of our plug-in. An input to the plug-in is an XES event log. XES is a standard format for event log data[5]. Event log data is then converted and stored in a MySQL[6] database. Tables EL and CL are created that contain event and case attributes correspondingly. Pre-defined RBIs are stored in the database as views. The plug-in provides an interface that allows users to define their own RBIs using SQL. We adopted SQL for this purpose because it is standardised, unambiguous, and widely-used, although

---

[4] http://www.promtools.org/prom6/
[5] http://www.xes-standard.org/
[6] http://www.mysql.com/

other formalised languages could also have been used, most notably 'set builder' (a.k.a. 'set comprehension') mathematical notation. To define a new RBI a user needs to provide an SQL statement characterising the value of the RBI per a given time slot. Table 1 shows an example of an RBI definition. When defining RBIs users can use pre-defined parameters implemented as functions (*t1(),t2()* and *R1()* in Table 1). They can also define their own parameters by providing parameter names and data types, and functions that return the values of the parameters will be created automatically and can be used in definitions of RBIs.

Table 1. Example of definition of an RBI using SQL

RBI "Distinct activities"	$Distinct\_Activities(t_1, t_2, r) \triangleq$
	$\|\{task \in A \mid \exists e \in E_{CTR}(t_1, t_2, r)[e_{task} = task]\}\|$
SQL definition	SELECT COUNT (DISTINCT $task$) FROM EL WHERE $time \geq t1()$
	AND $time < t2()$ AND $resource = R1()$ AND $type =$ 'complete'

To start the analysis a user needs to select a resource whose behaviour is to be analysed, the RBIs of interest, a time period and time series sampling rate. Optionally the user can define functions that return the beginning and the end of a time slot for a given point in time. Based on these inputs the plug-in will extract time series for the RBIs. The user then selects methods of time series analysis and provides corresponding input parameters.

For statistical analysis of RBI time series we use R[7] whose functionality is accessed from the plug-in using JRI Java/R Interface[8]. To detect change points we use the CPM framework that implements many popular parametric and non-parametric tests and is implemented as R package *cpm*[9]. The framework allows control over the level of changes that are detected by choosing the value of parameter ARL0, the "average number of observations between false-positive detections assuming that no change has occured" [15]. Higher values of ARL0 allow detection of only significant changes. For outlier detection we use the R package *extremevalues*[10] that implements Van der Loo's outlier detection methods [22]. For time series comparison we use R's *wilcox* method[11] that implements the Mann-Whitney U-test [7] and for trend fitting we use R's *lm* method[12]. The results of the analysis are visualised using the JFreeChart library[13].

---

[7] http://www.r-project.org/

[8] http://rforge.net/JRI/

[9] http://cran.r-project.org/web/packages/cpm/vignettes/cpm.pdf

[10] http://cran.r-project.org/web/packages/extremevalues/extremevalues.pdf

[11] http://stat.ethz.ch/R-manual/R-patched/library/stats/html/wilcox.test.html

[12] http://stat.ethz.ch/R-manual/R-devel/library/stats/html/lm.html
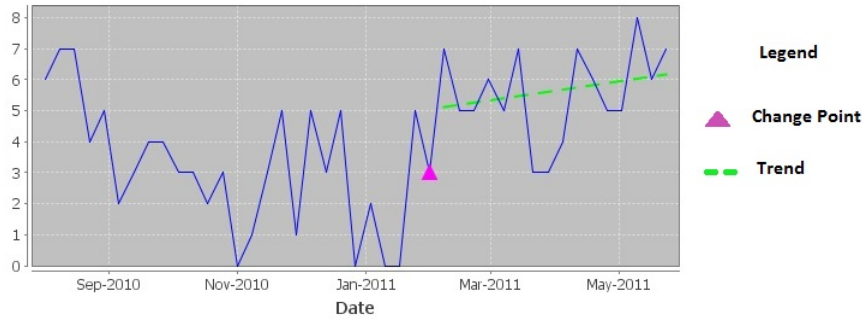
[13] http://www.jfree.org/jfreechart/

Fig. 4. The number of distinct activities completed by resource R9

4.2 Analysing the behaviours of resources in a German bank

In this section we demonstrate how the framework can be used to analyse resource behaviour by applying the techniques to an actual event log. An event log which describes a loan application process in a German bank was used in this experiment. The log has the following characteristics:
– Number of cases: 1731; log duration: 43 weeks;
– Number of activities: 25; number of resources: 220;
– Average case duration: 18.7 days; average number of resources in a case: 4.

The resources can also be involved in other processes, however, the organisation allows resources to spend up to 85% of their time on this loan application process. Hence, all results discussed in this section reflect resource behaviour in just one process but it can be considered as a representative set of data for the analysis. We selected the ten most frequent resources in the log (referred to here as R1–R10) for our analysis. In our experiments we used two different values of the time slot size: one week and one day. We looked at the daily values of an RBI "number of activity instances completed" for the ten resources. We found that some resources tend to complete slightly more activities in the middle of the week than on Mondays and Fridays while others do not display any differences within a week. In all experiments described below, the time slot size was one week. We show below RBIs from each resource behaviour category as examples.

Figure 4 depicts time series for the RBI "Number of distinct activities completed by a resource" extracted for resource R9. We can observe that the values of the RBI were decreasing until January 2011, but starting from February 2011 resource R9 began executing more distinct tasks. The plug-in was able to identify this change point using the Mann-Whitney test for identifying changes in location, as depicted by the triangle shape in Figure 4. Changes in resource behaviour can affect organisational performance. Automatic identification of such changes can help managers to take timely actions.

To demonstrate the resource utilisation indicators we used as examples the RBIs "Number of cases completed in which a given resource was involved" and "Percentage of cases completed in which a given resource was involved". Figure 5(a) shows that during two weeks (in February and May 2011) an abnormal number of cases were completed in which resource R1 was involved (marked as diamond shapes). In Figure 5(b) is the percentage of cases completed in which R1 was involved and we can see only one outlier (in February 2011). All outliers
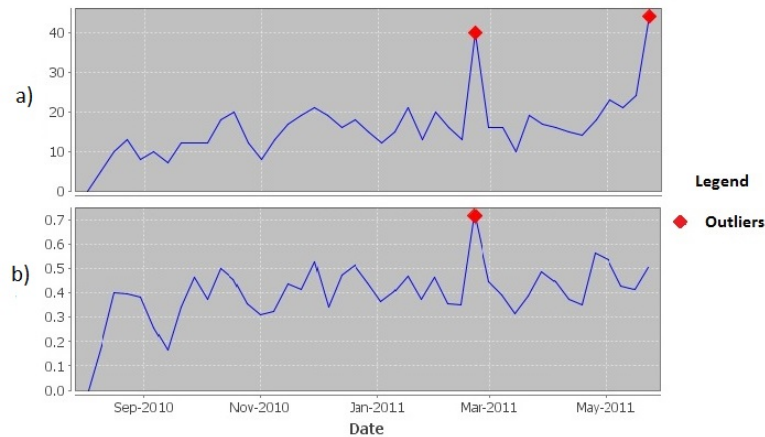
Fig. 5. Number (a) and percentage (b) of cases completed in which R1 was involved
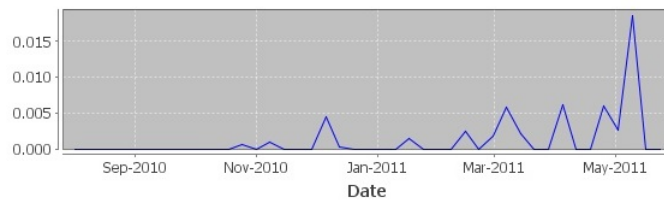

Fig. 6. Percentage of active time when resource R7 is involved in more than one activity

were identified by the plug-in. We can conclude that the reason for the higher number of cases processed by resource R1 in May 2011 was a higher case arrival rate, as the percentage is not abnormal, while the higher number of case completions in February 2011 cannot be explained in a similar manner. Abnormal resource workload can be a cause of process delays or low-quality outputs. Further investigations may help to discover reasons for such behaviour and conduct any corrective or preventive actions if necessary.

We also analysed the behaviour of the RBI "Multitasking" for these ten resources and found that all of them were involved in multiple activities on only rare occasions. Figure 6 depicts multitasking preferences for resource R7. Although we can observe a slight increase in tendency to multitask over time the percentage of time when the resource is working on more than one activity is very small (no more than 2% of the resource's active time).

Figure 7 demonstrates the performance of the productivity RBI "Average duration of a case in which a given resource was involved" for resource R1. We can observe that starting from November 2010 the average duration of cases in which resource R1 was involved is higher (typically around 25 days) when compared with the average duration of cases in which resource R1 was not involved (typically around 15 days). A conclusion one may draw from this is that R1 is a source of case delays. Alternatively it is possible that R1 is an expert who is usually involved in more complex and lengthy cases.
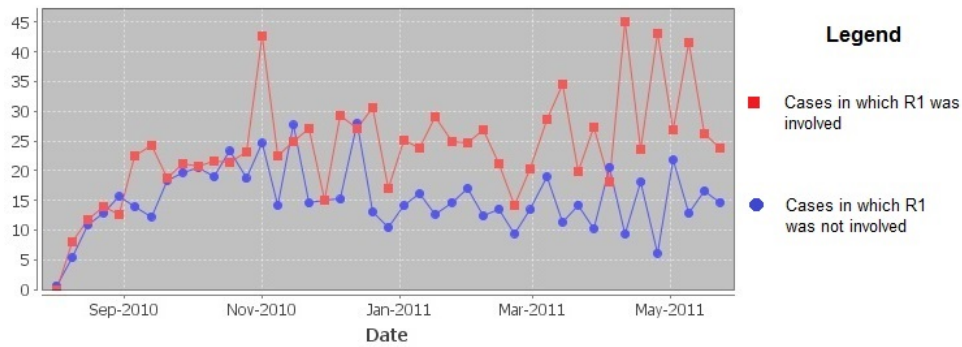
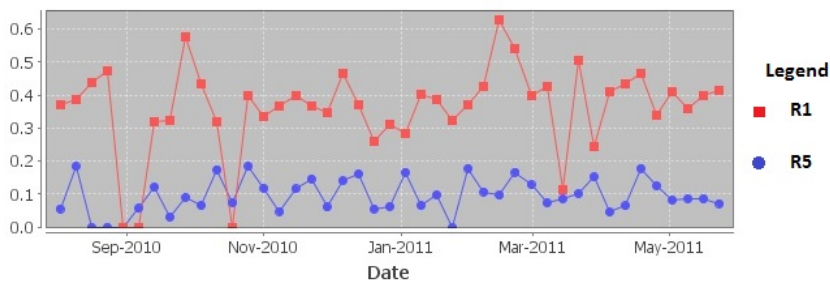**Fig. 7.** Average case duration (in days) for resource R1

Fig. 8. Percentage of resources involved in the same cases with a given resource

Figure 8 shows the percentage of resources that are involved in the same cases with a particular resource. We can see, for example, that during a week resource R1 interacts in different cases with approximately 40% of all resources involved in the process during that week, while resource R5 typically interacts with 5–15% of resources involved in the process. This RBI reflects the social position of a resource within the organisation. A high number of interactions with other employees often means that a resource has a higher influence in an organisation [4]. We can also see in Figure 8 that the social positions of the two resources did not change significantly during the period of analysis.

Having analysed the behaviours of the ten most frequent resources in the log we can see that most of the resources are active during the period of analysis and their behaviours did not change significantly. We saw that some resources are getting faster in execution of some tasks (e.g. R3) and some are often involved in long-running cases (e.g. R1). We learned that some resources do not process cases with high loan amounts requested. These are only few examples illustrating knowledge discovered from the log. Managers can use this knowledge to evaluate the performance of these resources and to plan their future workload and development. Identified outliers (e.g., in Figure 5) show behavioural anomalies that could have affected the process performance. Further investigation is required in order to understand their causes and consequences. It can provide insights for improving the process performance in the future. We showed that behaviour of

some RBIs can have different interpretations as the event log does not contain all information about resource behaviour. However, such RBIs can still provide valuable insights to managers as the managers may be able to interpret the RBIs using their own knowledge of the process. Automatic analysis of time series can help a manager quickly gain insights about the dynamics of resource behaviour, e.g., to get all RBIs where changes in behaviour have occurred. One can also find those resources whose behaviour significantly differs from others.

5 Conclusions

Human resources are responsible for carrying out business processes, however their actions can have both positive and negative impacts on organisational performance. Having objective knowledge about the behaviour of these resources will assist managers to evaluate their performance, to investigate issues and to apply appropriate rewards or mitigation actions.

In this paper we presented an automated, extensible framework for the analysis of the dynamics of resource behaviour using event logs. The framework can be used to measure and track resource's skills, utilisation, preferences, productivity and collaboration. As a starting point for such analysis we proposed a set of indicators for each category of resource behaviour. Our framework also provides an interface to define new RBIs. For each RBI, we extract a time series, analyse its dynamics and visualise the results.

The current framework concentrates on measurements for individual RBIs and any interdependencies among indicators are not considered. Hence, an extension of this work will be to combine several RBIs to a single performance measure. For this purpose we can use Data Envelopment Analysis – a non-parametric efficiency measurement method developed by Charnes et al. [2]. Another possible extension of this work is using the results of the analysis to enrich event logs with knowledge about resource behaviour.

We have applied the framework to a real event log and demonstrated that it can extract knowledge about resource behaviour. We have shown how our framework can be used to analyse various aspects of behaviour and demonstrated the different types of analysis that can be applied. We also explained how this analysis can help in evaluating a resource's performance and in identifying resource-related issues.

References

1. J.C.R.P. Bose, W.M.P. van der Aalst, I. Zliobaite, and M. Pechenizkiy. Dealing with concept drifts in process mining. *IEEE Transactions on Neural Networks and Learning Systems*, PP(99), 2013.
2. A. Charnes, W.W. Cooper, and E. Rhodes. Measuring the efficiency of decision making units. *European journal of operational research*, 2(6):429–444, 1978.
3. D.M. Hawkins and Q. Deng. A nonparametric change-point control chart. *Journal of Quality Technology*, 42(2):165–173, 2010.

4. Z. Huang, X. Lu, and H. Duan. Resource behavior measure and application in business process management. *Expert Systems with Applications*, 39(7):6458–6468, 2012.

5. International Organization for Standardization /International Electrotechnical Commission. *Information technology - Database languages - SQL (ISO/IEC 9075:2011)*, 2011.

6. J. Lijffijt, P. Papapetrou, and K. Puolamäki. Size matters: Finding the most informative set of window lengths. In *Machine Learning and Knowledge Discovery in Databases*, pages 451–466. Springer, 2012.

7. H.B. Mann and D.R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, 18(1):50–60, 1947.

8. P. Murphy. Service performance measurement using simple techniques actually works. *Journal of Marketing Practice: Applied Marketing Science*, 5(2):56–73, 1999.

9. J. Nakatumba and W.M.P. van der Aalst. Analyzing resource behavior using process mining. In *Proceedings of BPI'2009*, volume 43 of *LNBIP*, pages 69–80. Springer, 2010.

10. A. Neely, M. Gregory, and K. Platts. Performance measurement system design: a literature review and research agenda. *International Journal of Operations & Production Management*, 25(12):1228–1263, 2005.

11. S.S. Nudurupati, U.S. Bititci, V. Kumar, and F.T.S. Chan. State of the art literature review on performance measurement. *Computers & Industrial Engineering*, 60(2):279–290, 2011.

12. A. Pika, W.M.P. van der Aalst, C.J. Fidge, A.H.M. ter Hofstede, and M.T. Wynn. Predicting deadline transgressions using event logs. In *Proceedings of BPI'12*, volume 132 of *LNBIP*, pages 211–216. Springer-Heidelberg, 2013.

13. A. Pika, W.M.P van der Aalst, C.J. Fidge, A.H.M. ter Hofstede, and M.T. Wynn. Profiling event logs to configure risk indicators for process delays. In *Advanced Information Systems Engineering*, volume 7908 of *LNCS*, pages 465–481. Springer, 2013.

14. D.S.P. Rao, C.J. O'Donnell, G.E. Battese, and T.J. Coelli. *An introduction to efficiency and productivity analysis*. Springer, 2005.

15. G.J. Ross and N.M. Adams. Two nonparametric control charts for detecting arbitrary distribution changes. *Journal of Quality Technology*, 44(2):102–116, 2012.

16. M. Song and W.M.P. van der Aalst. Towards comprehensive support for organizational mining. *Decision Support Systems*, 46(1):300–317, 2008.

17. V. Thevendran and M.J. Mawdesley. Perception of human risk factors in construction projects: an exploratory study. *International Journal of Project Management*, 22(2):131–137, 2004.

18. G.M. Thompson and J.C. Goodale. Variable employee productivity in workforce scheduling. *European Journal of Operational Research*, 170(2):376–390, 2006.

19. W.M.P. van der Aalst. Business process simulation revisited. In J. Barjis, editor, *Enterprise and Organizational Modeling and Simulation*, volume 63 of *LNBIP*, pages 1–14. Springer, 2010.

20. W.M.P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag, Berlin, 2011.

21. W.M.P. van der Aalst, H.A. Reijers, and M. Song. Discovering social networks from event logs. *Computer Supported Cooperative Work (CSCW)*, 14(6):549–593, 2005.

22. M.P.J. van der Loo. Distribution based outlier detection for univariate data: discussion paper 10003. *Statistics Netherlands*, 2010.