Analyzing Resource Behavior Using Process Mining

Joyce Nakatumba and Wil M.P. van der Aalst

Eindhoven University of Technology
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands
{jnakatum, w.m.p.v.d.aalst}@tue.nl

Abstract. It is vital to use accurate models for the analysis, design, and/or control of business processes. Unfortunately, there are often important *discrepancies between reality and models*. In earlier work, we have shown that simulation models are often based on incorrect assumptions and one example is the speed at which people work. The "Yerkes-Dodson Law of Arousal" suggests that a worker that is under time pressure may become more efficient and thus finish tasks faster. However, if the pressure is too high, then the worker's performance may degrade. Traditionally, it was difficult to investigate such phenomena and few analysis tools (e.g., simulation packages) support workload-dependent behavior. Fortunately, more and more activities are being recorded and modern *process mining* techniques provide detailed insights in the way that people really work. This paper uses a new process mining plug-in that has been added to ProM to explore the *effect of workload on service times*. Based on historic data and by using regression analysis, the relationship between workload and services time is investigated. This information can be used for various types of analysis and decision making, including more realistic forms of simulation.

Key words: Process Mining, Yerkes-Dodson Law of Arousal, Business process Simulation.

1 Introduction

Organizations are increasingly using Process-Aware Information Systems (PAISs) to reduce costs and improve the performance and efficiency of important business processes. PAISs provide a means to support, control, and monitor operational business processes. Examples of PAISs are Workflow Management Systems (WFMSs), Business Process Management Systems (BPMSs) but also other "process-aware" systems, such as Enterprise Resource Planning Systems (e.g., SAP R/3, Oracle, JD Edwards, etc.), call-center systems, Product-Data Management Systems, and process-centric middleware (e.g., IBM's WebSphere, JBoss, etc.) [5]. While PAISs support processes they also record information about these processes in the form of so-called *event logs*, also known as audit trails or transaction logs [2]. In these logs, information is stored about activities as they are being executed. This information can include the times at which events were

executed, who executed these events, etc. This information can be used among other things, for performance analysis,e.g., the identification of bottlenecks in a process model. Event logs provide an excellent source of information for *process mining*, i.e., extracting non-trivial knowledge from historic data. In this paper, we advocate the use of process mining in order to extract characteristic properties of resources.

Many organizations have used *simulation* at some point to analyze, for example, the performance of their business processes. In most of these simulation approaches, however, the models used are very naive and do not use the information recorded in the event logs. We refer to this kind of simulation as *traditional simulation* [1]. Traditional simulation, therefore, *rarely uses historic information* and also typically suffers from the problem that *human resources are modeled in a rather naive way*. As a result, the simulation results obtained are seldom a good reflection of what is actually happening in the organization.

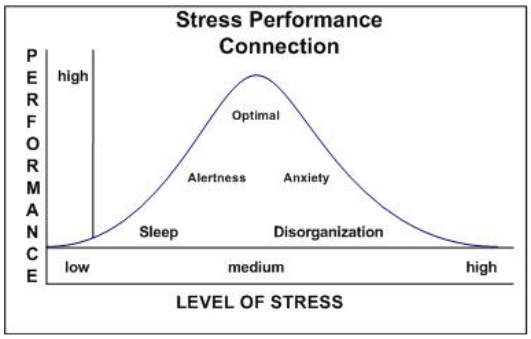


Fig. 1. Yerkes-Dodson Law modeled as U-shaped Curve. When the stress level is low, the performance is also low. This increases as the stress levels also increase up to a certain optimal level beyond which the performance drops (adapted from [12]).

In [1], we identify aspects of resource behavior that are not captured in today's simulation tools. In particular, we point out that people do not work at constant speeds and their processing speeds are usually influenced by their workload. We refer to this as *workload-dependent processing speeds* and the analysis of this phenomenon is the focus of this paper. There are a number of studies that suggest a relation between workload and performance of workers. In the literature, this phenomenon is known as the "Yerkes-Dodson Law of Arousal" [12]. This law models the relationship between arousal and performance as an inverse U-Shaped curve as depicted in Figure 1. This implies that for a given individual and a given set of tasks, there is an optimal level at which the performance of that individual has a maximal value and beyond this optimal level the worker's performance collapses. This aspect has been studied in social sciences and operations management. Until recently, there were no means for systematically observing such phenomena in the workplace. However, because human activity is increasingly recorded by PAIS technology and the availability of process min-

ing tools such as ProM, it is now possible to better characterize resource behavior based on empirical data. Therefore, this is important for two main reasons.

First of all, a better resource characterization will help *to make simulation models that are more realistic and that are tightly coupled to PAISs* [1, 9, 10]. This will enable the use of process mining for operational decision making, i.e., based on a reliable model and both real-time and historic data, it becomes worthwhile to use simulation on a daily basis. This paper therefore, is a first step approach to the use of process mining as a technique for the extraction of characteristic properties of resources from event logs, i.e., the effect of changing workload on resource behavior. This information can then be incorporated in simulation models. The results of various process mining techniques can be combined as shown in [9] to yield an integrated simulation model.

Secondly, good insights into the behavior and performance of people will assist in a *better work distribution*. One of the major objectives of a PAIS is to facilitate the distribution of work amongst the group of resources involved in a process. However, today's PAIS systems use very limited understanding of resource behavior. But with better characterization of resource behavior, this can act as a basis for making work allocation decisions in real life.

In this paper, we use linear regression analysis to quantify the "Yerkes-Dodson Law of Arousal" based on empirical data. *Linear Regression Analysis* is a statistical technique used for investigating and modeling the relationship between variables [7]. We use regression because of its various purposes, i.e., it can be used to describe and summarize a dataset through the regression equations, it can be used for prediction of the response variable based on the predictor variables, the variables in a regression model are usually related in a cause-effect relationship and so regression can be used in confirming such a relationship and also regression is a useful technique for parameter estimation. Although we use linear regression in this paper, there are more powerful regression techniques that can be used to truly capture the U-shape shown in Figure 1.

The remainder of the paper is organized as follows. First, we provide an overview of event logs and process mining in Section 2. Section 3 has a discussion of workload-dependent processing speeds. We explain how to extract the workload and processing speeds based on the information available in event logs in Section 4. In Section 5, we describe the application of our approach to a case study based on real-life logs to validate our approach. Section 6 has a discussion of related work and finally Section 7 gives conclusions.

2 Process Mining: An Overview

2.1 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 contains log entries about activities executed for a business process. We assume that it is possible to record events such that each event refers to an activity and

is related to a particular case (i.e., a process instance). For any process mining technique, an event log is needed as the input. In order to understand what an event log is, we define the concept of an *event*.

Definition 1 (Event, Property) Let $\mathcal{E}$ be the event universe, i.e., the set of all possible event identifiers, and $\mathcal{T}$ the time domain. We assume that events have various properties, e.g., an event has a timestamp, it corresponds to a particular activity, is executed by a particular resource and has a particular type. For each of these properties, there are functions $prop_{time} \in \mathcal{E} \rightarrow \mathcal{T}$ assigning timestamps to events, $prop_{act} \in \mathcal{E} \rightarrow \mathcal{A}$ assigning activities to events, $prop_{type} \in \mathcal{E} \rightarrow \{start, complete\}$ assigning event types to the events, and $prop_{res} \in \mathcal{E} \nrightarrow \mathcal{R}$ is a partial function assigning resources to events. For $e \in \mathcal{E}$, we define $\overline{e}$ as a shorthand for $prop_{time}(e)$, i.e., the time of occurrence of event e.

An event e is described by some unique identifier and can have several properties. In this paper, we use these properties which are; the timestamp of an event ($prop_{time}(e)$), the activity name ($prop_{act}(e)$), the name of the resource that executed the activity ($prop_{res}(e)$) and the event type of the activity ($prop_{type}(e)$). Note $prop_{res}$ is a partial function because some events may not be linked to any resource.

An event log is a set of events. Each event in the log 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 furthermore for each trace, time should be non-decreasing within each trace in the log.

Definition 2 (Event Log and Trace) 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 $1 \leq$ i $<$ j $\leq |\sigma| : \sigma(\text{i}) \neq \sigma(\text{j})$ 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 $\sigma_1 = \sigma_2$.

Note that $\overline{\sigma(i)} \leq \overline{\sigma(j)}$ means that time is non-decreasing (i.e., $prop_{time}(\sigma(i)) \leq prop_{time}(\sigma(j))$ if i occurs before j). The last requirement states that σ_1 and σ_2 should not have any overlapping events. This is done to ensure that events are globally unique and do not appear in multiple traces.

Table 1 shows a fragment of an event log with two traces and each trace consists of a number of events. For example, the first trace has three events ($1a$, $1b$, $1c$) with different properties. For event $1a$, $prop_{act}(1a)$ =A, $prop_{res}(1a)$ =Mary, $prop_{time}(1a)$ =20th November 2007 at 8:00am and $prop_{type}(1a)$ =start.

2.2 Process Mining

Process mining aims at the extraction of information from a set of real executions (event logs). As already stated, event logs are the starting point for any process mining technique. Before any technique can be applied to the event log, information can directly be obtained from the log through the *preprocessing* step.

Table 1. A fragment of an event log.

event	properties			
	activity	**resource**	**timestamp**	**type**
1a	A	Mary	20-11-2007:8.00	start
1b	A	Mary	21-11-2007:8.13	complete
1c	B	John	01-12-2007:8.16	start
2a	A	Angela	08-02-2008:8.10	start

This information can include the number of traces and events in the log, the activities and resources, and the frequency of their occurrences in the log, etc. Based on this information *log filtering* can be done, for example, to remove the resources with infrequent occurrence. After this step, then process mining techniques can be applied to the log to discover three different perspectives (process, organizational, case) through the *processing* step.

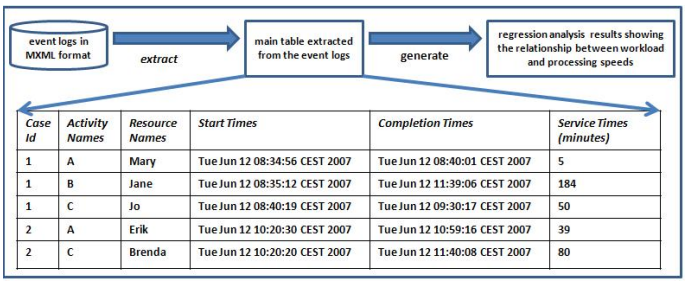
The *process* perspective focusses on the control-flow, i.e., the ordering of activities and the goal here is to find a good characterization of all the possible paths, e.g., expressed in terms of a Petri net [2]. The *organizational* perspective focusses on the resources, i.e., which performers are involved in the process model and how are they related. The goal is to either structure the organization by classifying people in terms of roles and organizational units or to show relation between individual performers (i.e., build a social network [11]). The *case* perspective focuses on properties of cases. Cases can be characterized by their paths in the process or by the values of the corresponding data elements, e.g., if a case represents a supply order it is interesting to know the number of products ordered. Orthogonal to these three perspectives, the result of a mining effort can refer to performance issues. For example, information about flow times and waiting times. The discovered process model can then be enhanced with this performance information.

3 Workload-Dependent Processing Speeds

In many systems, the speed at which resources work is partly determined by the amount of work at present. This is especially true for human beings; in busy periods people tend to increase their speed in order to process more cases. However, when people are given too much work over a long period of time, their performance then tends to drop. This phenomenon is known as the "Yerkes-Dodson Law of Arousal" [12] and is illustrated by the inverse U-Shaped curve depicted in Figure 1. If the law holds, the performance of people (i.e., the speed at which they work) is determined by the workload that is currently present in the system [8]. An example would be a production system where the speed of a server is relatively low when there is too much work (stress) or when there is very little work (laziness) [3].

In this paper, we discuss a new process mining technique implemented in our Process Mining framework (ProM), to quantify the relationship between

workload and processing speeds based on historic data. From the event logs expressed in standard Mining *XML* (MXML) format [4], we extract information about traces, the activities per trace, the resources that execute these activities, and their respective service times (this is measured in minutes and is explained in Section 4.2).



Case Id	Activity Names	Resource Names	Start Times	Completion Times	Service Times (minutes)
1	A	Mary	Tue Jun 12 08:34:56 CEST 2007	Tue Jun 12 08:40:01 CEST 2007	5
1	B	Jane	Tue Jun 12 08:35:12 CEST 2007	Tue Jun 12 11:39:06 CEST 2007	184
1	C	Jo	Tue Jun 12 08:40:19 CEST 2007	Tue Jun 12 09:30:17 CEST 2007	50
2	A	Erik	Tue Jun 12 10:20:30 CEST 2007	Tue Jun 12 10:59:16 CEST 2007	39
2	C	Brenda	Tue Jun 12 10:20:20 CEST 2007	Tue Jun 12 11:40:08 CEST 2007	80

Fig. 2. Overview of the approach. First an event log in MXML format is translated into a tabular format showing (a) case id's, (b) activity names, (c) resource names, (d) start times, (e) completion times, and (f) service times (difference between the completion and start times). This table is then used to calculate the workload and regression analysis is carried out to find the relationship between workload and processing speeds. This can be done at different levels of granularity, e.g., per activity, per resource, or per resource/activity combination.

Figure 2 shows a sample table of the basic information extracted from the event logs. Based on this information, we determine the workload and processing speeds. As will be shown in the next section, multiple definitions of the workload are possible. This workload information can be compared with the actual service times (last column in the main table shown in Figure 2), i.e., the time required to execute an activity (thus denoting the processing speed). Then using linear regression analysis, we quantify the relationship between the workload and the processing speeds. In the next section, we describe in detail how the workload and processing speeds are defined and measured based on the information in the event log.

4 Relationship Between Workload and Processing Speeds

4.1 Workload

As already stated, people do not work at constant speeds and their processing speeds are often influenced by the current workload.

The workload of a resource or a group of resources can be defined as either: (a) the number of work items waiting at the start of execution of an activity, i.e., the amount of work that has been scheduled for a given user or (b) the number of activities that have been executed over a particular period. In this

paper, we focus on the second option, i.e., the number of activities that have been executed over a particular period defines "how busy" the resource has been. We now define the notion of workload used in this paper.

Definition 3 (Workload) Let $\mathcal{T}$ be the time domain, $\mathcal{C}$ be a set of all possible traces, $L \subseteq \mathcal{C}$ be an event log, and $\mathcal{E}$ be a set of all possible event identifiers.

- We define the *event universe* of L as $\mathcal{E}_L = \{e \in \mathcal{E} \mid \exists_{\sigma \in L}\ e \in \sigma\}$.
- $\mathcal{E}_L$ is partitioned into two sets: $\mathcal{E}_L^s = \{e \in \mathcal{E}_L \mid prop_{type}(e) = start\}$ (i.e., all start events in L) and $\mathcal{E}_L^c = \{e \in \mathcal{E}_L \mid prop_{type}(e) = complete\}$ (i.e., all complete events in L).
- The workload calculation based on L is parameterized by the following four parameters: $\mathcal{E}_{ref}$, $\mathcal{E}_{load}$, h_{back}, and h_{forw}.
- $\mathcal{E}_{ref} \subseteq \mathcal{E}_L$ is the set of *reference events,* i.e., the events for which the workload is calculated.
- $\mathcal{E}_{load} \subseteq \mathcal{E}_L$ is the set of *load events,* i.e., the events considered when calculating the workload.
- $h_{back} \in \mathcal{T} \rightarrow \mathcal{T}$ is a function that defines the *start* of the time window given some reference time, i.e., for some reference time $t \in \mathcal{T}$, the time window starts at $h_{back}(t)$ (with $h_{back}(t) \leq t$).
- $h_{forw} \in \mathcal{T} \rightarrow \mathcal{T}$ is a function that defines the *end* of the time window given some reference time, i.e., for some reference time $t \in \mathcal{T}$, the time window ends at $h_{forw}(t)$ (with $t \leq h_{forw}(t)$).
- Based on L, $\mathcal{E}_{ref}$, $\mathcal{E}_{load}$, h_{back}, and h_{forw}, we define the workload function $busy \in \mathcal{E}_{ref} \rightarrow \mathbb{N}$, where $\mathbb{N}$ is the set of natural numbers $\{0, 1, 2, ..\}$ as follows: $busy(e) = |\{e' \in \mathcal{E}_{load} \mid h_{back}(\overline{e}) \leq \overline{e'} \leq h_{forw}(\overline{e})\}|$, i.e., the number of load events in the time window associated with a reference event e.

Function *busy* calculates the workload for all the reference events. An event e is a reference event, i.e., $e \in \mathcal{E}_{ref}$, if it can be associated to some service time. For example, one can take $\mathcal{E}_{ref} = \mathcal{E}_L^s$, i.e., all start events are reference events and by looking up the corresponding complete events it is possible to measure their service times. It is also possible to take $\mathcal{E}_{ref} = \mathcal{E}_L^c$ or even $\mathcal{E}_{ref} = \mathcal{E}_L$. In the later case there are two reference events for each activity. Based on the timestamp of some reference event $e \in \mathcal{E}_{ref}$, we calculate a time window that starts at $h_{back}(\overline{e})$ and ends at $h_{forw}(\overline{e})$. Note that the time window depends on the definition of the parameters h_{back} and h_{forw}. For example, if $h_{back}(t) = t - a$ and $h_{forw}(t) = t + b$, then events that occurred less than a time units before some reference event and not more than b time units after some reference event are considered. When the values chosen for a and b are long (i.e., in our approach a and b are between 1 to 24 hours), then we see a greater effect of the workload on the processing speed. Based on such a time window, function *busy* then simply counts the number of load events. The set of load events may be defined as $\mathcal{E}_{load} = \mathcal{E}_L^s$. It is also possible to take $\mathcal{E}_{load} = \mathcal{E}_L^c$ or even $\mathcal{E}_{load} = \mathcal{E}_L$.

Definition 3 looks at $\mathcal{E}_{ref}$ and $\mathcal{E}_{load}$ for the log as whole. However, it is possible to determine these sets of events *per activity*, *per resource*, or *per activity/resource combination.*

4.2 Processing Speeds

In this section, we define the processing speeds based on the information in the logs. The processing speeds can be defined as either the *flow time* (i.e., the time required to handle a case from beginning to end) or the *service times* (based on the actual processing time of individual activities). In this paper, we only consider the service times as a way of denoting the processing speeds. Given that we have the *start* and *complete* events of an activity recorded in the log, the service time is defined as the difference between the times at which these two events were executed.

Definition 4 (Service Time) Let L, $\mathcal{E}_L$, $\mathcal{E}_L^s$ and $\mathcal{E}_L^c$ be as defined in Definition 3. Function $st \in \mathcal{E}_L \rightarrow \mathcal{T}$ maps events onto the duration of the corresponding activity, i.e., the service time. We assume that there is a one-to-one correspondence between $\mathcal{E}_L^s$ and $\mathcal{E}_L^c$, i.e., any $e_s \in \mathcal{E}_L^s$ corresponds to precisely one event $e_c \in \mathcal{E}_L^c$ and vice versa. The service time of these events are equal, i.e., $st(e_s) = st(e_c) = \overline{e_c} - \overline{e_s}$.

Note that the above definition heavily relies on the assumption that there is a one-to-one correspondence between start and complete events. When reading the traces in the log, there are situations when for an activity only the *start* event is recorded and not the *complete* event or when the *complete* event is recorded and not the *start* event for the same activity. In order to avoid the recording of incorrect durations, we match the *start* and *complete* events by linking events that belong to the same trace and for which the activity names are the same. Events which can not be matched are discarded. Moreover, we have heuristics to determine when events were started based entirely on the complete events.

After obtaining the workload and the service times, we use simple linear regression analysis to find the relationship between workload (as the *independent variable*) and processing speed (as the *dependent variable*). In this case, we have one independent variable and one dependent variable, however, it is easy to add other independent variables (e.g., based on alternative definitions of workload). From the analysis we obtain parameters required for the construction of the regression equation given by: $y = \beta_0 + \beta_1 x + \varepsilon$ where: y is the dependent variable (processing speed expressed in terms of the service time, i.e., $st(e)$), x is the independent variable (workload, i.e., $busy(e)$), β_0 (intercept) is the value of y when $x = 0$, β_1 (slope) is the change in y produced by a unit change in x, ε is the error of prediction obtained using the regression equation.

Other parameters can also be obtained from the regression analysis which are; the *correlation coefficient* (r) is the degree to which two variables are linearly related ($-1 \leq r \leq 1$) and *r-square of the regression equation* (R^2, or the coefficient of determination), which is the proportion of variation in y accounted for by x ($0 \leq R^2 \leq 1$). Higher values of R^2 ($0.7 \leq R^2 \leq 1$) indicate a good fit of the regression equation to the data while the intermediate values ($0.5 \leq R^2 \leq 0.7$) show a moderate fit and low values ($0 \leq R^2 \leq 0.5$) indicate a poor fit. The approach described in this paper is implemented as a plug-in in the process mining

tool ProM. In the next section, we discuss the results from the application of this approach to real-life logs.

5 Experiments

We tested our approach and the implemented ProM plug-in on a real case study based on a process that handles the getting of building contracts in a Dutch municipality.

5.1 Case Study

The case study was conducted on real-life logs from a municipality in the Netherlands. This municipality uses a workflow system and the logs used are from a process that deals with the getting of a building permit. Through the preprocessing step we obtained important information about the log. The event log contains information about 2076 cases, 67271 events, 109 resources and 11 activities. The start date of the log is "2003-01-24" and the end date is "2005-11-08". We filtered the log to remove the resources and activities with infrequent occurrence and also only considered the events with both the *start* and *complete*. The information contained in the main table (as shown in Figure 2), can be viewed based on three perspectives, i.e, the resource, activity and resource/activity perspectives.

Table 2. Linear regression results based on the resource dimension.

resource names	**correlation coefficient (r)**	**R^2**	**intercept (β_0)**	**slope (β_1)**
jcokkie	0.44	0.19	22053	7860
bfemke	0.68	0.46	-20502	38537
klargen	0.84	0.71	-585057	704292
mbree	0.68	0.47	-1264	3849
clijfers	0.22	0.05	11850	21920
pkelders	0.17	0.03	1619	115.8
bgeveren	0.73	0.53	-299007	355963

Tables 2 and 3 show the linear regression results based on the *resource perspective* and the *resource/activity perspective* respectively[1]. After filtering events from the main table, based on the resource perspective, we select the events to use for the reference and load events. In this case study, the *complete*[2] events are selected and also $h_{forw}(t) = t+23hrs$ and $h_{back}(t) = t+23hrs$ where t is the time of execution of a reference event. The result of the relationship between workload and processing speed is reflected by the r and R^2 values. For example, resource "klargen" in row three of Table 2, has high positive values for r and R^2. This

[1] The resource names in Tables 2 and 3 have been changed to ensure confidentiality.
[2] Although we selected the *complete* events for the reference and load events, we could have also chosen the *start* events or both the *start* and *complete* events.

implies that "how busy" this resource has been in the past affects the speed at which he executes activities. Both tables also show the slope and intercept values which are used in the regression equation. For example, the regression equation for "klargen" in Table 2 is: *processing speed* = -585057.5 + 704292(*workload*), i.e., $\beta_0 = -585057.5$ and $\beta_1 = 704292$ in $y = \beta_0 + \beta_1 x + \varepsilon$. The results obtained in Table 2 are based on all the activities that the resources executed over the whole log. We point out that in real-life resources can be involved in multiple processes yet the event log records events for one particular process in isolation that a resource may be involved in. Hence the resource utilization is low in these logs. This affects the values obtained for r and R^2 (they are not as high as they may have been expected).

Table 3. Linear regression results for the resource/activity dimension. For example, for the fifth row "jcokkie" is the resource name and "CTT" is the activity name.

resource & activity names	**correlation coefficient (r)**	**R^2**	**intercept (β_0)**	**slope (β_1)**
pbakere/Publiceren	0.99	0.99	-14559.3	25824.7
pbakere/AR03Arcdossier	0.98	0.99	-612530	742325.5
jcokkie/BV99Convdoss	0.99	0.98	-14037.7	99539
jcokkie/CTT	0.78	0.61	-139809	86795
jcokkie/AR03Arcdossier	0.99	0.99	354495	258812.5
clijfers/BV26Financion	0.65	0.43	-41275.8	46161.6
clijfers/BV24Afwerkbesch	0.99	0.99	-129321	131731.7
clijfers/BV36W0Z	0.79	0.63	-263634	266631.2
nlijslet/BV26Bouwcontrole	0.97	0.95	-97185.4	102766.2
pkelders/BV06Milieu	0.73	0.53	-21966	2059.2
pkelders/BV29Gereed	0.99	0.99	-6940	6940
pkelders/BV28Gestat	0.57	0.30	-4961	4961
hwyman/BV26Belastingen	0.97	0.94	-9544.5	10640.5
groemer/BV24Afwerk	0.77	0.59	-76566	84550.7
dtruyde/BV06CCT	0.92	0.86	-263933	273645

To obtain the results shown in Table 3, we filter the log based on the resources to get the activities that each resource executes and the events per activity are used for obtaining the workload. Several values for R^2 in this table are greater than 0.7 which is a strong indication that most of the variability in the processing speeds is explainable by the workload. For example, for "pbakere&Publiceren" in row 1 of Table 3, $R^2 = 0.99$ which implies that 99% of the variability in the processing speed is dependent on the workload for this resource. We also point out that, although for some resources there is no significant relationship when all the activities they executed are considered (see Table 2) as reflected by the low r and R^2, there is a significant relationship when the individual activities are considered as reflected by the high r and R^2 values (see Table 3). For example, resource "jcokkie" in the first row of Table 2 has values of $r = 0.44$ and $R^2 = 0.19$, whereas in Table 3, in row 5 "jcokkie & AR03 Arcdossiers" with values of

$r = 0.99$ and $R^2 = 0.99$ and in row 4 "jcokkie & CTT" where $r = 0.78$ and $R^2 = 0.61$. These examples indeed suggest that the speed at which people work is indeed influenced by their workload.

6 Related Work

The work presented in this paper is related to earlier work on process mining and operations management. Recently many tools and techniques for process mining have been developed [2, 11]. Note that process mining is not restricted to control-flow discovery [2]. For example, in [11] the main aim is to build organizational models from event logs and analyze relationships between resources involved in a process.

The "Yerkes-Dodson Law of Arousal" [12] illustrated in Figure 1, is one of the main motivations for this paper. In operations management, substantial work has been done to operationalize this "law" using mathematical models and simulation in order to explore the relationship between workload and shop performance [3]. In [8] queues with workload-dependent arrival rates and service rates are considered. The authors of these papers investigate what the effect on production efficiency is based on controlling the arrival rates and service rates as a result of the workload present in the system. Juedes et al. [6] introduce the concept of workload-dependent processing speeds in real-time computing. In this study, they deal with a maximum allowable workload problem for real-time systems with tasks having variable workload sizes.

The related work mentioned above does not actually measure the relationship between workload and service times. This paper has presented such an analysis technique based on linear regression analysis. This is supported by a new plug-in in ProM and has been applied to several examples. We are not aware of other studies that try to discover phenomena such as the one described by the "Yerkes-Dodson Law of Arousal".

7 Conclusion

Although organizations use various analysis techniques to analyze their business processes, the results may be very misleading if the assumptions used are incorrect. For example, in most simulation tools service times are simply sampled from a probability distribution without considering the workload. In this paper, we presented an approach to quantify the relationship between workload and processing speed. This approach is based on regression analysis and is implemented as a new plug-in in ProM.

We consider this as a first step approach in the use of process mining techniques for the extraction of useful information from event logs that characterizes resource behavior and also as an addition to the repertoire of process mining techniques. We expect that process mining techniques will focus more and more on the behavior of workers once it becomes easier to discover processes.

Experimentation shows that the relationship described by the "Yerkes-Dodson Law of Arousal" really exists. However, to truly capture the inverse U-shape depicted in Figure 1, we need more sophisticated regression techniques. In this paper, we focus on the definition of workload as the number of work items that have been executed over a particular period, but there other workload definitions that are possible and can be explored. Our future research will aim at more powerful analysis techniques and a tight coupling between simulation and operational decision making. As discussed in [1], we want to make simulation more realistic by adequately modeling resources based on empirical data. Besides workload-dependent process times, we also take into account that people are involved in multiple processes, are available only part-time, work in batches. Experiments show that these factors really influence performance [1].

References

1. van der Aalst, W.M.P., Nakatumba, J., Rozinat, A., Russell, N.: Business Process Simulation: How to get it Right? In: vom Brocke, J., Rosemann, M. (eds.) International Handbook on Business Process Management. Springer, Berlin (2008)
2. van der Aalst, W.M.P., Weijters, A.J.M.M., Maruster, L.: Workflow Mining: Discovering Process Models from Event Logs. IEEE Transactions on Knowledge and Data Engineering. 16(9), 1128-1142 (2004)
3. Bertrand, J.W.M., van Ooijen, H.P.G.: Workload Based Order Release and Productivity: A Missing Link. Production Planning and Control 13(7), 665-678 (2002)
4. 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)
5. 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)
6. Juedes, D., Drews, F., Welch, L.: Workload Functions: A New Paradigm for Real-time Computing. In: 10th IEEE Real-Time and Embedded Technology and Applications Symposium Work-In Progress Session, pp. 25-28 (2004)
7. Montgomery, D.C., Peck, E.A.: Introduction to Linear Regression Analysis. Wiley & Sons (1992)
8. van Ooijen, H.P.G., Bertrand J.W.M.: The effects of a simple arrival rate control policy on throughput and work-in-progress in production systems with workload dependent processing rates. International Journal of Production Economics vol. 85, pp. 61-68 (2003)
9. Rozinat, A., Mans, R.S., Song, M., van der Aalst, W.M.P.: Discovering Simulation Models. Information Systems 34(3), 305-327 (2009).
10. Rozinat, A., Wynn, M.T., van der Aalst, W.M.P., ter Hofstede A.H.M., Fidge, C.: Workflow Simulation for Operational Decision Support Using Design, Historic and State Information. In Dumas, M., Reichert, M., Shan, M.C.(eds.) BPM 2008. LNCS, vol. 5240, pp. 196-211. Springer, Heidelberg (2008)
11. Song, M., van der Aalst, W.M.P.: Towards Comprehensive Support for Organizational Mining. Decision Support Systems 46(1), 300–317 (2008).
12. Wickens, C.D.: Engineering Psychology and Human Performance. Harper (1992)