Resource Utilization Prediction in Decision-Intensive Business Processes*

Simon Sperl[1], Giray Havur[1,2], Simon Steyskal[1,2], Cristina Cabanillas[2], Axel Polleres[2], and Alois Haselböck[1]

[1] Siemens AG Österreich, Corporate Technology, Vienna, Austria
`{name.surname}@siemens.com`
[2] Vienna University of Economics and Business, Austria
`{name.surname}@wu.ac.at`

Abstract. An appropriate resource utilization is crucial for organizations in order to avoid, among other things, unnecessary costs (e.g. when resources are under-utilized) and too long execution times (e.g. due to excessive workloads, i.e. resource over-utilization). However, traditional process control and risk measurement approaches do not address resource utilization in processes. We studied an often-encountered industry case for providing large-scale technical infrastructure which requires rigorous testing for the systems deployed and identified the need of projecting resource utilization as a means for measuring the risk of resource under- and over-utilization. Consequently, this paper presents a novel predictive model for resource utilization in decision-intensive processes, present in many domains. In particular, we predict the utilization of resources for a desired period of time given a decision-intensive business process that may include nested loops, and historical data (i.e. order and duration of past activity executions, resource profiles and their experience etc.). We have applied our method using a real business process with multiple instances and presented the outcome.

Key words: decision-intensive business processes, prediction model, resource utilization, risk management

1 Introduction

Human resource utilization in an organization can be seen as the proportion of time a person spends on working on allocated tasks. Poor utilization of human resources[2] relates to having resources unnecessarily idle or overloaded. This has a very negative effect on process performance measures such as process completion time, execution costs, and quality [1]. Specifically, while under-utilization of resources leads to higher process execution costs, over-utilization of resources may

* This work has been funded by the Austrian Research Promotion Agency (FFG) under the project grant 845638 (SHAPE) and the Austrian Science Fund (FWF) under the project grant V 569-N31 (PRAIS).

[2] From now on *resources* for the sake of brevity.

result in process delays. Therefore, decision makers (typically process managers) should be informed about the utilization of resources in their organizations for enabling appropriate controls that ensure a desired level of resource utilization.

In a scenario where the process model has no decision nodes, given a baseline schedule and resource allocation [2], deriving the utilization of resources would be trivial. However, actual processes usually have decision points that split the execution flow into different paths so that several cases are projected in the same process model [3]. Decision-intensive processes may contain (nested) loops [4], which actually makes both scheduling and resource allocation more difficult due to the increasing uncertainty. Nonetheless, these kind of processes are common in many domains, e.g. engineering, healthcare, insurance handling, and construction. An inadequate scheduling or allocation of resources may result in a poor resource utilization. While recent resource allocation approaches have already tried and addressed that kind of processes [2], to the best of our knowledge there is a lack of support for resource utilization prediction in such complex scenarios, as most of the existing techniques tend to simplify the application scope [5, 6, 7, 8]. This, in turn, negatively affects risk management in organizations, since process managers miss input that helps to improve the process models and hence, the execution performance of the processes.

In this paper we address that problem and describe a mathematical method for quantifying resource utilization with respect to the structural properties of non-deterministic processes and the historical executions of these processes. The input values that are used for our prediction model are intrinsically of a stochastic nature, i.e. in the form of probability density functions (PDF). They include, among others, the activity duration PDFs and the resource utilization PDFs. We propagate these input values towards the accumulated utilization function. This function provides a visual overview on the level of future resource utilizations. Therefore, an upcoming over- or under-utilization of resources can be observed. Moreover, we have defined two metrics which characterizes resource over-utilization and under-utilization. We have implemented our approach and demonstrated it with a real process related to large-scale technical infrastructure development and deployment.

We believe that our approach enhances the assessment of process behaviour with the resource perspective. It is especially useful for organizations who need to evaluate the utilization of resources for which they are accountable. With the help of our approach they can automatically get an answer to questions such as whether they have enough resources for the robust execution of their processes, in which periods of time they should expect a delay in the process execution, and when they can safely grant vacations to particular resources, among others.

The paper is structured as follows: Section 2 presents a scenario that motivates this work as well as related work. Section 3 formally defines the input required for our approach. Section 4 describes our approach for predicting resource utilization in decision-intensive business processes. Section 5 applies our method to a real process and presents the outcome, and Section 6 concludes the paper and outlines the future work.

2 Background

In the following, we describe an example scenario that motivates this work and shows the problems to be addressed, and then we outline related work.

2.1 Running Example

A company that provides large-scale technical infrastructure requires rigorous testing for the systems deployed. Each system consists of different types and number components that are developed and tested in parallel. In order to provide concise and clear examples while describing our method, we use the simple example process shown in Fig. 1. In this process, the *Develop* activity is followed by a *Test* activity, and this may repeat if the test fails. There is also the activity *Manage* which abstracts the potential contractual work running in parallel. There are two resources, namely *Jack* and *Jill*, who execute this process. A more complex process from a real scenario is used later in Section 5 for demostrating the applicability of our method.

2.2 Related Work

The approach presented in this paper is mostly related to resource-related risk monitoring and prediction. This risk occurs "due to the high variability that may affect operational processes in real world scenarios" [9]. The risk of inappropriate resource utilization has been identified in [10, 11]. Rosemann et. al. [10] provides a risk taxonomy from the project management perspective. They identify the organizational risk in their taxonomy (e.g., when a resource does not possess the required skills to carry out an activity, or when there is not enough resources to carry out activities on time). Similarly, the process related risk taxonomy of zur Muehlen et. al. [11] contains the resource perspective catagory about lack of resources and/or their skills for activity executions.

Information systems support processes by recording information about their executions in event logs [12]. In order to manage time and resource related risks in an *informed* fashion, a variety of event log mining and prediction mechanisms have been devised: Among others, process duration estimation [13, 14], deadline violation detection [6], resource profiling [7], resource behaviour measurement [5, 15], resource scheduling [16], resource recommendation [17], work prioritization [18], and process performance forecasting [19]. There are also several risk monitoring approaches [8, 20] combines the scope of several aformentioned mechanisms.

Our method requires extracting durations of activities, and experience of resources for each activity from the event logs for providing realistic resource utilization predictions. Durations can be obtained in a similar way to described by van der Aalst et al. [13] who provide reliable predictions. On the other hand, extraction of resource profiles including their experiences are delineated by Pika et. al. [7].

Within the context of quantifying the resource perspective, our method draws parallels with [15]. We further elaborate our mathematical model for refining our results over the structural properties of running processes. Rather than providing an overall view, Conforti et. al. [8] allow users to take resource-informed decisions at run-time. Our approach is similar to this work in the sense that it reports on resource utilization abnormalities thay may become a problem during the executions of processes.

Folino et al. [19] introduce a performance model for run-time process executions with respect to process variants such as workload and seasonality. Our prediction method can support such models for enriching the context from resource utilization point-of-view.

3 Input Data

The following input is required for resource utilization prediction in decision-intensive business processes:

Input 1 (**Business Process Model**). A business process model P is represented as a directed, connected graph (N, E) with $N = A \cup G \cup \{n_{start}, n_{end}\}$ denoting a finite set of nodes consisting of activities A, gateways G, and two respective start and end events n_{start} and n_{end}, and $E \subseteq N \times N$ representing a set of edges connecting the nodes.

We assume that our model consists of activities, XOR-gateways (decision points), AND-gateways (parallel execution gateways), a start-event, and an end-event. The process always terminates (i.e., it contains no livelocks or deadlocks).

In a real life application, an input process can be composed of the processes that are planned to be executed in the future, and of the unexecuted fragments of the already executing processes.

Input 2 (**Edge Execution Probability**). Given a process model $P = (A \cup G \cup \{n_{start}, n_{end}\}, E)$, each edge $e \in E$ leaving a XOR-gateway $g \in G_{XOR} \subseteq G$ of P is annotated with an edge execution probability $p_e \in [0, 1]$. Additionally edge execution probabilities p_e must satisfy $\sum\limits_{e \in E \cap (g \times N)} p_e = 1,\ \forall g \in G_{XOR}$.

The outgoing edges of XOR-gateways are annotated with edge execution probabilities (see Fig. 6).

Input 3 (**Activity Duration PDF**). For each activity $a \in A$, $D_a : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ denotes the PDF representing the duration PDF of activity a.

Example 1. For clarification, we provide the process in Fig. 1. A single loop of two consequent activities *Develop* and *Test* is presented in parallel with the activity *Manage*. The duration PDFs for *Manage* and *Develop* are normally distributed while *Test* has a fixed duration represented as a solid dot (i.e., a time shifted dirac delta function [3]). The duration of *Test* is *deterministically* 3 time-units (TU) whereas the duration of *Manage* is *on average* 2 TU.

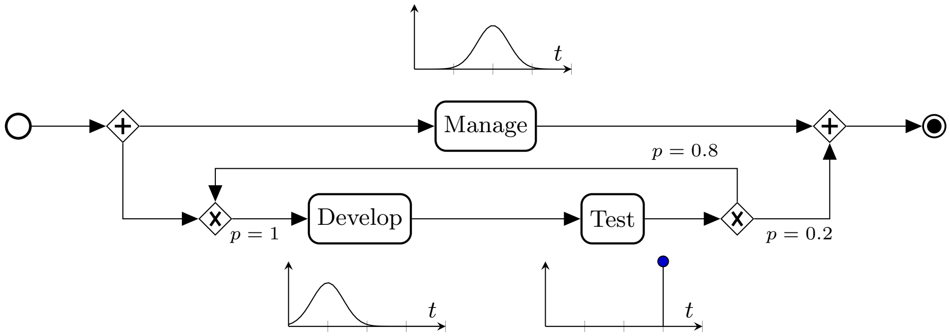


Fig. 1: The running example process with activity duration PDFs D_a

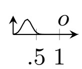
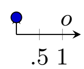
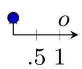
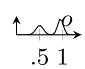
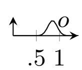
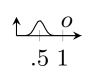
Activities	Resources: Jack	Resources: Jill	Description
Manage			The activity *Manage* utilizes Jack *about* 30%.
Develop			The activity *Develop* utilizes Jill *about* 100%, however there is also less of a chance that the same activity may utilize her *about* 50%.
Test			The activity *Test* utilizes Jack *about* 85% and Jill *about* 50%.

Fig. 2: Resource Utilization PDFs for the activities in Fig. 1

Input 4 (**Resource Utilization PDF**). Given a set of activities A of a business process B and a set of resources R, the resource utilization PDF is $U_{a,r} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ for activity $a \in A$ and resource $r \in R$. U defines the PDFs of the probable additional utilization the execution of an activity causes for all resources.

The resource utilization PDFs describe which resources are utilized to what extent while executing an activity. Intuitively, one can think of $U_{a,r}(x)dx$ as being the probability of r's utilization falling in the infinitesimal interval $[x, x + dx]$. We assume that utilization values are normalized so that $U_{a,r}(0)$ represents the probability of resource r being 0% utilized by the activity a. $U_{a,r}(1)$ specifies the probability of resource r's being utilized 100% by the activity a. The notion of utilization can be considered as "the percentage of the work day spent on a task" in our running example.

Example 2. See Fig. 2 for an example visualizing the definition of resource utilization PDF. For instance, $U_{Develop,Jill}(0.5) = 0.3$ means with 0.3 probability

[3] The Dirac delta function $\delta(x)$ is ∞ at $x = 0$ otherwise 0 and satisfies $\int_{-\infty}^{\infty} \delta(x)\,\mathrm{d}x = 1$.

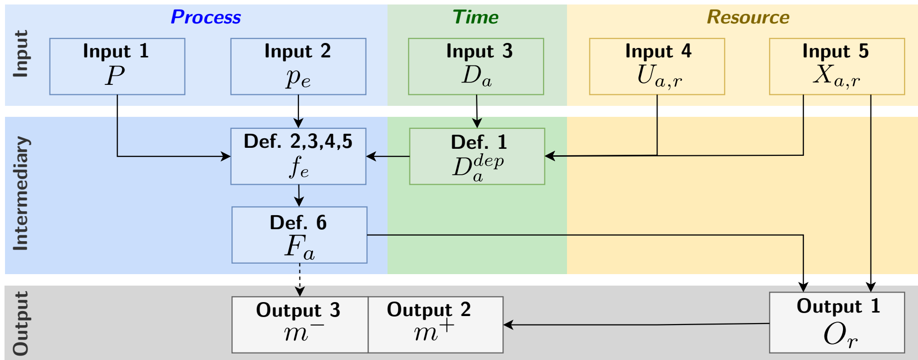


Fig. 3: Input, intermediary functions, and output of our method

Develop utilizes Jill 50%, and $U_{Develop,Jack}(0) = \infty$ means that Jack is never occupied by *Develop*.

Input 5 (**Experience Matrix**)**.** Given a set of activities A of a process model, and a set of resources R, the experience value $X_{a,r} : \mathbb{R}_{>0}$ where $a \in A$ and $r \in R$ is a multiplication factor (a scalar) for activity durations.

The *experience* of each resource r in every activity a is reflected in this matrix. This value theoretically has a range between zero to infinity which is extracted from activity execution durations of resources. Resources that execute activities faster than average have an experience value greater than 1.0.

We assume that the edge execution probabilities p_e, activity duration PDFs D_a, resource utilization PDFs $U_{a,r}$, and experience matrix X (*Input 2–5*) are extracted from the event logs obtained from the past executions of the process P (Input 1), where the resources and the durations of the past activity executions are recorded. A solution to this prediction problem is *total utilization PDFs over time* for each resource. Each function provides a utilization prediction for its respective resource.

4 Method

Following the problem definition, we first introduce intermediary functions that would allow us to compute the ultimate total utilization PDF, and afterwards we introduce quality metrics that would quantify the risk of abnormal resource utilization that may occur in the future. Fig. 3 is an overview of "*used by*" relation between input values and defined functions of our method (e.g., *Input 3* is used by *Definition 1*). Background colors blue, green and yellow indicates process-related, time-related and resource-related elements.

Definition 1 (Dependent Activity Duration PDF). *Given an independent activity duration PDF D_a, allocation PDF $U_{a,r}$, experience values $X_{a,r}$ for activities a and resources r, the (utilization-and-experience) dependent activity duration PDF $D_a^{dep} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is defined as follows:*

$$x_a = \sum_{r \in R} X_{a,r} \int_0^\infty u U_{a,r}(u) du$$

$$D_a^{dep}(t) = \frac{D_a(t x_a)}{\int_0^\infty D_a(t' x_a) dt'}$$

For each activity, an *experience* value x_a is derived from the experiences of all the resources that are potentially participants in a (i.e., $\exists x \in \mathbb{R}_{>0} : U_{a,r}(x) > 0$). x_a is used as a division factor for D_a, therefore the experience values above 1.0 reduce the width of D_a (i.e., they act as speed-up factor for execution times), and the opposite holds for the values smaller than 1.0.

4.1 Edge Transition Probability Function

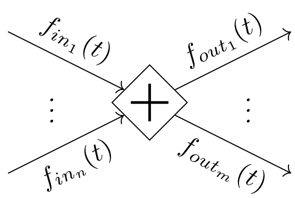
In order to compute how often an activity is being executed, we need to extract the probability values of edge traversals over time. Only activities can create time delays via their duration PDFs. Edge transitions are always instantaneous.

Definition 2 (Edge transition probability function). $f_e(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *denotes the probability that an edge* $e \in E$ *is traversed at time* $t \in \mathbb{R}_{\geq 0}$.

$$\begin{aligned} &P(\textit{An edge is traversed between } t_1 \textit{ and } t_2) \\ &= P(\textit{Edge is traversed before } t_2) - P(\textit{Edge is traversed before } t_1) \\ &= \int_0^{t_2} f(t')\, dt' - \int_0^{t_1} f(t')\, dt' \end{aligned}$$

Note that in general f_e is not a PDF. However, if the process contains no XOR-gateways all f_e are PDFs since $\forall e \in E : \int f_e(t)\, \mathrm{d}t = 1$.

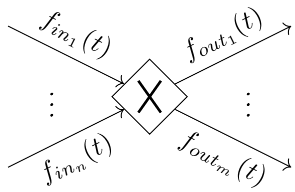
Definition 3 (f for AND-Gateway). *Given* $g \in G_{and} \subseteq G$ *with incoming edges* $\{in_1, \ldots, in_n\} = E \cap (N \times g)$ *and outgoing edges* $\{out_1, \ldots, out_m\} = E \cap (g \times N)$, *the edge transition probability function* $f_e(t)$ *for outgoing edges* e *is defined as*



$$f_{out_j}(t) = \left(\prod_{i=1}^{n} \int^{t} f_{in_i}(t')\, dt' \right) \frac{d}{dt} \quad , 1 \leq j \leq m$$

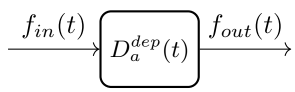
Having a generic analytical result about the edge transition behavior of the process is difficult unless the input is in form of time-shifted dirac delta functions or PDFs of an exponential distribution.

Definition 4 (f for XOR-Gateway). *Given $g \in G_{XOR} \subseteq G$ with incoming edges $\{in_1, \ldots, in_n\} = E \cap (N \times g)$, outgoing edges $\{out_1, \ldots, out_m\} = E \cap (g \times N)$, and edge execution probabilities $\{p_{out_1}, \ldots, p_{out_m}\}$, the edge transition probability function $f_e(t)$ for outgoing edges e is defined as*



$$f_{out_j}(t) = p_{out_j} \sum_{i=1}^{n} f_{in_i}(t) \quad , 1 \leq j \leq m$$

Definition 5 (f for Activities). *Given an activity $a \in A$ with one incoming edge e_{in}, one outgoing edge e_{out}, and activity duration PDF D_a^{dep}, the edge transition probability function $f_{out}(t)$ is defined as*



$$f_{out}(t) = f_{in}(t) * D_a^{dep}(t)$$

Note that the convolution operator $*$ for two functions f and g is defined as $(f * g)(t) = \int f(t')g(t - t') \, \mathrm{d}t'$.

We compute edge transition probability functions directly on the process. Another way of doing this is described in [21]. Their method requires decomposition of the process into process blocks.

4.2 Estimating Total Resource Utilization

In order to be able to define the total resource utilization function, we need to know the probability density of an activity a being executed at time t. From edge transition probabilities for the incoming edge of a, the probability density that an activity is executed for each point in time is represented by the activity execution probability function $F_a(t)$.

Definition 6 (Activity execution probability function). *The function $F_a(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denotes the probability density that an activity $a \in A$ is executed at time $t \in \mathbb{R}_{\geq 0}$.*

$$\begin{aligned} F_a(t) &= P(\textit{Activity is currently being executed at } t) \\ &= P(\textit{Activity is entered before } t) - P(\textit{Activity is left before } t) \\ &= \int_0^t f_{in}(t') \, dt' - \int_0^t f_{in}(t') * D_a^{dep}(t') \, dt' = \int_0^t f_{in}(t') \, dt' - \int_0^t f_{out}(t') \, dt' \end{aligned}$$

Example 3. Our running example with the activity duration PDFs D_a in Fig. 1 would then result into the activity execution probability functions F_a presented in Fig. 4. *Manage* is immediately executed once. As *Develop* and *Test* repeat

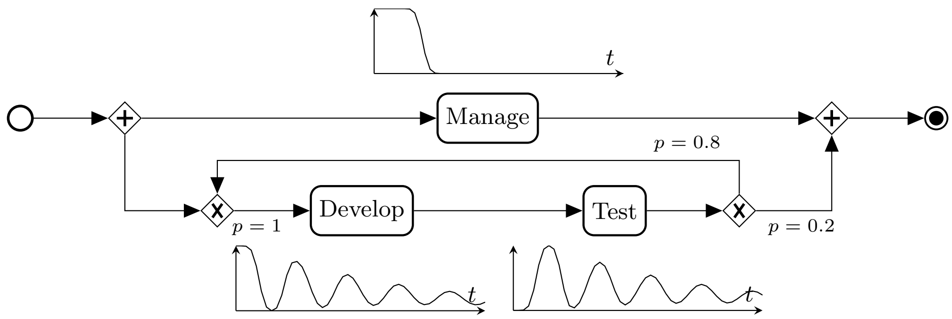


Fig. 4: The running example process with activity execution prob. functions F_a

(infinitely) in the loop, each repetition lowers the execution probability of the next iteration.

Output 1 (Total utilization PDF over time)**.** O_r is the utilization PDF of a resource $r \in R$ at time $t \in \mathbb{R}_{\geq 0}$, where $A = \{a_1, \ldots, a_n\}$. A value $O_r(t, u)$ represents the probability density that r is utilized by an amount of u percent of his/her time (e.g. $u = 1$ means full time, $u > 1$ means over-utilization) at time t. The operator $*_u$ is the convolution over the parameter u.

$$
\begin{aligned}
O_r &: (\mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}) \rightarrow \mathbb{R}_{\geq 0} \\
O_r(t, u) &= \delta(u) \\
&\quad *_u \ (U_{a_1,r}(u) F_{a_1}(t) + \delta(u)(1 - F_{a_1}(t))) \\
&\quad \ldots \\
&\quad *_u \ (U_{a_n,r}(u) F_{a_n}(t) + \delta(u)(1 - F_{a_n}(t)))
\end{aligned}
$$

For each resource, there is one total utilization PDF over time. In these PDFs, the activity execution probability functions and resource utilization PDFs of the respective resources are combined.

Example 4. In Fig. 5, the total utilization PDF over time for Jack O_{Jack} is based on (1) our running example with the activity execution probability functions F as seen in Fig. 4, and (ii) the resource utilization PDFs U as seen in Fig. 2. $O_{Jack,Test}(t, u) = (W_{Jack,Test}(u) F_{Test}(t) + \delta(u)(1 - F_{Test}(t)))$ is the probable utilization of Jack by the activity *Test* shown in top-left corner. In a similar way, $O_{Jack,Manage}$ is on the top-right corner. We do not show $O_{Jack,Develop}$, since it has no influence on Jack's total utilization (see Fig. 2). We can clearly observe in $O_{Jack,Manage}$ that the activity *Manage* is non-repeating. The combined total utilization PDF for Jack O_{Jack} is shown in bottom of the Figure. It shows how the probable utilizations of Jack's activities combine into a period of over-utilization as shown in the red region. Such region representing over-utilization are of special interest for the decision makers (e.g., project managers) who are also responsible for time and resource management.

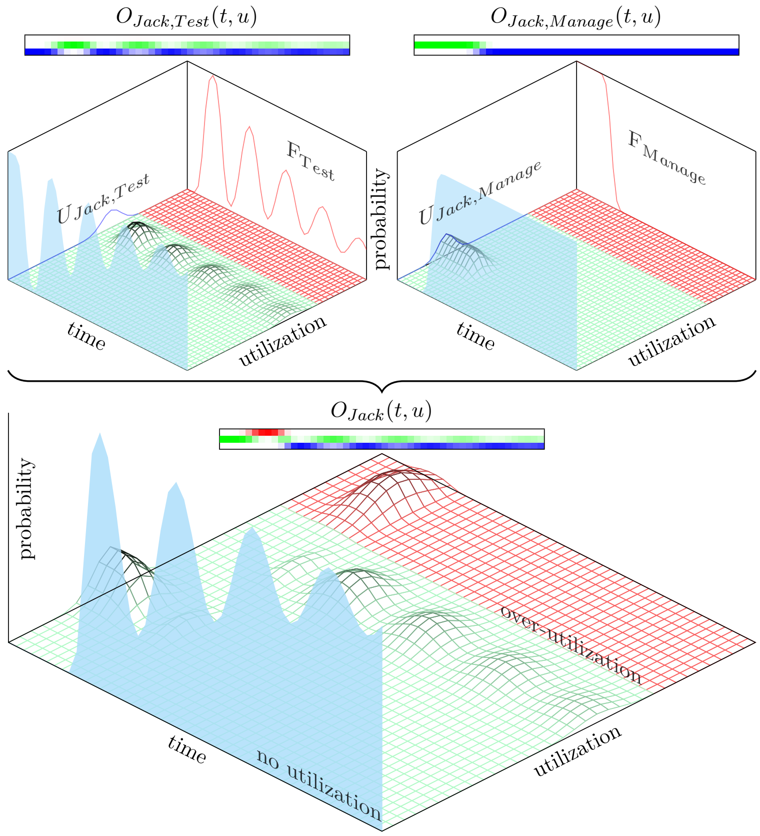


Fig. 5: Total Utilization PDF over time $O_{\texttt{Jack}}$ of our running example and the relevant pieces it is composed of. Directly above the plots, we provide a more condensed visualization where the x-axis is time, and the y-axis is utilization. The color intensity is directly proportional with the probability density value. Note that the blue parts are of infinite slope caused by $\delta(u)$.

4.3 Quality Metrics

Based on resource utilization PDFs O_r, we can define various metrics for quantifying abnormal resource utilization. These metrics can be used as optimization criteria for managing organizations and their schedules.

Output 2 (Resource over-utilization metric)**.** The resource over-utilization metric $m^+{}_r : \mathbb{R}_{\geq 0}$ for a resource $r \in R$ is the volume of utilization larger than 1.

$$m^+{}_r = \int\limits_0^\infty \int\limits_1^\infty oO_r(t,u)\,\mathrm{d}u\,\mathrm{d}t$$

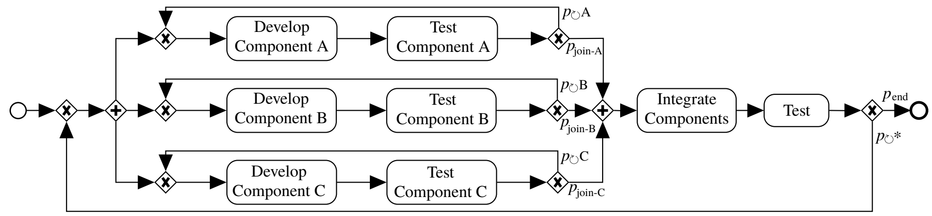


Fig. 6: An often-encountered development process with multiple parallel "Develop and Test" cycles with a final integration step

The total over-utilization m of a process is the sum of over-utilization of all resources: $m^+ = \sum_{r \in R} m^+{}_r$

In other words, m^+ is a value characterizing resource over-utilization for a whole organization (i.e., for all resources and the processes involved).

Output 3 (Resource under-utilization metric)**.** The resource under-utilization metric $m^-{}_r$ for a resource $r \in R$ is the accumulated volume under the optimal resource occupancy threshold on the occupancy axis (cf. 1.0).

$$F_P(t) = \int_0^t f_{e_{start}}(t')\,\mathrm{d}t' - \int_0^t f_{e_{end}}(t')\,\mathrm{d}t'$$

$$m^-{}_r = \int_0^\infty F_P(t) \int_0^1 |1-u| O_r(t,u)\,\mathrm{d}u\,\mathrm{d}t$$

In other words, m^- is a value characterizing resource under-utilization for a whole organization (i.e., for all resources and the processes involved). F_P in the formula above is similar to F_a in Definition 6 in the sense that F_P denotes the probability that a process P is executed at time $t \in \mathbb{R}_{\geq 0}$.

5 Application to a Real Process

We tested our method in a setting where the utilization of 10 resources is forecasted in 10 process instances of the process shown in Fig. 6. In this realistic process, each system consists of different types and number of components that are developed and tested in parallel. All *Develop* activities must be tested independently, and repeated development effort is expected. Additionally, a final *Integrate* and *Test* phase is mandatory for the combined system, which may cause additional repetitions of the entire process. It is expected for resources to work on multiple processes in parallel.

Table 1 describes the properties of 10 process instances with different starting times and resource utilizations: The process name (**id**), the starting time (t_s), the initial letter of the resource allocated to a certain activity (r), mean duration of the activity (∂), repetition probability of the iteration "*Develop* and *Test*" for component X ($p_{\circlearrowleft X}$) are given (cf. $p_{\circlearrowleft *}$ is the edge execution probability of

Table 1: Properties of process instances

		Dev_A		$Test_A$			Dev_B		$Test_B$			Dev_C		$Test_C$			Int_*		$Test_*$		
id	t_s	r	∂	r	∂	$p_{\circlearrowleft A}$	r	∂	r	∂	$p_{\circlearrowleft B}$	r	∂	r	∂	$p_{\circlearrowleft C}$	r	∂	r	∂	$p_{\circlearrowleft *}$
1	40	B	50	G	20	0.6	E	12.5	H	5	0.3	I	25	J	10	0.3	J	22.5	C	9	0.1
2	40	H	40	G	16	0.3	I	10	H	4	0.6	E	20	H	8	0.1	F	18	I	7.2	0.15
3	40	F	70	E	28	0.2	C	17.5	J	7	0.6	C	35	F	14	0.2	B	31.5	I	12.6	0.1
4	40	J	60	I	24	0.2	G	15	H	6	0.5	E	30	F	12	0.3	B	27	I	10.8	0.1
5	40	B	40	C	16	0.6	B	10	B	4	0.3	I	20	B	8	0.3	B	18	I	7.2	0.1
6	70	G	20	A	8	0.4	E	5	F	2	0.4	E	10	F	4	0.4	B	9	G	3.6	0.2
7	100	B	80	E	32	0.4	E	20	H	8	0.8	C	40	B	16	0.1	D	36	A	14.4	0.1
8	110	H	40	I	16	0.5	I	10	J	4	0.8	A	20	D	8	0.5	D	18	G	7.2	0.1
9	120	H	60	C	24	0.6	G	15	J	6	0.3	A	30	H	12	0.5	J	27	C	10.8	0.1
10	170	B	30	G	12	0.7	E	7.5	F	3	0.6	E	15	D	6	0.5	D	13.5	E	5.4	0.3

restarting the process). The different components of each process have different failure rates and duration distributions.

In order to generate the utilization functions for each resource, we simulate the process instances with respect to their properties (cf. Table 1) and apply the methods described previously. Note that we leave out the process execution paths with a probability lower than 0.1%.

Fig. 7 shows the visualization of the utilization functions for each resource. For instance, Boris is overoccupied most of the times because he is allocated to 5 Develop, 3 Test, and 4 Integration activities which are overlapping. Grace becomes overoccupied due to the start of project-6 at day 70. Edgar, Florence, and Henry are also expected to be visibly overoccupied in some periods during the execution of these 10 projects, though their situations are not as critical as Boris's, because their activities are not as many times overlapping as the activities of Boris. David's utilization looks exceptionally low. Moreover, by using the quality metrics defined in Section 4.3, we are informed that there is a 10 times greater risk of resource under-occupancy ($m^- = 2665.45$) than resource over-occupancy ($m^+ = 273.44$). This can be intuitively confirmed by comparing the amount of blue and red areas on Fig. 7.

Fig. 7 and the quality metrics suggest that: (i) There are more resources than needed in this setting, and (ii) demand for resources can still be balanced, especially those of Boris's for a more robust execution.

The computational complexity of our method is equivalent to the complexity of numerical integration which is P-complete [22]. Its implementations perform in quasilinear time. Therefore, our method responds to the problems with real-world sizes in a few seconds, and it is suitable for both design-time and run-time prediction tasks.

6 Conclusions and Future Work

In this paper we have introduced a novel method for predicting resource utilization in decision-intensive business processes. Our approach facilitates to obtain a resource utilization overview for the whole organization where the processes have a stochastic nature. As a result, the decision makers are provided with actual

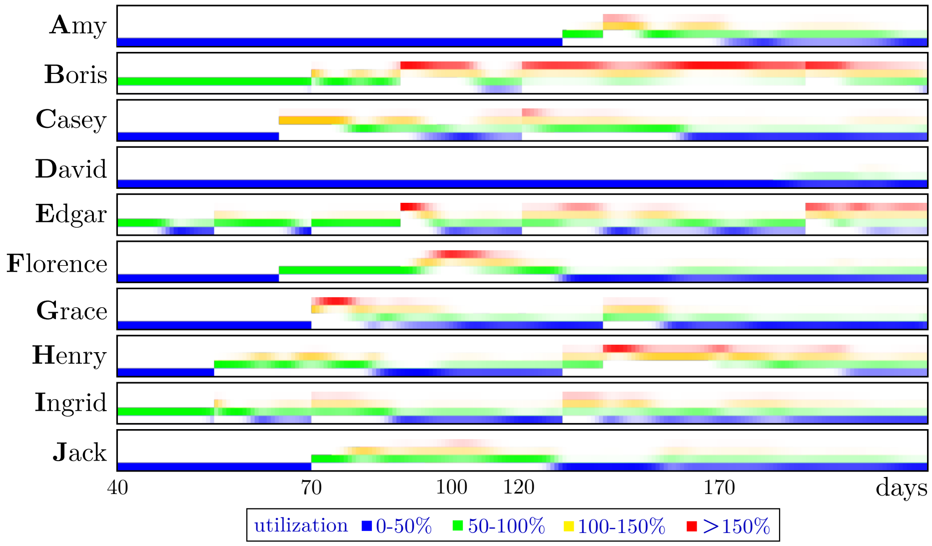


Fig. 7: Compact visualization of resource utilization forecast with 10 resources and 10 projects of type Figure 6 over one year

insights about their organizational feasibility. One advantage of our approach is that it can be readily used in practice, and it can be incorporated in BPM systems as a supplementary risk monitoring element.

Our future work primarily involves conducting exhaustive evaluations to assess the applicability of the approach in more real settings and compare the performance results. We also aim at integrating the approach into a BPM system as well as at adapting the current design-time method to be used at run time too, i.e. to make it more dynamic such that the resource utilization predictions are updated during the execution of the process instances.

References

1. Michael Rosemann and Jan vom Brocke. The six core elements of business process management. In *Handbook on business process management 1*, pages 105–122. Springer, 2015.
2. Giray Havur, Cristina Cabanillas, Jan Mendling, and Axel Polleres. Resource Allocation with Dependencies in Business Process Management Systems. In *BPM (Forum)*, volume 260, pages 3–19, 2016.
3. Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo A Reijers, et al. *Fundamentals of Business Process Management*, volume 1. Springer, 2013.
4. Qiang Liu and Jin He. Study of complex loop patterns based on time petri net. In *Computer Supported Cooperative Work in Design, 2007. CSCWD 2007. 11th International Conference on*, pages 801–805. IEEE, 2007.
5. Zhengxing Huang, Xudong Lu, and Huilong Duan. Resource behavior measure and application in business process management. *Expert Systems with Applications*, 39(7):6458–6468, 2012.

6. Anastasiia Pika, Wil MP van der Aalst, Colin J Fidge, Arthur HM ter Hofstede, and Moe T Wynn. Predicting deadline transgressions using event logs. *Lecture Notes in Business Information Processing*, 132:211–216, 2012.
7. Anastasiia Pika, Michael Leyer, Moe T Wynn, Colin J Fidge, Arthur HM Ter Hofstede, and Wil MP Van Der Aalst. Mining resource profiles from event logs. *ACM Transactions on Management Information Systems (TMIS)*, 8(1):1, 2017.
8. Raffaele Conforti, Massimiliano de Leoni, Marcello La Rosa, Wil MP van der Aalst, and Arthur HM ter Hofstede. A recommendation system for predicting risks across multiple business process instances. *Decision Support Systems*, 69:1–19, 2015.
9. Paolo Ceravolo, Antonia Azzini, Ernesto Damiani, Mariangela Lazoi, Manuela Marra, and Angelo Corallo. Translating process mining results into intelligible business information. In *Proceedings of the The 11th International Knowledge Management in Organizations Conference on The changing face of Knowledge Management Impacting Society*, page 14. ACM, 2016.
10. Michael Rosemann and Michael Zur Muehlen. Integrating risks in business process models. *ACIS 2005 Proceedings*, page 50, 2005.
11. Michael Zur Muehlen and Danny Ting-Yi Ho. Risk management in the bpm lifecycle. In *International Conference on Business Process Management*, pages 454–466. Springer, 2005.
12. Wil MP van der Aalst. *Process mining: data science in action*. Springer, 2016.
13. Wil MP Van der Aalst, M Helen Schonenberg, and Minseok Song. Time prediction based on process mining. *Information systems*, 36(2):450–475, 2011.
14. Andreas Rogge-Solti and Mathias Weske. Prediction of business process durations using non-markovian stochastic petri nets. *Information Systems*, 54:1–14, 2015.
15. Marijke Swennen, Niels Martin, Gert Janssenswillen, Mieke J Jans, Benoît Depaire, An Caris, and Koen Vanhoof. Capturing resource behaviour from event logs. In *SIMPDA*, 2016.
16. Arik Senderovich, Matthias Weidlich, Avigdor Gal, and Avishai Mandelbaum. Mining resource scheduling protocols. In *International Conference on Business Process Management*, pages 200–216. Springer, 2014.
17. Michael Arias, Eric Rojas, Jorge Munoz-Gama, and Marcos Sepúlveda. A framework for recommending resource allocation based on process mining. In *International Conference on Business Process Management*, pages 458–470. Springer, 2015.
18. Suriadi Suriadi, Moe T Wynn, Jingxin Xu, Wil MP van der Aalst, and Arthur HM ter Hofstede. Discovering work prioritisation patterns from event logs. *Decision Support Systems*, 2017.
19. Francesco Folino, Massimo Guarascio, and Luigi Pontieri. Discovering context-aware models for predicting business process performances. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 287–304. Springer, 2012.
20. Raffaele Conforti, Sven Fink, Jonas Manderscheid, and Maximilian Röglinger. Prism–a predictive risk monitoring approach for business processes. In *International Conference on Business Process Management*, pages 383–400. Springer, 2016.
21. Wil MP Van der Aalst, Kees M Van Hee, and Hajo A Reijers. Analysis of discrete-time stochastic petri nets. *Statistica Neerlandica*, 54(2):237–255, 2000.
22. Akitoshi Kawamura. Computational complexity in analysis and geometry. *University of Toronto, Toronto, Ont., Canada*, 2011.