Workflow Performance Analysis and Simulation based on Multidimensional Workflow Net

Abstract

Workflow model performance analysis plays an important role in the research of workflow techniques and efficient implementation of workflow management. Instances dwelling times (IDT) which consist of waiting times and handle times in a workflow model is a key performance analysis goal. In a workflow model the instances which act as customers and the resources which act as servers form a queuing network. Multidimensional workflow net (MWF-net) includes multiple timing workflow nets (TWF-nets) and the organization and resource information. This paper uses queuing theory and MWF-net to discuss mean value and probability distribution density function (PDDF) of IDT. It is assumed that the instances arrive with exponentially distributed inter-arrival times and the resources handle instances within exponentially distributed times or within constant times. First of all, the mean value and PDDF of IDT in each activity is calculated. Then the mean value and PDDF of IDT in each control structure of a workflow model is computed. According to the above results a method is proposed for computing the mean value and PDDF of IDT in a workflow model. Finally an example is used to show that the proposed method can be effectively utilized in practice.
**Keywords:** MWF-net, Performance Analysis, Dwelling time, Probability density

1. Introduction

Workflow technology is becoming increasingly important for achieving a process oriented view of the organization and subsequently process automation. Workflow management systems (WfMS) prove to be an effective means realizing full or partial automation of a business process [1]. Confronted with globalization and ever increasing competition, Quality of Service (QoS) requirements on WfMS, like performance, soundness, and availability, are of crucial importance. Businesses must ensure that the systems they operate not only provide all relevant services, but also meet the performance expectations of their customers. To avoid the pitfalls of inadequate QoS, it is necessary to analyze the expected performance characteristics of WfMS and workflow models. The methods used to do this are part of the discipline called Performance Engineering [3].

A business process is a set of one or more linked procedures or activities that collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships [1]. Despite the abundance of workflow management systems developed for different types of workflow based on different paradigms [4-7], the lack of rigorous theoretic foundation and then effective model verification and analysis methods has blocked workflow
techniques’ research and application[15],[35],[36].

The rationality and correctness analysis should be carried out from four aspects that are relevant for workflow modeling and workflow execution: process control logic, timing constraint logic, resource dependency logic, and information dependency logic [15], [34]. The correctness analysis of process control logic aims to avoid the deadlocks or structural conflicts in the execution of a workflow model caused by the errors in its process control. Some verification and conflict detection methods have been discussed in [2],[5],[8],[10],[35],[41],[43],[44]. The objective of resource dependency logic verification is to prove correctness of the static or dynamic resource allocation rules and consistency with the process control logic. The information dependency logic cares about the internal consistency of a workflow-related data and the correctness of temporary relation among different workflow application data. The timing constraint verification and analysis deal with the temporal aspects of a workflow model such as deadlines[9],[11],[36], time scales[12],[13],[34],[37],[38],[39],[42] schedulability analysis[33], and boundedness verification [14] and time violation handling [16], [17]. Quality of Service in Flexible Workflows is discussed in [40]. A workflow net similarity measure method is introduced in [51].

The above analysis can ensure only the functionally working workflow (correctness) but not its operational efficiency. The performance level[15]–[24],[31],[53], on the
other hand, aims to evaluate the ability of the workflow to meet requirements concerning some key performance indicators such as, maximal parallelism, throughput, service levels, and sensitivity. The analysis of resource availability and utilization, and average turnaround time is performed at this level. Performance analysis of workflow is of great importance in both enterprise applications [54] and scientific computing [52]. Yet it has not got enough attention of researchers commensurate with its importance until now [29]. The performance analysis of a workflow model (business process) is different from that of WfMS architecture [25], [26].

The performance analysis can be conducted only after the rationality and correctness analysis has been carried out. So it is assumed that there are no temporal and logical errors in the considered workflow models at the performance analysis stage.

*PN (Perti Net)* are the only formal techniques able to be used for structural modeling and a wide range of qualitative and quantitative analysis [29]. PN-based workflow management systems are widely used because of formal semantics, local state-based system description, and abundant analysis techniques [27]. So PNs are a naturally selected mathematical foundation for the formal performance analysis of workflow models. Many researchers use PN techniques to study workflow [4-5], [7]–[10], [14], [18]–[22] since Zisman used PN to model workflow processes [28].

A *PN* is a graphical and mathematical modeling tool. It consists of places, transitions, and arcs that connect them. Input arcs connect places with transitions, while output
arcs start at a transition and end at a place. There are other types of arcs, e.g. inhibitor arcs. Places can contain tokens; the current state of the modeled system (the marking) is given by the number (and type if the tokens are distinguishable) of tokens in each place. Transitions are active components. They model activities which can occur (the transition fires), thus changing the state of the system (the marking of the Petri net). Transitions are only allowed to fire if they are enabled, which means that all the preconditions for the activity must be fulfilled (there are enough tokens available in the input places). When the transition fires, it removes tokens from its input places and adds some at all of its output places. We usually use a bar to represent a transition, a circle to represent a place, and a dot to represent a token.

*PNs* which model workflow process definition are called *WF-nets (Workflow nets)* [4], [32]. A *PN* is called a *WF-net* if and only if:

1) PN has two special places: a source place and a sink place. The source place has no input transitions while the sink place has no output transitions; and

2) If we add a new transition to *PN* which connects source place with the sink place, then the resulting *PN* is strongly connected.

A *WF-net* presents only process control specification of a workflow model. In order to perform its time dimension verification and analysis, its specification should be extended to express its temporal behavior. Various works [12, 14, 46-50] introduce time into *PN*-based workflow models. Based on the semantics of *Time Petri Net (TPN)*,
Time Workflow net (TWF-net) [12, 46, 47] is proposed by regarding a timing constraint as a delay pair consisting of its lower and upper bounds. The definitions and notations of TWF-net coming from [12, 46, 47, 50] is briefly introduced here.

TWF-net is a three tuple \((WF\text{-}net, FI, M)\), where \(WF\text{-}net\) is a Workflow net. \(WF\text{-}net\) is also a three tuple \((P, T, F)\). \(P=\{p_1, p_2, \ldots, p_m\}\) is a set of places representing the state of a instance or the condition of its output transitions; \(T=\{t_1, t_2, \ldots, t_n\}\) is a set of transitions representing activities of the workflow model; \(F\) is a set of directed arcs linking places and transitions, and employed to describe precedence relations among activities; \(FI\) is a set of nonnegative real number pairs \([l, u]\) related to each transition, which is used to represent the minimum firing time and the maximum firing time respectively; \(M\) is a vector of \(m\)-dimensional markings where \(M(p)\) denotes the number of tokens representing the number of instances in \(p\).

There are usually two types of transitions in TWF-net, i.e., activity transitions and routing transitions. The former ones represent the activity nodes in a workflow model. The latter ones determine the control structures among former ones, e.g., and-split, and-join, or-split and or-join. Routing transitions are associated with a time interval \([0, 0]\) because they fire once they are enabled. For simplicity the time interval tags of routing transitions are omitted. Assume transition \(t\) is associated with a time interval \([l, u]\), \((0 \leq l \leq u)\). And let \(s\) and \(\tau(t)\) denote the enabled time and the actual firing time of \(t\), respectively. We have \(s+l \leq \tau(t) \leq s+u\).
The definition of *MWF-net* (*Multidimensional Workflow net*) is proposed by [15]. *MWF-net* describes the relations between multiple workflow processes, and the resource and organization structure they share. It is a five tuple \((W, O, R, F_P, F_R)\) where

- \(W\) is a set of *TWF-nets*.
- \(O\) is a set of roles defined in the organization perspective while
- \(R\) is a set of resource pools defined in the resource perspective;
- \(F_P\) describes mapping relation between process perspective and organization perspective while \(F_R\) represents binary relation between organization perspective and resource perspective.

Methods are discussed to compute the workload that arrival instances generate for the various resource pools and the lower bound of average turnaround time of instances [15]. This paper adopts *MWF-nets* [15] as a base mechanism to represent a performance analysis oriented workflow model.

### 2. Related works

A high-level stochastic *PN* (*SPN*) is used to model the routing constructs of a workflow, and then a method to compute throughput time of the process is presented [20]. Based on four performance equivalent formulae, the performance of a workflow is approximately analyzed in [21]. These two techniques both aim at calculating instances’ execution time and ignoring waiting time. The probability density of execution time is not taken into account, and cannot be applied to a workflow process of which the resources have stochastic service time. All the control structures are
mapped into Generalized stochastic PN (GSPN) [19], [22], and then a method based on a CTMC (continuous time discrete state Markov chain) to obtain lower bounds of the execution performance is discussed. A so-called load equivalence aggregation model derived from GSPN has been developed in [18], and then some performance related measures of human resources in a workflow by obtained by simulating the model. By defining change time, a performance evaluation model for the dynamic workflow changes is brought forward in [23]. However, the technique can be used for only acyclic time WF-net in which the arrival intervals of instances are constant. A queuing network is used to model the workflow [24], [31]. A method is yielded to identify the critical path of a workflow model and determine the minimum number of servers for the critical activity [24]. Some approximate approaches are employed in [31] for workforce configuration, and then the corresponding network is analyzed. But these techniques are not immediately applicable since they both assume that dedicated servers exist for an activity’s execution. Existing modeling and analysis techniques used to resolve different aspects of workflow performance-related problems have mainly two shortcomings, which restrict their application in practice. One is that few characteristics of IDT are considered. The other is that no accurate approaches are used to calculate PDDF of IDT in the whole workflow model when the arrival intervals are stochastic and handle times are partly stochastic and partly constant. A method is given to get PDDF of IDT when handle times of all resources are stochastic in [16]. In
practice some handle times are constant. Although some literatures take IDT into account in their workflow models, only some simple disciplines, such as constant arrival interval [23], [24] mean execution time [20], [21] are supposed. They are not necessarily true in actual workflow systems.

This paper uses queuing theory and MWF-net to discuss mean value and PDDF of IDT. It is assumed that the instances arrive with exponentially distributed inter-arrival times and the resources handle instances within exponential distributed times or within constant times. There are still no formal papers that focus on this problem so far. This paper will discuss this question.

Section 3 introduces some relevant Queuing Theory and queuing model in the workflow model. Section 4 provides an algorithm computing the instances’ dwelling time probability density at a transaction (activity). Section 5 discusses the method to compute the instances’ dwelling time probability density at four control structures of workflow models. Section 6 discusses the method to compute the instances’ dwelling time probability density at a workflow model. Section 8 presents an example to evaluate a business process by performance analysis.

3. Queuing models in workflow models

It is assumed in this paper that the firing of a transition (execution of the corresponding activity) in MWF-net needs the support of a specific resource. The
situation that one transition is projected to several resources can be transformed to this mapping relation by redefining roles and organization structure [15].

In the framework of MWF-nets [15], a resource pool is a class of individual resources that have the same skills and capability and performs the same set of roles. \( C = [c_1, c_2, \ldots, c_q] \) which is called resource state of the MWF-net means that there are \( c_j \) individual resources in each resource pool \( r_j \). Each role can be allocated to support several transitions firings and is performed by an individual resource of one of the appointed resource pools. Each resource pool may be appointed to undertake many roles. In the enactment environment, the handling of an instance generated by the firing of a transition is projected (using role as the medium) to one of the individual resources that have the probability to be appointed as the performer.

A **Queuing network (QN)** is a system which contains an arbitrary, but finite, number \( m \) of queues. Customers, sometimes of different classes, travel through the network and are served at the nodes [45]. A workflow model is an activity network where activities are interconnected by four workflow control structures, i.e., sequence, concurrency (AND), alternative (OR) and iteration (LOOP) [7]. According to queuing theory, the workflow model is a queuing network where each activity is acting as a node. The handleings of instances in the resource pool of each activity form a queuing model in which the instances act as customers and the resources act as servers. Each instance must queue for the service if the resources are busy or at least one instance queues for
service before it. Its handle time is specified by the firing delay of the corresponding transition. Its dwelling time consists of its waiting time and handle time.

Many business processes have time constraints called deadlines in their corresponding workflows. Hence, the IDT in a workflow model should satisfy the deadline requirement. Sometimes their service time and arrival intervals in a workflow are stochastically distributed or constant values. In order to evaluate a business process a method to compute the PDDF of IDT in a workflow model is discussed in the following.

Let $f_d(t)$, $f_w(t)$ and $f_h(t)$ denote the PDDF of dwelling, waiting and handle time respectively. Thus PDDF of IDT at a workflow will be calculated by three steps:

- Calculate PDDF of IDT in the resource pool of an activity;
- Calculate PDDFs of IDT in the resource pool groups of four control structures;
- Calculate PDDF of IDT in the resource pool group of the whole workflow.

4. **PDDF of IDT in the resource pool of an activity**

   It is assumed that each queuing system has infinite capacity. The service time of each resource is exponentially distributed with an average service rate $\mu_j$ (instances per hour) or is a constant value $1/\mu_j$. The instances arrive with exponentially distributed inter-arrival times at an average rate of $\lambda_j$ (instances per hour). Therefore the workflow model can be modeled as an M/M/C-M/D/C mixed queuing network.
where each activity is an independent M/M/C or M/D/C queuing system.

It is supposed that there are $c_j$ resources in the resource pool of Activity $j$ and $m_j$ instances in the resource pool of Activity $j$. Each resource serves for one queue, and each instance will enter the shortest queue when it arrives. Let $l_j$ denote the instances’ number in the shortest queue, then $l_j = \text{Round}(m_j/c_j)$. When an instance arrives, if $m_j \geq c_j$, it has to queue. Otherwise, it is processed directly. Let $P_{m_j}$ denote the probability with which $m_j$ instances are in the resource pool of Activity $j$. Let $P_{\min(l_j)}$ denote the probability with which the length of the shortest queue is $l_j$.

According to queuing theory [30], we know that:

$$P_0 = \left[ \sum_{n=0}^{c_j} \left( \frac{\lambda_j}{\mu_j} \right)^n / n! + \frac{\left( \lambda_j \right)^{c_j+1}}{c_j! \left( \mu_j \right)^{c_j+1}} \left( c_j \mu_j - \lambda_j \right) \right]^{-1}$$  \hspace{1cm} (1)

$$P_{m_j} = \begin{cases} 
\frac{P_0/m_j}{\left( \frac{c_j! \mu_j}{\mu_j} \right)^{m_j} \left( \lambda_j / \mu_j \right)^{m_j}} & (m_j \leq c_j) \\
\frac{P_0/c_j}{\left( \frac{c_j \mu_j}{\lambda_j} \right)^{m_j} \left( \lambda_j / \mu_j \right)^{m_j}} & (m_j > c_j)
\end{cases}$$  \hspace{1cm} (2)

$$P_{\min(l_j)} = \sum_{m_j = c_j}^{c_j + c_j - 1} P_{m_j}$$  \hspace{1cm} (3)

### 4.1 When handle times are exponentially distributed

Let $fd_j(t | l_j)$ denote the PDDF of the last instance in the queue whose length is $l_j$. As is known, its dwelling time distribution is an *Erlang* distribution. Then according to queuing theory [30], we have

$$fh_j(t) = \mu_j e^{-\mu_j t}$$  \hspace{1cm} (4)

When $l_j \geq 1$, according to queuing theory [30], $fw(t)$ of Activity $j$ can be stated as
\[ f_{W_j}(t) = \sum_{l_j=0}^{\infty} P_{m_{l_j}} f_{R_j}(t | l_j) = \sum_{l_j=0}^{\infty} \left( \sum_{m_{l_j}=c_j}^{c_j-1} P_{m_{l_j}} \right) \frac{\mu_j (\mu_j)^{l_j-1} e^{-\mu_j t}}{(l_j-1)!}. \]

It also can be illustrated as

\[ f_{W_j}(t) = \sum_{l_j=0}^{\infty} \left( \frac{P_0 (\mu_j * c_j)^i - (\lambda_j)^i}{(c_j - 1)!^2 (\mu_j * c_j - \lambda_j)} \right) \left( \frac{\lambda_j}{\mu_j * c_j} \right)^{l_j} e^{-\mu_j t} \left( \frac{\mu_j}{l_j - 1} \right)^{l_j} \] (5).

Let \( A_j = \frac{P_0 (\mu_j * c_j)^i - (\lambda_j)^i}{(c_j - 1)!^2 (\mu_j * c_j - \lambda_j)} \) and \( B_j = \left( \frac{\lambda_j}{\mu_j * c_j} \right)^{c_j} \), then Eq. (5) can be illustrated as

\[ f_{W_j}(t) = \sum_{l_j=0}^{\infty} A_j * B_j^{l_j} * e^{-\mu_j t} \left( \frac{\mu_j}{l_j - 1} \right)^{l_j} = A_j B_j e^{(B_j-1)\mu_j t} \] (6).

According to queuing theory [30], when an instance arrives, there are a probability that it must queue \((l_j > 0)\) and a probability that it is processed directly \((l_j = 0)\). It can be expressed as

\[
\int_0^x f d_j(t) dt = \int_0^x f w_j(t) \int_0^x f h_j(w) dw dt + P_{\min(0)} \int_0^x f h_j(t) dt \\
= \int_0^x f w_j(t) \int_0^{c_j-1} \mu e^{-\mu_j x} dw dt + \sum_{m_j=0}^{c_j-1} P_{m_j} \int_0^{c_j} \mu e^{-\mu_j x} dt \] (7).

The differential equation about \( x \) of Eq. (7) is stated as

\[
\int_0^x f d_j(x) = \mu e^{-\mu_j x} \int_0^x f w_j(t) e^{\mu_j x} dt + \sum_{m_j=0}^{c_j-1} P_{m_j} * \mu e^{-\mu_j x} \\
= \mu e^{-\mu_j x} \int_0^x A_j B_j e^{B_j \mu_j x} dt + \sum_{m_j=0}^{c_j-1} P_{m_j} * \mu e^{-\mu_j x} \] (8).

Specially, if there is only one resource in the resource pool of Activity \( j \), we have
\( f_d(x) = \left( \mu_j - \lambda_j \right) e^{(\lambda_j - \mu_j)x} + \left( \mu_j \times \frac{\mu_j - \lambda_j}{\mu_j} - (\mu_j - \lambda_j) \right) \times e^{-\mu_j x} \)

\( = (\mu_j - \lambda_j) e^{(\lambda_j - \mu_j)x} \) (9).

Thus the instances’ \( f_d(t) \) in the resource pool of an activity is figured out.

### 4.2 When handle times are constant value

When an activity is automatically performed by computers the handle times can be viewed as constant value. Here it is supposed that there is only one resource in the resource pool of each activity. Let \( f_d(t | m_j) \) denote the PDDF of the last instance in the queue whose length is \( m_j \). As is known, its dwelling time distribution is an average distribution from \((m_j - 1)/\mu_j \) to \( m_j/\mu_j \) if \( m_j \geq 2 \), if \( m_j = 1 \), its dwelling time is \( 1/\mu_j \). Then we have:

\[
P_0 = \frac{\mu_j - \lambda_j}{\mu_j}
\]

(10)

\[
P_{m_j} = \frac{\mu_j - \lambda_j}{\mu_j} \left( \frac{\lambda_j}{\mu_j} \right)^{m_j}
\]

(11)

\[
f_d(t) = P_{m_j} \times \mu_j = \frac{\mu_j - \lambda_j}{\mu_j} \left( \frac{\lambda_j}{\mu_j} \right)^{m_j} \times \mu_j \left\{ t \subset \left( \frac{m_j - 1}{\mu_j}, \frac{m_j}{\mu_j} \right) \right\}
\]

(12)

Each basic control structure corresponds to a resource pool group. The resource pool
group consists of all the resource pools of the activities in the control structure. Let
\( \text{PDDF} \) denote the \( \text{PDDF} \) of \( IDT \) in the resource pool group of a control structure
interconnected by Activity 1, Activity 2, ... and Activity \( n \). It is assumed that the
instances’ dwelling time in the resource pool of each activity is independent.

5.1 \textit{PDDF of IDT in the resource pool group of concurrent control structure}

Fig. 1(a) illustrates the flowchart of a concurrent structure while Fig. 1(b) and Fig.
1(c) describe its corresponding \textit{MWF-net} model and \textit{QN} model, respectively. The
control structure consists of three activities: Activity 1, Activity 2 and Activity 3. \( t_1, t_2 \)
and \( t_3 \) in Fig. 1(b) correspond to Activity 1, Activity 2 and Activity 3 in Fig. 1(a).
The instances’ dwelling time in the resource pool group of a concurrent control structure is the longest dwelling time in its branches. An instance in a concurrent control structure with \( n \) branches can be divided into \( n \) independent sub-instances which may run in parallel. Suppose Activity \( i \) and Activity \( j \) in Fig. 1 are interconnected by a concurrent control structure. Thus we have
\[ \int_0^x t * f \dot{d}_0(t) \, dt = \int_0^x u * f \dot{d}_1(u) \int_0^v f \dot{d}_j(v) \, dv \, du + \int_0^x u * f \dot{d}_j(u) \int_0^v f \dot{d}_j(v) \, dv \, du \]  

(13).

The differential equation about \( x \) of Eq. (13) is stated as
\[ f \dot{d}_y(x) = f \dot{d}_1(x) \int_0^v f \dot{d}_j(v) \, dv + f \dot{d}_j(x) \int_0^v f \dot{d}_j(v) \, dv \]  

(14).

When a concurrent control structure is composed of activity 1, activity 2… and activity \( n \), Eq. (14) can also be illustrated as
\[ f \dot{d}_{12...n}(x) = \sum_{i=1}^n \left( \frac{f \dot{d}_1(x)}{\int_0^v f \dot{d}_1(t) \, dt} \prod_{k=1}^n \int_0^v f \dot{d}_k(t) \, dt \right) \]  

(15).

### 5.2 PDDF of IDT in the resource pool group of sequential control structure

The instances’ dwelling time in the resource pool group of a sequential control structure is the sum of instances’ dwelling time at all of its branches.
Fig. 2 Flowchart, MWF-net and QN of an example sequential control structure

Let \( u_1, u_2, \ldots \) and \( u_n \) be the instances’ dwelling time in Activity 1, Activity 2… and Activity \( n \) interconnected by a sequential control structure. Let \( t \) be the instances’ dwelling time in the structure. We know that \( t = u_1 + u_2 + \cdots + u_n \). Thus \( PDF \) of \( IDT \) in the resource pool group of the sequential control structure can be expressed as

\[
\int_0^t \int_{0 \leq u_1 + u_2 + \cdots + u_n \leq t} f_{\text{pdf}}(u_1 + u_2 + \cdots + u_n) \, du_1 \, du_2 \cdots du_n \, dt
\]

\[
= \int_0^t \int_{0}^{\infty} \cdots \int_{0}^{\infty} f_{\text{pdf}}(u_1, u_2, \ldots, u_n) \prod_{i=1}^{n} du_i \, du_2 \cdots du_n \, dt
\]

(16)
5.3 PDDF of IDT in the resource pool group of alternative control structure

Each instance in the resource pool group of an alternative control structure is exclusively served at branch $i$ with probability $P_i$.

![Flowchart](image)

![MWF-net](image)

![QN](image)

Fig. 3 Flowchart, MWF-net and QN of an example alternative control structure
When an alternative control structure consists of Activity 1, Activity 2… and Activity n, PDDF of IDT in the resource pool group of the alternative control structure is shown as

\[ f_{d_{i_2...n}}(t) = \sum_{i=1}^{n} P_i f_{d_i}(t) \]  

(17).

5.4 PDDF of IDT in the resource pool group of loop control structure

In a loop control structure the instances return to be served again in the resource pool of certain activity with probability \( q \) after they pass it. For example, activity \( i \) and Activity \( j \) in Fig. 4 are interconnected by a loop control structure.
Fig. 4 Flowchart, MWF-net and $QN$ of an example loop control structure

The instances return to be served again in the resource pool of Activity $i$ via the resource pool of Activity $j$ with probability $q$ after they pass it. Then the instances will be served for $n$ times by the resources with probability $P_n = q^{n-1}(1-q)$. $PN$ of the loop control structure shown in Fig. 5 (a) is equivalent to $PN$ of the alternative structure shown in Fig. 5 (b).
In fact the probability with which the instances are served for 3 or more times in the resource pool group of loop control structure is so small that it may be ignored. Then the loop control structure is regarded as an alternative structure with 3 branches. Let $f_d(t)$, $f_d(t)$ and $f_d(t)$ imply PDDF of IDT in the first three branches of equivalent alternative structure in Fig. 2(b). According to Eq. (14), PDDF of IDT in the resource pool group of a loop structure is stated as

$$f_d(t) = (1-q) \cdot f_d(t) + q(1-q) \cdot f_d(t) + q^2(1-q) \cdot f_d(t)$$  

(18).

Branch 2 is sequential control structure as well as branch 3 in Fig. 2(b). According to Eq. (16), PDDF of IDT in each branch can be obtained as

$$\int_0^t f_d(t)dt = \int_0^t \int_0^{t-u} \int_0^{t-u-v} f_d(w)f_d(v)f_d(u)dwdvdu$$  

(19).

And

$$\int_0^t f_d(t)dt = \int_0^t \int_0^{t-u_1} \cdots \int_0^{t-u_{n-1}} \prod_{k=1}^{(n+1)/2} f_d(u_{2k-1})\prod_{k=1}^{(n+1)/2} f_d(u_{2k})du_1du_2\cdots du_n$$  

(20).
6. PDDF of IDT in the resource pool group of a workflow model

The resource pool group of a workflow model consists of all the resource pools of the activities in the model. When the PDDF of IDT in the resource pool of each activity of a workflow model is calculated, PDDF of IDT in the resource pool group of each control structure of a workflow will be figured out according to Eq. (15), Eq. (16), Eq. (17) and Eq. (18). Then each control structure is considered as an activity with the same PDDF of IDT. After this step, the activities are reduced and the workflow model is simplified. Then PDDF of IDT in the resource pool group of each control structure of the simplified workflow model is computed. And the model is simplified again until there is one single activity in the model. PDDF of IDT in the resource pool group of the single activity is PDDF of IDT in the resource pool group of the workflow model. Thus PDDF of IDT in the resource pool group of a workflow model is acquired.

7. Computational Experiments and Example Analysis

We assume that independent arrival times of the instances before the resource pool of each activity. However, because the arrival times are determined by the structure of the workflow model, we should discuss whether the assumption is feasible or not. If the PDDF of IDT worked out by our method can well fit the reality, we consider our method to be feasible. We develop a computer program to test and verify our method. The program has a graphical interface where users can create workflows models. It
may run workflow models in simulated environments and output the results. The interface of the program with 3 example workflow models is shown in Fig. 6. First we create several abstract workflow models with the program. Each model consists of at least two basic control structures. Then we run these models and create the histograms for the results. And then we compute PDDF of IDT for these models and create the curves of these PDDF of IDTs with Mathematica®. At last we compare the curves with the histograms. If they can well fit each other accordingly, we say our method is feasible.

Fig. 6 The program interface with 3 example workflow models

See Fig. 7, we create 3 abstract workflow models to test our method. For simplicity, the resource pool of each activity has and only has one resource, and no resource pool
shares resource with another.

![Workflow models](image)

Fig. 7 workflow models for testing the method

The data of the three models are listed below.

For model 1, \( f_a(t) = 10e^{-10t}, \ f_{h_1}(t) = 15e^{-15t}, \ f_{h_2}(t) = 20e^{-20t}, \ f_{h_3}(t) = 30e^{-30t}, \ f_{h_4}(t) = 20e^{-20t} \).

For model 2, \( f_a(t) = 7.5e^{-7.5t}, \ f_{h_1}(t) = 12e^{-12t}, \ f_{h_2}(t) = 24e^{-24t}, \ f_{h_3}(t) = 30e^{-30t}, \ f_{h_4}(t) = 10e^{-10t}, \ P(\text{loop split} \to \text{Activity 4}) = 0.8, \ P(\text{loop split} \to \text{Activity 3}) = 0.2. \)

For model 3, \( f_a(t) = 8e^{-8t}, \ f_{h_1}(t) = 20e^{-20t}, \ f_{h_2}(t) = 12e^{-12t}, \ f_{h_3}(t) = 10e^{-10t}, \ f_{h_4}(t) = 15e^{-15t}, \ f_{h_5}(t) = 24e^{-24t}, \ P(\text{or split} \to \text{Activity 1}) = 0.2, \ P(\text{or split} \to \text{Activity 3}) = 0.3, \ P(\text{or split} \to \text{Activity 4}) = 0.5. \)

According to Eq. (8) and Eq. (15)-(20), it is easy to obtain:
\[ f_{d_{\text{model}_1}}(t) = -15e^{-30t} + \frac{100}{3}e^{-25t} - 20e^{-20t} - \frac{55}{3}e^{-10t} + 20e^{-5t} - 100e^{-10t} t \]  

(21),

\[ f_{d_{\text{model}_2}}(t) = -16.57e^{-29.76t} + 1.1713e^{-28.8t} + 16.74e^{-23.76t} - 0.7588e^{-22.8t} + 0.77419e^{-18t} - 8.1248e^{-4.5t} + 6.7685e^{-2.5t} + \left(-28.754e^{-29.76t} - 67.998e^{-23.76t} + 12.09e^{-22.8t}\right)t + 145.06t^2e^{-23.76t} \]  

(22),

\[ f_{d_{\text{model}_3}}(t) = \frac{16744}{525}e^{-18.4t} - \frac{7088}{105}e^{-16t} + \frac{1848}{105}e^{-14t} + \frac{7176}{525}e^{-10.4t} + \frac{456}{105}e^{-7.6t} \]  

(23).

We generate the curves of PDDF of IDT in model 1,2,3 by Mathematica\textregistered. Then we run our program to simulate these models and copy the results to Microsoft Excel\textregistered for creating the histograms. The curves and histograms of model 1,2,3 are illustrated in Fig. 8, Fig. 9 and Fig. 10, respectively. From these curves and histograms, it is obvious that the histogram fit the curve well for each model. So our method is feasible.
Fig. 8 The function curve and histogram of model 1
Fig. 9 The function curve and histogram of model 2
Now we describe how our method is utilized in practice. When a business process is modeled with a workflow model, performance analysis can be performed to evaluate its attributes, such as reliability and soundness [40]. When PDFF of IDT in the resource pool group of the workflow model is figured out, the reliability of the business process on any given condition will be achieved. Suppose that the flowchart and PN of a product design process can be stated in Fig. 11. The arrival and departure processes of its activities make up of an M/M/C queuing network.
Fig. 11 Flowchart and PN of a product design process

Let \( fa(t) \) denote PDDF of the instance arrival intervals. Parameters of the workflow model are shown as

\[
fa(t) = 6e^{-6t}
\]

\[
P(t_3 \rightarrow \text{firing}) = 0.5
\]

\[
P(t_4 \rightarrow \text{firing}) = 0.5
\]

\[
P(t_5 \rightarrow \text{firing}) = 0.1
\]
\((f_{h_1}(t), f_{h_2}(t), f_{h_3}(t), f_{h_4}(t), f_{h_5}(t)) = (22e^{-22t}, 10e^{-10t}, 9e^{-9t}, \frac{6}{7}e^{-\frac{6}{7}t}, \frac{13}{7}e^{-\frac{13}{7}t})\).

Let \(P(0 < t < T)\) and \(E(t)\) be the proportion of the instances that depart before the deadline \(T\) and the mean dwelling time of the instances. Then \(P(0 < t < T)\) and \(E(t)\) are stated as:

\[
P(0 < t < T) = \int_0^T f_d(t) dt
\]
(24)

\[
E(t) = \int_0^\infty t \cdot f_d(t) dt
\]
(25)

According to Eq. (8) and Eq. (15)-(20), we have

\[
f_{d_{\text{workflow}}}(t) = 6.54e^{-16t}(0.375(1-e^{-12t}) - (1-e^{-6t}) + 4.05(1-e^{-4t}) + 1.35(1-t*e^{-12t} - e^{-12t}))
+ (1-e^{-6t})(-2.42e^{-6t} + 1.84e^{-12t} + 3.32e^{-4t} + 6.64t*e^{-12t})
\]
(26)

Eq. (26) is PDDF of IDT in the workflow as shown in Fig. 11. The instances dwelling times distribution in the workflow is illustrated in Fig. 12 (generated by Mathematica®).

![Fig. 12 The curve of \(f_{d_{\text{workflow}}}(t)\)](image-url)
With $PDDF$ of $IDT$ in a workflow model, the performance of corresponding business process can be analyzed precisely. It is helpful to improve the business level and resulting core competitive ability for an enterprise.

8. Conclusion

This paper has presented a theoretical method to calculate $PDDF$ of $IDT$ in the resource pool group of a workflow model where the activities are structured and predictable. An example has shown its availability in practice. This paper for the first time considers all the necessary information for the performance related theoretical analysis of a workflow model. Firstly, an MWF-net is used to the model the workflow. Then, it is assumed that the handle time of each resource is exponentially distributed or a constant value and the instances arrive with exponentially distributed inter-arrival times. Since activities are the basic units of the workflow, $PDDF$ of $IDT$ in the resource pool of an activity is calculated firstly. Then $PDDF$ of $IDT$ in the resource pool group of each kind of control structure is discussed. At last a simulation proves that our method is right and effective.

During the discussion of workflow performance related analysis, it is assumed that service time of each resource is exponentially distributed or a constant value and the instances arrive with exponentially distributed inter-arrival times and the resources do not need to repair. The techniques proposed in this paper need to be extended to deal
with the case that service time and arrival interval are normally distributed and the resources have time to repair. This will be left for future exploration.

Acknowledgements

We would like to thank the editor-in-chief, the editors and the referees for their kind and careful work on this manuscript. This work was founded by Natural Science Foundation of China (Project No. ******* , ******* and ******* ) and the Fundamental Research Funds for Central Universities (FRF-TP-12-047A).

References


of workflow systems based on stochastic petri net models,” Lecture Notes Computer Science #2480, 2002, pp. 64–79.


2003


