

# Stochastic Model for Multiple Classes and Subclasses Simple Documents Processing

Pierre Moukeli Mbindzoukou<sup>1,2</sup>, Arsène Roland Moukougou<sup>3</sup>, Marius Massala<sup>4</sup>

<sup>1</sup>Laboratoire Africain d'Informatique et de Mathématique Appliquée (LAIMA), Institut Africain d'Informatique, Libreville, Gabon

<sup>2</sup>Département Informatique, Institut Supérieur de Technologie (IST), Libreville, Gabon

<sup>3</sup>Département Mathématique, Université des Sciences et Techniques de MASUKU, Franceville, Gabon

<sup>4</sup>Département de Géographie, Université Omar Bongo, Libreville, Gabon

Email: pierre.moukeli@gmail.com, arsenem@hotmail.com, mariusmassala@gmail.com

**How to cite this paper:** Mbindzoukou, P.M., Moukougou, A.R. and Massala, M. (2021) Stochastic Model for Multiple Classes and Subclasses Simple Documents Processing. *Intelligent Information Management*, 13, 124-140.

<https://doi.org/10.4236/iim.2021.132006>

**Received:** January 9, 2021

**Accepted:** March 6, 2021

**Published:** March 9, 2021

Copyright © 2021 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

The issue of document management has been raised for a long time, especially with the appearance of office automation in the 1980s, which led to dematerialization and Electronic Document Management (EDM). In the same period, workflow management has experienced significant development, but has become more focused on the industry. However, it seems to us that document workflows have not had the same interest for the scientific community. But nowadays, the emergence and supremacy of the Internet in electronic exchanges are leading to a massive dematerialization of documents; which requires a conceptual reconsideration of the organizational framework for the processing of said documents in both public and private administrations. This problem seems open to us and deserves the interest of the scientific community. Indeed, EDM has mainly focused on the storage (referencing) and circulation of documents (traceability). It paid little attention to the overall behavior of the system in processing documents. The purpose of our researches is to model document processing systems. In the previous works, we proposed a general model and its specialization in the case of small documents (any document processed by a single person at a time during its processing life cycle), which represent 70% of documents processed by administrations, according to our study. In this contribution, we extend the model for processing small documents to the case where they are managed in a system comprising document classes organized in subclasses; which is the case for most administrations. We have thus observed that this model is a Markovian  $M^{L \times K}/M^{L \times K}/1$  queues network. We have analyzed the constraints of this model and deduced certain characteristics and metrics. *In fine*, the ultimate objective of our work is to design a document workflow management system, integrating a component of global behavior prediction.

---

## Keywords

Document Processing, Workflow, Hierarchic Chart, Counting Processes, Stochastic Models, Waiting Lines, Markov Processes Priority Queues, Multiple Class and Subclass Queues

---

## 1. Introduction

Document processing is a central activity in public and private administrations. The dematerialization of documents requires the digital management of the workflows of said documents. However, many questions arise when documents are submitted to a system for processing, with the primary concern of optimizing said system; that is, to reduce the size of its queues and document processing times at each station. To this end, the system must be able to answer certain questions: how much time does a document spend in a queue? How long is a queue at any given time? What is the minimum speed required for a station not to exceed a maximum document threshold in its queue? If a station becomes saturated, how many new stations must be created to alleviate its burden?... So many questions to which we try to provide an answer through the model that we propose in this contribution for the processing of documents. Quite naturally, the answers provided should be transposable to similar problems.

Document management issue was raised for a long time, mainly with the appearance of office automation in the 1980s, which led to dematerialization and Electronic Document Management (EDM). Likewise, workflow management has experienced significant development [1]-[12], mainly focused on the industry. However, it seems to us that document workflows have not had the same interest for the scientific community. But nowadays, the emergence and supremacy of the Internet in electronic exchanges and e-Government [13] [14] [15] [16] [17] are leading to a massive dematerialization of documents; which requires a conceptual reconsideration of the organizational framework for the processing of said documents in both public and private administrations [17] [18]. This problem seems open to us and deserves the interest of the scientific community [19]. Indeed, EDM has mainly focused on the storage (referencing) and circulation of documents (traceability). It paid little attention to the overall behavior of the system in processing documents. This is the subject of the work that our team has been carrying out for some years.

In a first contribution, we proposed a general model relating to document processing systems [19]. This model needed to be adapted to specific cases of certain types of documents. This is why, in a second contribution [20], we specialized this general model in the particular case of small documents (any document processed by a single person at a time during its processing life cycle) which represent 70% of documents processed by administrations, according to our study. In this contribution, we extend the model of small documents

processing [20] to the case where they are managed in a system comprising document classes organized in subclasses; which is the case for most administrations. In our work, we noted that this new model corresponds to a Markovian  $M^{L \times K}/M^{L \times K}/1$  queues network, which we had to analyze the constraints and deduce certain characteristics and metrics. Finally, the ultimate aim of our work is to design a document workflow management system, integrating a component of global behavior prediction.

The second part of this article supplements the definitions proposed by MOUKELI *et al.* [19] [20]. The next part sets out in detail the problem briefly described in the introduction. The fourth part is devoted to the description of our model; starting with general considerations, we describe our mathematical model with its static and dynamic components. The next part is devoted to the study of the performances of the system. Finally, we conclude with the perspectives of this work.

## 2. Definitions

In previous works [19] [20], the main concepts related to document processing have been defined. These include, by way of reminder, the following concepts: managerial unit, document, document processing, processing time, processing station or unit, document transmission, document processing chart, organization chart, document tracking, simple document, mail, hierarchical organization chart, hierarchical level and hierarchical line. In addition, the following definitions are added.

**Definition 1:** *Class of documents.* Set of documents relating to the same subject, field or sector of activity. Thus, financial documents, legal documents or personnel documents form classes of documents. For the purposes of this article, at each processing station, class  $C$  documents (example: class of financial documents) are classified by a priority service discipline in subclasses  $C_1, C_2, \dots, C_p$ . Documents of subclass  $C_1$  have priority over those of subclasses  $C_2, \dots, C_p$  and so on. The documents of the same subclass  $C_1$  are stored in FIFO; so the service discipline within a subclass is FIFO.

**Definition 2:** *Multiple class and subclass queue.* This is a queue whose documents are arranged by classes and within a class, the documents are arranged by priority subclasses. The discipline of service is both inter-class and intra-class. The choice of a class by the processing station is made fairly. All the classes have the same probability of being chosen (equiprobability).

**Definition 3:** *Multiple class and subclass processing station.* This is, in a managerial unit, a person (or expert) equipped with all the amenities, able to process documents stored in a queue of multiple classes and subclasses. As the classes are equally likely, the station randomly chooses the class with which it must process the leading document.

**Definition 4:** *Network of processing stations with multiple classes and subclasses.* Network of stations, each node of which is a processing station with a

multiple class and subclass queue. In the context of this study, this graph is a tree corresponding to the hierarchical organization chart of the administration in charge of processing documents.

### 3. Problem of Processing Simple Documents in a Network with Multiple Class and Subclass Queues

The general issue of document processing was raised by MOUKELI and NEMBE [20]. That relating to simple documents in a hierarchical environment was posed by MOUKELI *et al.* [20]. In summary, according to those works, document processing occupies a central place in the daily life of administrations, at the level of all jobs. This generates a significant flow of information, the analysis and monitoring of which are essential for efficient processing. These documents, which may have priorities relative to each other, are processed at each node, in the form of queues [21]-[30]. With the dematerialization of documents, the modeling of these flows becomes a scientific requirement, with a view to giving each station an operational framework reflecting the advantages of an environment using paper documents, without its flaws. To facilitate the analysis of this data flow, MOUKELI and NEMBE [19] proposed a general mathematical model to represent it and predict its behavior. MOUKELI *et al.* [20] have adapted this general model to the particular case of simple documents, in particular due to the fact that these documents represent, according to their study, 70% of the documents processed in hierarchically organized administrations. Remember that a simple document is a document whose processing, at any time, is ensured by a single person throughout the processing chain [20].

The model studied by MOUKELI *et al.* [20] is based on the fact that each workstation that processes simple documents has a priority-discipline queue, in which all the documents awaiting processing are stored. In practice, if this model effectively adapts to the stations which are the leaves of the processing tree, the intermediate nodes and the root often store the documents in the queue by class (each class containing documents from the same sector activity). Each station randomly chooses the class for which it must process the leading document.

Let us take as an example a General Manager, on whose desk there are several batches of initials to be processed (**Figure 1** below). All the lots constitute its queue. His secretary ranks the initials of the same batch in order of importance, from highest and most urgent or important, to lowest and lowest priority. The secretary arranges the documents in each initial in order of arrival. The Managing Director randomly chooses a batch and processes the first document of the initialing at the head of this batch; then he removes this document from the signature and sends it back to his secretary; then he makes a new choice, and so on.

Suppose now that we are trying to dematerialize the office of the Director General with his initials; that is, replace it with a digital platform. How to model the batches of documents (batches of initials) that the decision-maker will have to continue to process as he did with paper documents, taking advantage of the



**Figure 1.** Picture of batches of initials to be processed.

advantages of digital and paper environments, but without the disadvantages of paper? How to model the sets of initials and their contents? One of the correct answers is to use multi-class and multi-subclass document queues. To this end, the present work is a precursor. The remainder of this document describes our strategy.

#### 4. Modeling of a Simple Document Processing System with Multiple Classes and Subclasses

Regarding our problem, a group of documents linked to the same sector, but to be treated in order of priority (example: a batch of initials of financial documents), will be represented by a *class*. In this class, documents of the same priority (for example the contents of an initials) belong to the same *subclass*. In each subclass (like each initials), the documents are listed in order of arrival. The station randomly chooses the class for which it must process the leading document. For us, in this contribution, it is a question of extending the model of MOUKELI *et al.* [20] in the context of a network of queues with multiple classes and subclasses, and of examining certain metrics and performance measurements of such a system.

In the previous part, we exposed the problem relating to the processing of simple documents in a hierarchical system of servers each equipped with a queue with multiple classes and subclasses. In this part, we detail the model that we have just briefly sketched. Beginning with a presentation of general considerations on the model, we then describe our mathematical model with its static and dynamic components.

##### 4.1. General Considerations and Specific Problems of Multi-Class and Sub-Class Queues

As stated by MOUKELI and NEMBE [20], the graph is the most appropriate mathematical tool to model the flow of documents exchanged between processing stations. In the case of a hierarchical organization, this graph is reduced to a tree [20]:  $T = (V, E, R)$ ; where  $V$  is the set of vertices or workstations;  $E$  is the set of edges, that is to say pairs of vertices  $(a, b) \in V^2$  such that  $a$

is the hierarchical leader of  $b$ , which also means that  $a$  and  $b$  can exchange documents for processing; finally,  $R \in V$  is the root node of the tree. This tree results from the hierarchical organization of the administration, in which, we assume that the processing stations are interconnected by a computer network similar to this tree.

In the following, we will refer to this hierarchical organization or document processing platform as a “*system*”.

Starting from the specifics of the MOUKELI *et al.* [20] model, simple documents in the same queue are processed by a station according to their priority, and with equal priority, according to the order of arrival. Since the station can only process one document at a time, it chooses a class to process the document at the top. To make this choice, the station has several options:

- *Random choice*: the station randomly chooses the class for which it must process the head document. Knowledge of the law of probability governing this choice is necessary to describe the general behavior of the system; knowing that there is a non-zero probability that cases of famine will occur (abandoned classes);
- *Choice of the oldest document*: each document being labeled with the date of arrival at the station, the latter chooses the class whose leading document is the oldest. This choice amounts to an implicit merger of the classes; because the documents, with equal priority, are ultimately processed in FIFO. This choice leads the station to behave according to the model of MOUKELI *et al.* [20];
- *Cyclic choice*: the station chooses the class in turn. This model amounts to considering that all the documents at the head of the queue arrived at the same time. It also leads to an implicit fusion of classes; ultimately, the station will behave according to the model described by MOUKELI *et al.* [20].

With regard to the three previous hypotheses, we will, in the rest of this contribution, study the case of random choice, by making the hypothesis that all the classes are equiprobable.

In our previous studies [19] [20], we identified six problems resulting from the modeling of document processing by a static graph, namely:

- 1) How to determine the variables for measuring system performance?
- 2) The static graph does not account for the dynamic behavior of the system;
- 3) The inputs to the system depend on the outputs;
- 4) How does the general model behave when it is applied to the specific case of simple documents, particularly in terms of indicators and performance measures?
- 5) A non-priority document may remain in the queue indefinitely if higher priority documents systematically arrive in said queue;
- 6) A document can rotate indefinitely in the hierarchy line due to the constant back-and-forth caused by consecutive changes.

Solutions to these problems have been proposed [19] [20]. In addition, there

are specific problems with multiple class and subclass queues with random class choice:

1) The random choice of the class influences the overall behavior of the system, because it impacts the metrics and the performance of said system. This incidence in fact depends on the law of probability which governs the choice of the class made by the station.

2) How to prevent cases of famine of certain classes inherent in the random choice made by the station? This situation can possibly exacerbate problems 5 and 6.

3) After processing, a station can change the class of a document; for example, a financial document may turn out to have a legal character; after a first treatment, it will pass from the Finance class to the Legal class. This is a same node routing. This operation can cause the document to loop; that is to say never to leave the station. How to detect and prevent such a situation?

The answers to these questions are provided in the remainder of this article and represent our main contribution.

## **4.2. Mathematical Model of the Multiple Class and Subclass Queue System**

The mathematical model that we propose aims to describe the static and dynamic behavior of a system made up of workstations dedicated to the processing of simple documents, each provided with a queue with multiple classes and subclasses, each of the classes being priority service. In accordance with the model of MOUKELI *et al.* [20], from which ours derives, this system consists of two components:

- A static component modeled by a tree reflecting the hierarchical organization of the administration in charge of document processing. The nodes of this tree form the states of a Markov chain, and the edges represent the probabilities of transitions between these states. The properties of the static component are described through the states and the matrix of transition probabilities. These probabilities depend on the nature of the simple documents and the processing capacity of the stations.
- A dynamic component made up of a network of servers each with a multiple class and subclass queue. The formal system is a quadruplet  $(V, E, R, S)$ ; where  $V$  is the set of vertices of the tree;  $E$  is the set of edges;  $R \in V$  is the root of the tree; and  $S$  is the set of multiple class and subclass queues.

The following two sections formally describe these static and dynamic components.

### **4.2.1. Static Model of a Multi-Class and Sub-Class Queue Server System**

In accordance with the model of MOUKELI *et al.* [20], the static component of the system is represented by a network of workstations dedicated to the processing of simple documents. The system is modeled by a tree, which reflects the hie-



rarchical organization of the administration responsible for processing documents. In this section, we focus on the generalities of the model and on the analysis of constraints related to queues with multiple classes and subclasses.

### 1) *General of the Static Multiple Class and Subclass Queue Model*

Let  $T = (V, E, R)$  be a tree, such that  $V$  is the set of vertices or workstations;  $E$  is the set of edges, that is to say pairs of vertices  $(a, b) \in V^2$  such that  $a$  is the hierarchical leader of  $b$ ; finally,  $R \in V$  is the root node of the tree. Nodes are workstations and edges represent the hierarchical relationship between these stations. Each edge  $(a, b) \in V^2$  can be labeled with the cost of crossing the transition from node  $a$  to node  $b$  or vice versa; which gives us a vector of valuations.

On this tree, with each node  $x$  is associated a queue with  $L$  priority classes, denoted by  $S_x$ . Each class  $C_{xi}$  (where  $C_{xi} \in S_x$ ,  $1 \leq i \leq L$ ,  $x \in V$ ) of node  $x$ , contains all the simple documents of activity sector  $i$ , waiting to be processed by  $x$ . These documents are listed in order of priority; and with equal priority in order of arrival. We have  $K$  priority levels. In each class  $C_{xi}$ , the waiting time for a document corresponds to the processing time that will have preceded it in the  $S_x$  queue; knowing that the processing time of a document  $d$  is given by the following formula [20]:

$$T_p(d) = T_w(d) + T_s(d), \quad (1)$$

$T_p, T_w, T_s$  respectively denoting the processing, waiting and service times of  $d$ .

When a document  $d$  is received at station  $x$ , the latter assigns it a priority (subclass)  $j$  and a class  $i$  and inserts it at the end of subclass  $C_{xij}$  (where  $1 \leq i \leq L, 1 \leq j \leq K$ ).

### 2) *Constraints related to queues with multiple classes and subclasses with priority*

The dynamic behavior of our document processing model shows that system outputs (processed documents) can become system inputs (documents to be processed). This dynamic behavior imposes on our system, two types of constraints whose solutions are known [20], namely:

- A document waiting in a class can remain there indefinitely because higher priority documents of the same class are systematically received by the station;
- There is a non-zero probability that a document will loop through the system due to constant back and forth in the processing hierarchy.

In addition to these two constraints, there are two new constraints related to the use of queues with multiple classes and subclasses with priority and random choice of the class whose leading document will be processed; which influences the overall behavior of the system:

- The random choice of the class means that cases of starvation can appear in the system, because the model admits a non-zero probability that a class is ignored with its documents; that is, said class may never be chosen. The system must therefore make it possible to detect such cases of famine and re-



solve them.

- As indicated above (problem 9), by routing on a node, a document can change class and even priority in the new class. This transfer of documents from one class to another on a node can create an infinite loop, so that a document never leaves a node. The system must therefore make it possible to detect such cases and resolve them.

Knowing that each node  $x$  of the system has  $L$  priority classes, the probability that the node chooses a class for the processing of its head document is  $1/L$  in the case of a fair choice (assumption of the uniform law). By recording the successive random choices of the station, it is possible to detect the occurrence and resolve cases of famine among its classes:

- **Detection of starvation cases:** Let  $x \in V$  be a station in the system,  $S_x$  the queue associated with it,  $C_{xi} \in S_x$  the class  $i$  of the queue,  $\delta_x(t)$  the number of documents processed by the station  $x$  in the interval  $[0, t]$ , and  $\delta_{xi}(t)$  the number of documents of class  $i$  processes by  $x$  in the interval  $[0, t]$ .

In a uniform distribution, the probability that class  $i$  be chosen is:

$$\begin{cases} p(i) = \frac{\delta_{xi}(t)}{\delta_x(t)} \\ et \\ \lim_{t \rightarrow \infty} \frac{\delta_{xi}(t)}{\delta_x(t)} = \frac{1}{L} \end{cases} \quad (2)$$

But if for a given station  $x$ , there is a class  $C_{xi}$ , such that:  $\lim_{t \rightarrow \infty} \frac{\delta_{xi}(t)}{\delta_x(t)} = 0$ , then

there is a famine for  $C_{xi}$ , if this class is not empty.

- **Starvation cases resolution:** If a starvation case is detected, the station concerned must act so that its next choice, instead of being random, goes to the penalized class. This process can be repeated until fairness is restored. Each station must therefore keep accounts of its choices, and a threshold  $\varepsilon \left( 0 \leq \varepsilon < \frac{1}{L} \right)$  must be set beyond which the class can be considered to be in

a state of famine: formally if  $\frac{\delta_{xi}(t)}{\delta_x(t)} \leq \varepsilon$ , then choose class  $i$  at time  $t+1$ .

Finally, with regard to infinite loops in the same node, a first solution to guard against this is to prohibit the transfer of documents from one class to another on the same node. But even with this extreme solution, error is possible. The solution we recommend consists in preventing a document from going back through a class in which it had already been on the same node. The implementation of this solution is presented below.

#### 4.2.2. Dynamic Model of the Multiple Class and subclass Queue System

As for the static part, let  $T = (V, E, R)$  be a tree, such that  $V$  is the set of vertic-

es or workstations,  $E$  the set of edges and  $R \in V$  the root node of the tree. The observed phenomenon is modeled as follows, through its dynamic component, which is used to describe the temporal behavior (evolution) of the system, represented by a tree network of  $N$  nodes each associated with a queue with multiple classes and subclasses.

For this purpose, we consider, at each station  $x \in V$ , a queue  $S_x$  with  $L$  classes of documents noted  $C_{x1}, C_{x2}, \dots, C_{xL}$ . Each class  $C_{xi}$  is made up of  $K$  subclasses or priority levels of the system, denoted  $C_{xi1}, C_{xi2}, \dots, C_{xiK}$ . Each class  $C_{xi}$  (where  $1 \leq i \leq L$ ) has a priority service discipline: documents in subclass  $C_{xij}$  have priority over documents in subclass  $C_{xi(j+1)}$  ( $1 \leq j < K$ ); and each subclass  $C_{xij}$  has a FIFO discipline.

At each node  $x \in V$ , classes are equiprobable. The class whose head document will be processed is chosen randomly. As in the model of [20], at any node  $x \in V$ , any document  $d$  of priority  $j$  ( $1 \leq j < K$ ) of the class  $C_{xi}$ , which has spent a waiting time greater than  $T$  ( $T > 0$ ) in a subclass  $C_{xij}$  gains in priority and goes to the next higher priority subclass  $C_{xi(j-1)}$ ; formally:

$$\forall d \in C_{xij} / T_{w0j}(d) > T \Rightarrow d \in C_{xi(j-1)}; \quad (3)$$

where:  $1 < j \leq K$ ,  $1 < i \leq L$ ,  $x \in V$  and  $T_{w0j}(d)$  is the waiting time of document  $d$  in subclass  $j$  regardless of class.

The remainder of this subsection formally describes the multiple class and subclass queuing system model: the network, the service discipline, and some system variables.

### 1) Multiple class and subclass queue, priority and random service discipline

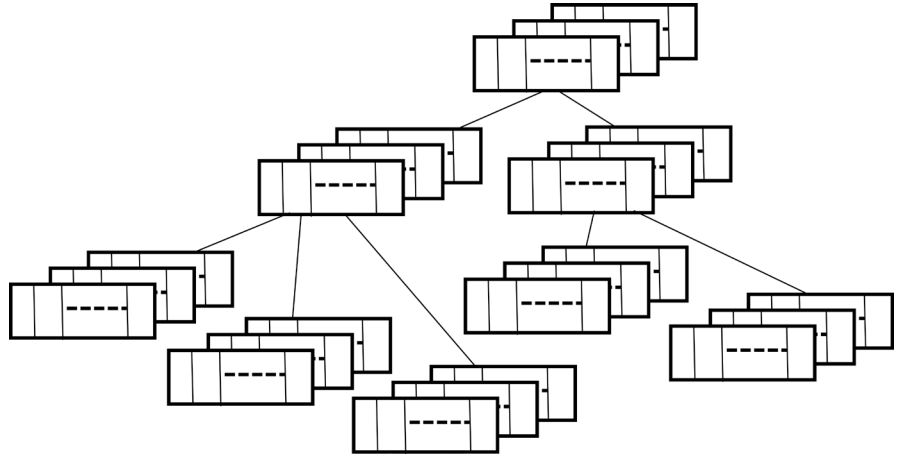
As in the model described by [20], at each node  $x \in V$ , documents of subclass  $C_{xij}$  ( $1 \leq i \leq L, 1 \leq j \leq K$ ) arrive in the queue following a Poisson process of parameter  $\lambda_{ij}$ . The length of service (or service time) of a document of subclass  $C_{xij}$  is a random variable which follows an exponential law of parameter  $\mu_{ij}$ . We will assume that the service times of documents in subclass  $C_{xij}$  are independent and identically distributed (iid). In addition, the document arrival processes in each node are independent from one another and from service times. Finally, each queue, assumed to have unlimited capacity, is associated with a single processing unit (server), also known as a service station.

### 2) Service discipline of a multiple classes and subclasses document queue

The queue  $S_x$  has  $L$  classes of documents, associated with a station  $x \in V$ , has a uniform service discipline: documents at the head of each class  $C_{xi}$  in this queue have the same probability of being chosen, that is  $1/L$ . However, within each class, we have priority service discipline. Thus, within class  $C_{xi}$ , documents of subclass  $C_{xij}$  have priority over those of subclass  $C_{xi(j+1)}$  ( $1 \leq i \leq L, 1 \leq j \leq K$ ). In each subclass  $C_{xij}$ , service discipline is FIFO. We will denote by  $M^{L \times K} / M^{L \times K} / 1$  such a queue, described formally below.

### 3) Description of queue $M^{L \times K} / M^{L \times K} / 1$

In our model, a queue  $M^{L \times K} / M^{L \times K} / 1$  is associated with each node (Figure 2). It



**Figure 2.** Example of tree network of 8 servers each having a 3-class queue.

is made up of  $L$  classes; and each class is itself made up of  $K$  document sub-classes. Each class can be thought of as an  $M^K/M^K/1$  type queue. In the system, each queue  $M^{L \times K}/M^{L \times K}/1$  is modeled as a birth and death process taking its values in  $\mathbb{N}^{L \times K}$ , where  $\mathbb{N}$  denotes the set of natural numbers.

Let us consider first each class  $C_{xi}$  of node  $x \in V$ . We denote by  $S_{xi}(t)$  the state of  $C_{xi}$ , the  $i^{\text{th}}$  class of node  $x$  at time  $t$ . The state space of the class is described by  $K$ -tuples  $(n_{i1}, \dots, n_{iK}) \in \mathbb{N}^K$ ; where  $n_{ij}$  is the number of documents of priority  $j$  in class  $i$ , at node  $x$ ,  $1 \leq i \leq L$  and  $1 \leq j \leq K$ . We denote by  $|S_{xi}(t)|$ , the length (number of documents) of class  $i$  at node  $x$  at time  $t$ .

Let us now consider  $S_x(t)$ , the state of the queue  $M^{L \times K}/M^{L \times K}/1$  associated with node  $x$ , at time  $t$ ; the state space of this queue is described by a matrix  $L \times K$ :

$$\begin{bmatrix} n_{11} & \cdots & n_{1K} \\ n_{i1} & \cdots & n_{iK} \\ n_{L1} & \cdots & n_{LK} \end{bmatrix} \in \mathbb{N}^{L \times K}; \quad (4)$$

where  $n_{ij}$  is the number of documents of priority  $j$  in class  $i$ ,  $1 \leq i \leq L$  et  $1 \leq j \leq K$ .

We denote by  $|S_x(t)|$ , the length of queue  $M^{L \times K}/M^{L \times K}/1$  associated with node  $x$  at time  $t$ .

#### 4) Description of $M^{L \times K}/M^{L \times K}/1$ queue network

Remember that our system is represented by a network of servers each associated with a queue  $M^{L \times K}/M^{L \times K}/1$ ; and the formal system is a quadruplet  $(V, E, R, S)$ ; where  $V$  is the set of vertices of the tree;  $E$  is the set of edges;  $R \in V$  is the root of the tree; and  $S$  is the set of multiple class and subclass queues.

The system is described with the following variables (knowing that in the formulas below, it is considered that  $x \in V$ ,  $1 \leq i \leq L$  and  $1 \leq j \leq K$ ):

- $N$  number of network nodes.
- $L$  number of classes in each queue.

- $K$  number of subclasses in each class.
- $n_{xij}$  number of documents of priority  $j$ , in class  $i$ , at node  $x \in V$ .
- The number of documents of class  $i$  at node  $x$ :

$$n_{xi} = \sum_{j=1}^K n_{xij}. \quad (5)$$

- The number of documents at node  $x$ :

$$n_x = \sum_{i=1}^L n_{xi} = \sum_{i=1}^L \sum_{j=1}^K n_{xij}. \quad (6)$$

$$n_x = \begin{bmatrix} n_{x11} & n_{x12} & \cdots & n_{x1K} \\ n_{x21} & n_{x22} & \cdots & n_{x2K} \\ \vdots & \vdots & \ddots & \vdots \\ n_{xL1} & n_{xL2} & \cdots & n_{xLK} \end{bmatrix}. \quad (7)$$

- The number of network documents:

$$n = \sum_{x \in V} n_x. \quad (8)$$

- The state of the  $i^{th}$  document class at node  $x$  ( $1 \leq i \leq L$ ):

$$S_{xi} = (n_{xi1} \cdots n_{xiK}), \quad (9)$$

- The state of queue  $M^{L \times K} / M^{L \times K} / 1$  at node  $x$ :

$$S_x = (S_{x1} \cdots S_{xL}) = \begin{bmatrix} n_{x11} & n_{x12} & \cdots & n_{x1K} \\ n_{x21} & n_{x22} & \cdots & n_{x2K} \\ \vdots & \vdots & \ddots & \vdots \\ n_{xL1} & n_{xL2} & \cdots & n_{xLK} \end{bmatrix}. \quad (10)$$

- The overall state of network:

$$S = (S_1 \cdots S_N) \\ = \begin{bmatrix} n_{111} & n_{112} & \cdots & n_{11K} \\ n_{121} & n_{122} & \cdots & n_{12K} \\ \vdots & \vdots & \ddots & \vdots \\ n_{1L1} & n_{1L2} & \cdots & n_{1LK} \end{bmatrix} \begin{bmatrix} n_{211} & n_{212} & \cdots & n_{21K} \\ n_{221} & n_{222} & \cdots & n_{22K} \\ \vdots & \vdots & \ddots & \vdots \\ n_{2L1} & n_{2L2} & \cdots & n_{2LK} \end{bmatrix} \cdots \begin{bmatrix} n_{N11} & n_{N12} & \cdots & n_{N1K} \\ n_{N21} & n_{N22} & \cdots & n_{N2K} \\ \vdots & \vdots & \ddots & \vdots \\ n_{NL1} & n_{NL2} & \cdots & n_{NLK} \end{bmatrix}. \quad (11)$$

where  $\mu_{xij}$  is the service rate for documents of priority  $j$  in class  $i$ , at node  $x$ .

- The average service rate for documents of class  $i$ , at node  $x$ :

$$\mu_{xi} = \frac{1}{K} \sum_{j=1}^K \mu_{xij}. \quad (12)$$

- The service rate at station  $x$ :

$$\mu_x = \frac{1}{L} \sum_{i=1}^L \mu_{xi}. \quad (13)$$

- $P_{xij}^{ykl}$  is the probability that a document of priority  $j$  of class  $i$  at node  $x$ , is transferred into a document of priority  $l$  in class  $k$  at node  $x$  (or routing probability); with  $(x, y) \in E$  (an edge of tree) or  $(x = y) \in E$  (change of class on the same node),  $1 \leq i \leq L, 1 \leq j \leq K, 1 \leq k \leq L$  et  $1 \leq l \leq K$ .
- $P_0^{xij}$  is the probability that a priority  $j$  document enters the network in class  $i$  of node  $x$ .

- $P_{xij}^0$  is the probability that a priority  $j$  document of class  $i$  leaves the network after having been served at node  $x$ .
- $\lambda_0^{xij}$  is the rate of arrival in the system, at node  $x$ , of documents of priority  $j$  in class  $i$ .
- The overall rate of arrival of documents from outside to the network:

$$\lambda = \sum_{x \in V} \lambda_0^{xij}. \quad (14)$$

- $\lambda_{xij}$  is the rate of arrival at node  $x$  of priority  $j$  documents in class  $i$ .
- The multi-class and subclass queue network, thus defined, is described by a Markov process  $X$  with values in  $\left(\left(\mathbb{N}^K\right)^L\right)^N = \mathbb{N}^{K \times L \times N}$ , where  $\mathbb{N}$  is the set of natural numbers.

## 5. Study of System Performance

In this part we focus on some performance indicators of a simple document processing system comprising a tree network of nodes each associated with a queue with multiple classes and subclasses. We are interested in performance indicators, system saturation and infinite loops.

### 5.1. Performance Indicators

#### 5.1.1. Measures Relating to the Static Component of the Model.

Below we have a summary of some indicators that can be calculated with our system, knowing that in each expression we have:  $T = (V, E, R)$ ,  $x \in V$ ,  $1 \leq i \leq L$  and  $1 \leq j \leq K$ .

- Size of class  $i$  of node  $x$  (number of documents in the class or queue  $C_{xi}$ ):

$$n_{xi} = |C_{xi}| = \sum_{j=1}^K n_{xij}; \quad (15)$$

where  $n_{xij}$  is the number of documents of priority  $j$  in the class  $i$  at node  $x$ .

- Station load (number of documents in the multiclass and subclass queue  $S_x$  associated with station  $x$ ):

$$n_x = |S_x| = \sum_{i=1}^L n_{xi} = \sum_{i=1}^L \sum_{j=1}^K n_{xij}. \quad (16)$$

- System load (number of documents in the system):

$$n = \sum_{x \in V} n_x. \quad (17)$$

- Average processing time at station  $x$ , of a document of class  $i$  and subclass  $j$ :

$$MeanTime(x_{ij}) = \frac{1}{n_{xij}} \sum_{k=1}^{n_{xij}} T_p(d_{xijk}); \quad (18)$$

where  $d_{xijk}$  denotes document number  $k$  of priority  $j$  of class  $i$  at node  $x$ .

- Average processing time of a document of class  $i$  at node  $x$ :

$$MeanTime(x_i) = \frac{1}{n_{xi}} \sum_{j=1}^K n_{xij} MeanTime(x_{ij}). \quad (19)$$

- Average processing time of a document by station  $x$ :

$$MeanTime(x) = \frac{1}{n_x} \sum_{i=1}^L n_{xi} MeanTime(x_i). \quad (20)$$

- Average processing time of a document by system  $T$ :

$$MeanTime(T) = \frac{1}{n} \sum_{x \in V} n_x MeanTime(x). \quad (21)$$

### 5.1.2. Measurements Relating to the Dynamic Component of the Model

Remember that our system is made up of a network of servers each equipped with a  $M^{L \times K}/M^{L \times K}/1$  queue. Knowing that each class  $C_{xi}$  of any node  $x$ , is comparable to the queue  $M^K/M^K/1$  with priority discipline, the measures relating to the application of Little's theorem defined in [21] are applicable to each class. A complementary study will make it possible to determine the dynamic behavior of the queue  $S_x$  associated with each node  $x$  and of the entire system.

## 5.2. System Saturation and Infinite Loops

In the [20] model, the implementation of incremental priority causes a queue to gradually turn into a FIFO; moreover, this model admits infinite loops. These same problems arise for our model, at the level of each class. The solutions proposed in [20] are applicable in our model, at the level of each class.

In addition, a document can loop on the same node due to local routing, through the change of class of a document. The solution we recommend to prevent such a loop is to prevent a document from going through the same class twice on the same node. For this purpose, we mark each document of the class in which it is passed.

- 1) Let  $d$  be a document being processed by station  $x$  at time  $t$ ,
- 2) Let  $S_x(t)$  be the multiple class and subclass queue associated with node  $x$  at time  $t$ , and  $C_{xi}(t)$  (with  $1 \leq i \leq L$ ) the state of class  $i$  at time  $t$ ,
- 3)  $M(d)$  the mark carried by document  $d$ .

Then we have the relation:

$$d \in C_{xi}(t) \Rightarrow d \in C_{xi'}(t+1) \Leftrightarrow i' \notin M(d); \quad (22)$$

With  $1 \leq i \leq L, 1 \leq i' \leq L$  and  $i \neq i'$ .

## 6. Conclusions and Perspectives

In this article, we have presented a mathematical model which is an extension of that of [20], in the case of queues with multiple classes and subclasses. This model is based on a real situation, represented by batches of documents placed on the work table of a decision maker, in progress or awaiting processing. The dematerialization of this working environment involves the transformation of such batches of documents into digital objects, which we have modeled in the form of queues. Other models are certainly possible.

Our model leads to the manipulation of markovian objects of type  $M^{L \times K}/M^{L \times K}/1$ . An in-depth study of the properties of these objects is necessary. Moreover, in

our model, each station randomly chooses the class with which it must process the leading document. We relied on a uniform law to govern this choice. We believe that it is necessary to carry out an empirical study to better understand the law of probability which governs this choice; artificial intelligence can be used for this purpose. These are all open issues.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] van der Aalst, W.M.P. and van Hee, K. (2002) Workflow Management Models, Methods and Systems. [http://ceit.aut.ac.ir/~sa\\_hashemi/My%20Teachings/MS-CEIT-Business%20Process%20Re-Engineering/Resources/MIT%20Press%20-%20Workflow%20Management-Models.%20Methods%20&%20Systems.pdf](http://ceit.aut.ac.ir/~sa_hashemi/My%20Teachings/MS-CEIT-Business%20Process%20Re-Engineering/Resources/MIT%20Press%20-%20Workflow%20Management-Models.%20Methods%20&%20Systems.pdf)
- [2] Gottschalk, F., van der Aalst, W.M.P., Jansen-Vullers, M.H. and La Rosa, M. (2008) Configurable Workflow Models. *International Journal of Cooperative Information Systems*, **17**, 177-221. <https://doi.org/10.1142/S0218843008001798>
- [3] Georgakopoulos, D., Hornick, M. and Sheth, A. (1995) An Overview of Workflow Management from Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, **3**, 119-153. <https://doi.org/10.1007/BF01277643>
- [4] Pesic, M., Schonenberg, M.H., Sidorova, N. and van der Aalst, W.M.P. (2007) Constraint-Based Workflow Models: Change Made Easy. Springer, Berlin. [https://doi.org/10.1007/978-3-540-76848-7\\_7](https://doi.org/10.1007/978-3-540-76848-7_7)
- [5] Shi, M., Yang, G., Xiang, Y. and Wu S. (1998) Workflow Management Systems: A Survey. *Proceedings of IEEE International Conference on Communication Technology*, Beijing, 22-24 October 1998, 6.
- [6] White, S.A. (2004) Process Modeling Notations and Workflow Patterns. IBM Corporation, BP Trends, March. <http://www.bptrends.com>
- [7] Zhang, J., Kuc, D. and Lu, S. (2012) Confucius: A Tool Supporting Collaborative Scientific Workflow Composition. *IEEE Transactions on Services Computing*, **7**, 2-17.
- [8] Deelman, E., Gannon, D., Shields, M. and Taylor, I. (2008) Workflows and e-Science: An Overview of Workflow System Features and Capabilities. *Future Generation Computer Systems*, **25**, 528-540. <https://doi.org/10.1016/j.future.2008.06.012>
- [9] Ouyang, C., Adams, M., Wynn, M.T. and ter Hofstede, A.H.M. (2010) Workflow Management. In: *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems*, Springer-Verlag, Berlin, 387-418. [https://doi.org/10.1007/978-3-642-00416-2\\_18](https://doi.org/10.1007/978-3-642-00416-2_18)
- [10] Zhao, Y., Li, Y.F., Raicu, I., Lu, S.Y. and Zhang, X. (2014) Architecting Cloud Workflow: Theory and Practice. 2014 *IEEE International Conference on Computer and Information Technology*, Xi'an, 11-13 September 2014, 466-473. <https://doi.org/10.1109/CIT.2014.81>
- [11] Patnaik, S. and Sulatna, T. (2018) Workflow in Treatment Process. *International Journal of Trend in Scientific Research and Development*, **2**, 582-584.



- <https://doi.org/10.31142/ijtsrd18564>
- [12] Lazowska, E.D., Zahorjan, J., Graham, G.S. and Sevcik, K.C. (1984) Quantitative System Performance: Computer System Analysis Using Queuing Network Models. Prentice-Hall, Inc., Upper Saddle River.
  - [13] United Nations (2018) United Nations E-Government Survey 2018. UN Department of Economic and Social Affairs. European Commission: The EU E-Government Action Plan 2016-2020: Accelerating the Digital Transformation of Government. Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions; Brussels, 19.4.2016, in Line with COM (2016) 179 Final.  
<https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52016DC0179>
  - [14] International Telecommunication Union (2008) Electronic Government for Developing Countries. ICT Applications and Cybersecurity Division, Policies and Strategies Department; ITU Telecommunication Development Sector.  
<http://www.itu.int/ITU-D/cyb/app/e-gov.html>
  - [15] Moukeli, M.P. and Mackaya, M. (2017) Assessment of E-Government Development in the Economic and Monetary Community of Central African States (EMCCAS). *International Journal of Emerging Trends & Technology in Computer Science*, **6**, 25-34. <http://www.ijetcs.org>
  - [16] Jeremy, R., Stouby, P.J., Lise, T.H. and Zahir, I. (2015) Managing e-Government: Value Positions and Relationships. *Information Systems Journal*, **25**, 531-571.  
<https://doi.org/10.1111/isj.12052>
  - [17] Bauereiss, T. and Hutter, D. (2014) Possibilistic Information Flow Control for Workflow Management Systems. *Electronic Proceedings in Theoretical Computer Science*, **148**, 47-62. <https://doi.org/10.4204/EPTCS.148.4>
  - [18] Chang, J. and Blei, D.M. (2010) Hierarchical Relational Model for Document Networks. *The Annals of Applied Statistics*, **4**, 124-150.  
<https://doi.org/10.1214/09-AOAS309>
  - [19] Moukeli, M.P. and Nembe, J. (2016) A Stochastic Model for Document Processing Systems. *International Journal of Information Engineering and Electronic Business*, **5**, 52-59. <http://www.mecs-press.org>  
<https://doi.org/10.5815/ijieeb.2016.05.07>
  - [20] Moukeli, M.P., Moukhoukou, A., Naccache, D. and Tskhovrebashvili, N. (2019) A Stochastic Model for Simple Document Processing. *International Journal of Information Technology and Computer Science*, **7**, 43-53.  
<http://www.mecs-press.org/ijitcs>  
<https://doi.org/10.5815/ijitcs.2019.07.06>
  - [21] Gross, D. and Harris, C.M. (1998) Fundamentals of Queuing Theory. Wiley, Hoboken.
  - [22] Jaiswal, N.K. (1968) Priority Queues. Mathematics in Science and Engineering, Volume 50, Academic Press, Cambridge.
  - [23] Chen, H. and Yao, D. (2001) Fundamentals of Queuing Networks: Performance, Asymptotic and Optimization. Springer, Berlin.
  - [24] Singh, M.N. and Yadav, R. (2014) Research Paper on Stack and Queue. *International Journal of Innovative Research in Technology*, **1**, 584-586.
  - [25] Obzherin, Y.E. (2016) Semi-Markovian Model of Two-Line Queuing System with Losses. *Intelligent Information Management*, **8**, 17-26.  
<https://doi.org/10.4236/iim.2016.82003>

- [26] Kopp, V.Y., Obzherin, Y.E. and Peschansky, A.I. (2011) Stationary Characteristics of Queueing System GI/M/2/0. Collection of Scientific Papers of Sevastopol National University of Nuclear Energy and Industry, **4**, 191-197.
- [27] Bolch, G., Greiner, S., de Meer, H. and Trivedi, K.S. (2006) Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications. Second Edition, Wiley-Interscience, Hoboken.  
<https://doi.org/10.1002/0471791571>
- [28] Gautam, N. (2012) Analysis of Queues: Methods and Applications. CRC Press, Boca Raton.
- [29] Coleman, D.R., Belov, L., Basche, T.F. and Cotte, P.A. (2001) Method and Apparatus for Managing and Navigating within Stacks of Document Pages. United States Patent, No. US 6,262,732 B1, Jul. 17, 1-7.
- [30] Takagi, H. (1991) Queueing Analysis: A Foundation of Performance Evaluation. Volume 1: Vacation and Priority Systems. *ACM SIGMETRICS Performance Evaluation Review*, **19**, No. 2. <https://doi.org/10.1145/122564.1045501>