Journal of Applied Mathematics and Physics, 2025, 13(8), 2608-2620
https://www.scirp.org/journal/jamp
ISSN Online: 2327-4379
ISSN Print: 2327-4352

Automated Tools for Creating Student Mark Assessment and Rating Grids for Burundi's Baccalaureate Master Doctorate System

Hilaire Nkunzimana, Emile Niyongabo, Reverien Ndaboroheye, Jérémie Ndikumagenge

Center for Research in Infrastructure, Environment and Technologies (CRIET), University of Burundi, Bujumbura, Burundi
Email: hilaire.nkunzimana@ub.edu.bi, niyemi2014@gmail.com, revendab86@gmail.com, jeremie.ndikumagenge@ub.edu.bi

How to cite this paper: Nkunzimana, H., Niyongabo, E., Ndaboroheye, R. and Ndikumagenge, J. (2025) Automated Tools for Creating Student Marks Assessment and Rating Grids for Burundi's Baccalaureate Master Doctorate System. *Journal of Applied Mathematics and Physics*, **13**, 2608-2620.
https://doi.org/10.4236/jamp.2025.138148

Received: June 15, 2025
Accepted: August 17, 2025
Published: August 20, 2025

Abstract

Tools for modeling and design that have been scientifically validated are needed to develop automated systems that possess the required functional standards. A number of variables must come together in order to create an automated system that satisfies the technological requirements and set standards. These include the degree of flexibility and conformity to the application domains, the application domain's nature and environment, the functional requirements, and the resources that are available. These also involve the rigor of the modeling and design tool selection. This paper aims to develop a model for the automated generation of student mark assessments and rating grids in Burundi's Baccalaureate Master Doctorate (BMD) system using Petri nets. The last will be followed by a study of its structural and/or behavioral properties of the automated system represented by the Petri nets, as well as the algorithm's temporal complexity establishment.

Keywords

Modeling, Petri Nets, Incidence Matrix, Functional Characteristics and Properties, Iterative Algorithm, Mark Assessments and Ratings Grids

1. Introduction

The implementation of automated systems with specific functionalities and meeting specific reliability criteria requires modelling and simulation work at the pre-design stage [1].

Petri nets are a recent development among formal mathematical modelling tools. However, they have already proved to be suitable formal tools for the design and implementation of automated management systems. Indeed, since their invention in the 1960s, Petri nets, with their vast and simple ontological base, have offered developers of automated systems a range of methods for formally studying the properties and therefore the functional specifications of the future system at the modelling stage [2].

Scholars and researchers such as Bouali Mohamed [2] [3] have emphasized the need of conducting an a priori analysis of the future system's behavior during the pre-design stage, as well as the future system's technical and functional specifications, before it is implemented. The goal of this upstream activity is to reduce design, implementation, and operational costs and expenses. The future system is modeled, designed, and synthesized through use of several and various analysis approaches, such as coverage graphs, marking graphs, Farkas formula, and the Gauss pivot method [3].

The operational management of the BMD system at the University of Burundi necessitates an integrated management system due to the volume and complexity of student assessment duties at multiple levels, which vary from course allocation to teacher to students' grids marks publication on a regular basis. In this instance, utilizing Petri nets or similar modelling tools appears to be a strategic and reasonable choice for reducing or minimizing implementation costs while ensuring future system's stability.

The current work effort will provide a model of a future system for the automatic design of student assessment ratings in the Burundi's BMD system, which will be part of the system for managing academic activities and processes within the Burundi High Education System.

2. Materials, Tools and Methods

2.1. Materials and Tools

This work will be done out using data from students' ratings grids who were routinely enrolled in the course register in the BMD system at the University of Burundi throughout prior 2022 academic year. The marks model forms and ratings grids assessment framework are in agreement with the administrative and documents that regulate Burundi's BMD system.

Carl Adam Petri, a German mathematician, developed a modelling technique known as Petri nets. He developed a widely applicable mathematical tool for expressing the links between conditions and occurrences, as well as modelling the behavior of discrete-event dynamical systems. **Figure 1** depicts the overall use and role of the aforementioned abstract mathematical modelling tool.

The Gaussian pivot technique is used to extract structural properties from transition invariants. They rely on the incidence matrix, which is the system's action vector represented by Petri nets, rather than labelling. They allow for the investigation of network structure in the absence of dynamic evolution. Behavioral features, on the other hand, allow for the labelled analysis of Petri nets as well as the study of network structure based on the system's dynamic behavior evolution.

In this work, we want to employ Petri nets as tools to analyze the functional and technological specifications of the future system, with the goal of ensuring a particular level of reliability.

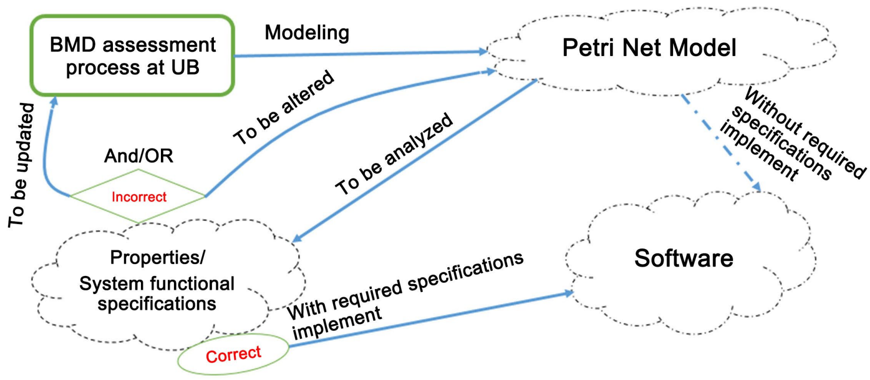

Figure 1. An overview of the Petri nets structural environment.

2.2. Methods

As shown in **Figure 1**, the design approach begins by simulating a non-computerized information system: the BMD System Assessment approach at the University of Burundi. The system is then described or simulated with Petri nets. The next stage is to evaluate the system attributes for accuracy and ensure that they meet the functional requirements. Finally, presuming that all of the requirements are met, we will put the software program into operation. Otherwise, we will adjust the BMD System assessment Process at the University of Burundi or the Petri Net model, if necessary.

The design of the future automated tools for generating student mark assessments and rating grids is based on Petri net analysis methods, including the transition invariant method. The principles for developing a procedure that can be executed by a programmable logic controller for the mark assessments and rating grids generation make it possible to develop an automated module for preparing such dashboard easily.

3. Results and Discussion

The current work includes a Petri net model that represents the future system to be developed and implemented, as well as an iterative algorithm written in Java language style, the effectiveness of which is demonstrated by its quadratic complexity in the worst-case.

3.1. Petri Net Model

A Petri net (RdP) is a directed network or graph with two types of nodes: places (circles) and transitions (bars) including directed arcs connecting the nodes. An arc connects a place to a transition or a transition to a place, never between nodes with same types. As described above, the Petri net is a bipartite oriented graph [2].

A place is a state variable of the system to be simulated, whereas a transition is an event or action that causes changes in the system's state variables. A circle at any given time contains a specific number of markers or tokens that will change over time. The last represents the current value of the state variable. Each arc is allocated a precise value known as its weight [2] [3]. An unlabeled Petri net is a bipartite graph consisting of places, transitions, and arcs that connect transitions and places [1]. It is represented by a quadruplet $Q = \langle P, T, I, O \rangle$, where:

$P = \{p_1, p_2, \cdots, p_m\}$ or $P = \{p_i\}$, $i \in \{1, \cdots, n\}$ is a finite, non-empty set of places.

$T = \{t_1, t_2, \cdots, t_n\}$ or $T = \{t_j\}$, $j \in \{1, \cdots, m\}$ is a finite, non-empty set of transitions.

$P \cap T = \varnothing$: the sets P and T are disjoint.

$|P \cup T| = |P| + |T|$: the total number of nodes in the graph.

The forward incidence application, $I : P \times T \to \mathbb{N}$, uses a set of natural integers to represent direct arcs between places and transitions.

The backward incidence application, $O : T \times P \to \mathbb{N}$, uses a set of natural integers to represent the direct arcs connecting transitions and places [1]. A Petri net, as seen in **Figure 2**, can depict the process of creating a mark-rating grid. **Table 1** summarizes the structure of the aforementioned graph, which represents the new system for constructing mark ratings grids, where **Figure 1** depicts the Petri net of the future system for preparing and generating student mark assessments and ratings grids [4].

The assessment system begins when the required quantity of tokens is present at the transition's entry points. As shown in **Figure 2**, place P° has a token in its place. From then, assuming the RdP is still alive for a certain number of Petri Nets marking states, the modelled system generates grid mark assessment states based on the specified University of Burundi criteria and rules.

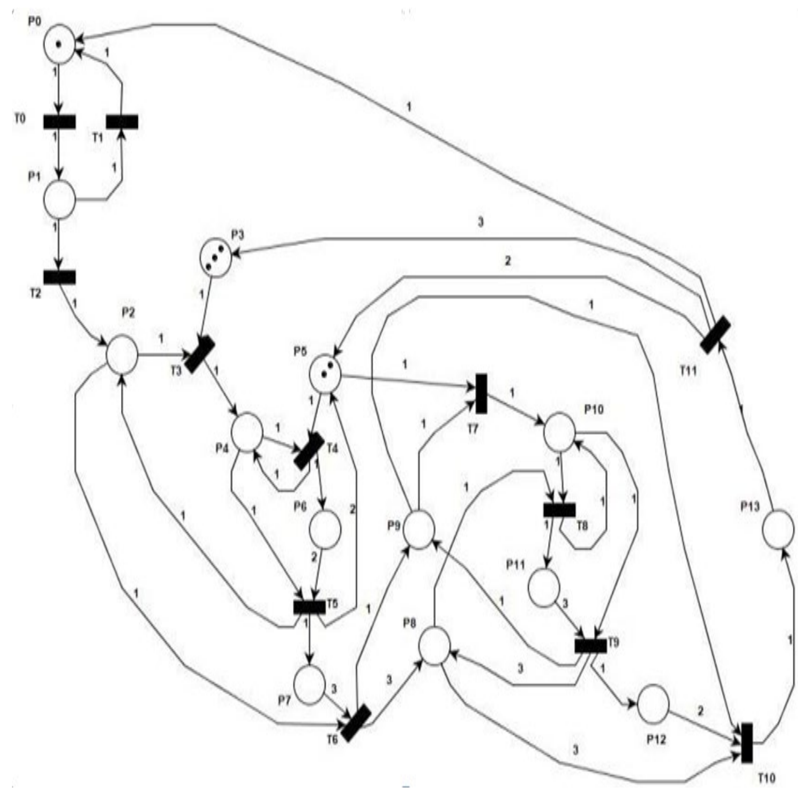

Figure 2. Burundi's BMD system uses a Petri net architecture to model student marking, evaluation, and rating grids.

Table 1. Places and transitions respectively modelling the objects and events of the future system.

Places	Description of the role in the model	Transitions	Description of the role in the model
P_0	Models a teacher who wants to connect	T_0	Models the authentication of the teacher
P_1	Models the authentication parameters entered by a teacher	T_1	Models failed authentication
P_2	Model the connected teacher	T_2	Models successful authentication
P_3	Models the number of courses allocated and assessed	T_3	Models the choice of course
P_4	Models the course chosen by the teacher to create a form	T_4	Models the selection of a student and registration of the mark received in the course
P_5	Models the number of students registered for a given exam session	T_5	Models the creation of a mark form for a selected course
P_6	Models the number of students awarded a mark in a course	T_6	Models the verification of records created whether they are equal to the number of courses and the activation of the deliberative process
P_7	Models the number of mark records created	T_7	Models the selection of a student for the deliberative purpose
P_8	Models the marks created and available (This must be equal to the number of courses)	T_8	Models the path taken by the marks records to fill in the marking grid for a selected student
P_9	Models the possibility of deliberating a or on student	T_9	Models the deliberated student
P_{10}	Models a student under deliberative process	T_{10}	Models the generation of the marking grid if the number of students enrolled is equal to the number of deliberated students
P_{11}	Models the marks obtained by a student in its various courses	T_{11}	Models the end of the deliberation process
P_{12}	Models the number of deliberate students	-	-
P_{13}	Models the deliberation marking grid developed and generated	-	-

3.2. Analysis of the Liveness and Reinitializability Proprieties

- **Incidence matrix**

We intend to use the incidence matrix or action vector calculated from the incidence matrix "After" W^+ and the incidence matrix "Before" W^- to shed light on the behavioral properties or functional requirements of the system simulated using Petri nets [2].

Incidence matrix "After"

$$W^+ = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Incidence matrix "Before"

$$W^{-}=\begin{pmatrix}1&0&0&0&0&0&0&0&0&0&0&0\\0&1&1&0&0&0&0&0&0&0&0&0\\0&0&0&1&0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0&0&0\\0&0&0&0&1&1&0&0&0&0&0&0\\0&0&0&0&1&0&0&1&0&0&0&0\\0&0&0&0&0&2&0&0&0&0&0&0\\0&0&0&0&0&0&3&0&0&0&0&0\\0&0&0&0&0&0&0&0&1&0&3&0\\0&0&0&0&0&0&0&1&0&0&1&0\\0&0&0&0&0&0&0&0&1&1&0&0\\0&0&0&0&0&0&0&0&0&3&0&0\\0&0&0&0&0&0&0&0&0&0&2&0\\0&0&0&0&0&0&0&0&0&0&0&1\end{pmatrix}$$

Incidence matrix:

$$W=W^{+}-W^{-} \tag{1}$$

$$W=\begin{pmatrix}-1&+1&+0&+0&+0&+0&+0&+0&+0&+0&+0&+1\\+1&-1&-1&+0&+0&+0&+0&+0&+0&+0&+0&+0\\+0&+0&+1&-1&+0&+1&-1&+0&+0&+0&+0&+0\\+0&+0&+0&-1&+0&+0&+0&+0&+0&+0&+0&+3\\+0&+0&+0&+1&+0&-1&+0&+0&+0&+0&+0&+0\\+0&+0&+0&+0&-1&+2&+0&-1&+0&+0&+0&+2\\+0&+0&+0&+0&+1&-2&+0&+0&+0&+0&+0&+0\\+0&+0&+0&+0&+0&+1&-3&+0&+0&+0&+0&+0\\+0&+0&+0&+0&+0&+0&+3&+0&-1&+3&-3&+0\\+0&+0&+0&+0&+0&+0&+1&-1&+0&+1&-1&+0\\+0&+0&+0&+0&+0&+0&+0&+1&+0&-1&+0&+0\\+0&+0&+0&+0&+0&+0&+0&+0&+1&-3&+0&+0\\+0&+0&+0&+0&+0&+0&+0&+0&+0&+1&-2&+0\\+0&+0&+0&+0&+0&+0&+0&+0&+0&+0&+1&-1\end{pmatrix}$$

- **Transition invariants**

The invariants ensure that the future system has a certain level of functional or behavioral reliability and robustness, due to the absence of functional blocking of part or all of the system and the return to the initial state, regardless of the system's state at given time.

Transition invariants allow researchers to investigate two essential aspects of the Petri net: liveliness and reinitializability. With these features, the model ensures an adequate level of reliability and resilience for the future system. The goal of liveliness is to determine whether the system has the ability to reproduce or skip all transitions at any given time.

A transition $t \in T$ is alive if crossing t is always possible for any accessible marking *M*, regardless of the system's evolution [2].

A Petri net is repetitive if it has an initial marking (M_0) and a sequence of crossings (S_j) from M_0 that always leads back to M_0. In terms of the incidence matrix, a Petri net is repetitive if and only if there is a strictly positive vector $S_j > 0$ with $W \cdot S_j = 0$ [2].

A t-invariant is an n-vector S_j with non-negative integer members such that [2]:

$$M_j = M_0 + W \cdot S_j \Leftrightarrow M_j = M_0 \text{, because } W \cdot S_j = 0 \tag{2}$$

Using the incidence matrix below, we can calculate the transition invariants.

$$W = \begin{pmatrix} -1 & +1 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +1 \\ +1 & -1 & -1 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 \\ +0 & +0 & +1 & -1 & +0 & +1 & -1 & +0 & +0 & +0 & +0 & +0 \\ +0 & +0 & +0 & -1 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +3 \\ +0 & +0 & +0 & +1 & +0 & -1 & +0 & +0 & +0 & +0 & +0 & +0 \\ +0 & +0 & +0 & +0 & -1 & +2 & +0 & -1 & +0 & +0 & +0 & +2 \\ +0 & +0 & +0 & +0 & +1 & -2 & +0 & +0 & +0 & +0 & +0 & +0 \\ +0 & +0 & +0 & +0 & +0 & +1 & -3 & +0 & +0 & +0 & +0 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +3 & +0 & -1 & +3 & -3 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +1 & -1 & +0 & +1 & -1 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +0 & +1 & +0 & -1 & +0 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +1 & -3 & +0 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +1 & -2 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +1 & -1 \end{pmatrix}$$

List the columns $C_k, k \to 1$ to n that cancel a column $C_l, l \to 1$ to m using the Gauss pivot method.

Using the Gauss pivot method, the column T_0 cancels out by combining the columns T_0 and T_1 while the column T_2 cancels out by combining the columns T_0, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, T_{10} et T_{11}

1) $T_0 = T_0 + T_1$
2) $T_2 = T_0 + T_2 + 3T_3 + 6T_4 + 3T_5 + T_6 + 2T_7 + 6T_8 + 2T_9 + T_{10} + T_{11}$

We have the sequences of the following transitions, which allow us to return to the initial state:

So, for

1) $S_1 = \langle T_0 T_1 \rangle$
2) $S_2 = \langle T_0 T_2 3T_3 6T_4 3T_5 T_6 2T_7 6T_8 2T_9 T_{10} T_{11} \rangle$

These can and should be expressed in matrix form, as shown in expressions 3 and 4. The elements of each matrix are the coefficients of the different transitions that make up each sequence of transitions, i.e. the number of times a transition appears in each crossing sequence. The resulting matrices must be column matrices.

1) $S_1 = (1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0)^{\mathrm{T}}$
2) $S_2 = (1\,0\,1\,3\,6\,3\,1\,2\,6\,2\,1\,1)^{\mathrm{T}}$

The sum of the initial marking matrix with the product matrix between the incidence matrix and a vector of S_q sequences confirms the accuracy of the assertion *Z*.

$$M_k = M_0 + \begin{pmatrix} -1 & +1 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +1 \\ +1 & -1 & -1 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 \\ +0 & +0 & +1 & -1 & +0 & +1 & -1 & +0 & +0 & +0 & +0 & +0 \\ +0 & +0 & +0 & -1 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +3 \\ +0 & +0 & +0 & +1 & +0 & -1 & +0 & +0 & +0 & +0 & +0 & +0 \\ +0 & +0 & +0 & +0 & -1 & +2 & +0 & -1 & +0 & +0 & +0 & +2 \\ +0 & +0 & +0 & +0 & +1 & -2 & +0 & +0 & +0 & +0 & +0 & +0 \\ +0 & +0 & +0 & +0 & +0 & +1 & -3 & +0 & +0 & +0 & +0 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +3 & +0 & -1 & +3 & -3 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +1 & -1 & +0 & +1 & -1 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +0 & +1 & +0 & -1 & +0 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +1 & -3 & +0 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +1 & -2 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +1 & -1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = M_0 \quad (3)$$

$$M_j = M_0 + \begin{pmatrix} -1 & +1 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +1 \\ +1 & -1 & -1 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 \\ +0 & +0 & +1 & -1 & +0 & +1 & -1 & +0 & +0 & +0 & +0 & +0 \\ +0 & +0 & +0 & -1 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +3 \\ +0 & +0 & +0 & +1 & +0 & -1 & +0 & +0 & +0 & +0 & +0 & +0 \\ +0 & +0 & +0 & +0 & -1 & +2 & +0 & -1 & +0 & +0 & +0 & +2 \\ +0 & +0 & +0 & +0 & +1 & -2 & +0 & +0 & +0 & +0 & +0 & +0 \\ +0 & +0 & +0 & +0 & +0 & +1 & -3 & +0 & +0 & +0 & +0 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +3 & +0 & -1 & +3 & -3 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +1 & -1 & +0 & +1 & -1 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +0 & +1 & +0 & -1 & +0 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +1 & -3 & +0 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +1 & -2 & +0 \\ +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & +1 & -1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 1 \\ 3 \\ 6 \\ 3 \\ 1 \\ 2 \\ 6 \\ 2 \\ 1 \\ 1 \end{pmatrix} = M_0 \quad (4)$$

Using the state change formula and with the starting point (the initial marking), when we cross the crossing sequences S_1 and S_2, we return to the starting point, namely to the initial marking of that Petri Net.

So $M_j = M_0 + W \cdot S_1 = M_0$ and $M_j = M_0 + W \cdot S_2 = M_0$

3.3. Algorithm for Generating Student Mark Assessments and Rating Grids for Burundi's BMD System

Listing 1 represents the process of creating student mark assessments and rating grids for Burundi's BMD System, and it reflects the typical scenario. Because the number of students and courses is usually large, the marking grids' matric dimensions are frequently increased.

As a result, determining the temporal complexity is critical for developing academic software applications. The next session will concentrate on the complexity concerns around computationally acceptable time [5] [6].

```
Algorithme UB_Deliberation
Number NcoursEv, NetudiantInscrit, NfichesCreee, recupereNetudiant, recupereNfiche, recupereNcours, etudiantDelibere;
Boolean login,pwd,EtatProfConnect =false;

Begin
  NfichesCreee =0;
  etudiantDelibere =0;
  Set(login);
  Set(pwd);

  While(!login && !pwd)
    Set(login);
    Set(pwd);
  EndWhile

  EtatProfConnect = true;
  /*Saisir le nombre d'etudiants*/
  Set(NetudiantInscrit) ;

  While(NetudiantInscrit<=0)
    Set(NetudiantInscrit);
  EndWhile

  Set(NcoursEv);

  While(NcoursEv <=0)
    Set(NcoursEv);
  EndWhile

  /*Creation de la fiche*/
  recupereNcours= NcoursEv;
  while(NcoursEv>0)
    Get("Saisir le nom du cours");
    NfichesCreee++ ;
    NcoursEv--;
  EndWhile

  recupereNetudiant = NetudiantInscrit;
  recupereNfiche = NfichesCreee ;

  While(NfichesCreee > 0)
    NetudiantInscrit = recupereNetudiant ;
    While(NetudiantInscrit>0)
      Get("Saisir la note de l'étudiant dans la fiche");
      NetudiantInscrit-- ;
    EndWhile
    NfichesCreee -- ;
  EndWhile

  /*Récupération de la note de l'étudiant dans la fiche*/
  EtatProfConnect = false ;
  NetudiantInscrit = recupereNetudiant ;

  If(recupereNfiche== recupereNcours)
    While(NetudiantInscrit >0)
      recupereNfiche = recupereNcours ;
      While(recupereNfiche>0)
        Get("Récupérer la note de l'étudiant dans la fiche");
        recupereNfiche-- ;
      EndWhile
      Get("Délibérer un étudiant");
      etudiantDelibere ++ ;
      NetudiantInscrit -- ;
    EndWhile
  EndIf
  NetudiantInscrit = recupereNetudiant ;

  If(etudiantDelibere== NetudiantInscrit)
    Get("Elaborer et générer une grille de deliberation");
  EndIf

EndAlgorithme
```

Listing 1. Program instructions generating student mark assessment.

3.3.1. The Algorithm Complexity Computation

Time complexity is determined in the worst-case conditions for an algorithm. In the method below at section 3.3.2, (n) represents the execution time based on argument n and c_i represents the time cost of line i. We will assume that the teacher properly entered the login and password, that the number of students enrolled is strictly greater than zero, and that the number of courses assessed is strictly greater than zero too. If this is not the case, the loops will range from 1 to +∞. So this is the circumstance where the teacher is unable to connect, when the number of students and courses is reduced by one.

For the purpose of simplicity, we assume that the instructions in lines 1 - 31 represent the activity's regular course, as this is required for the basic procedure of planning, building, and generating the grid marking and ratings. Under normal conditions, this section of the instructions has a constant complexity of $C_0 \ldots 0$.

$$C_0 = \sum_{t=1}^{31} C_t \tag{5}$$

With C_t equal to 0 for lines 10, 15, 17, 19, 23, 25, 29 and 30, 37, 40, 49, 53, 67. The same applies to line 71, which is zero, and 72, which represents the algorithm's end (stop) Condition.

3.3.2. The Algorithm Complexity Expression

In order to determine the worst-case time complexity of this algorithm, we start at line number 32 and stop at line 70. The algorithm complexity stands as follows:

$$\begin{aligned} T(n) = {} & \sum_{i=NcoursEv}^{1}\left(C_{32}+C_{33}+C_{34}+C_{35}+C_{36}\right)+C_{38}+C_{39} \\ & +\sum_{i=NfichesCreee}^{1}\left(\left(C_{41}+C_{42}\right)+\sum_{j=NetudiantInscrit}^{1}\left(C_{43}+C_{44}\right.\right. \\ & \left.\left.+C_{45}+C_{46}\right)+C_{47}+C_{48}\right)+C_{51}+C_{52}+C_{54} \\ & +\sum_{i=NetudiantInscrit}^{1}\left(C_{55}+C_{56}+\sum_{j=recupereNfiche}^{1}\left(C_{57}+C_{58}\right.\right. \\ & \left.\left.+C_{59}+C_{60}\right)+C_{61}+C_{62}+C_{63}+C_{64}\right)+C_{65}+C_{66}+C_{68} \\ & +C_{69}+C_{70} \end{aligned} \tag{6}$$

$$\begin{aligned} T(n) = {} & \left(C_{32}+C_{33}+C_{34}+C_{35}+C_{36}\right)\sum_{i=NcoursEv}^{1}1+C_{38}+C_{39} \\ & +\sum_{i=NfichesCreee}^{1}\left(\left(C_{41}+C_{42}\right)+\left(C_{43}+C_{44}+C_{45}\right.\right. \\ & \left.\left.+C_{46}\right)\sum_{j=NetudiantInscrit}^{1}(1)+C_{47}+C_{48}\right)+C_{51}+C_{52}+C_{54} \\ & +\sum_{i=NetudiantInscrit}^{1}\left(C_{55}+C_{56}+\left(C_{57}+C_{58}+C_{59}\right.\right. \\ & \left.\left.+C_{60}\right)\sum_{j=recupereNfiche}^{1}1+C_{61}+C_{62}+C_{63}+C_{64}\right) \\ & +C_{65}+C_{66}+C_{68}+C_{69}++C_{70} \end{aligned} \tag{7}$$

$$\begin{aligned} T(n) = {} & \left(C_{32}+C_{33}+C_{34}+C_{35}+C_{36}\right)\times NcoursEv+C_{38}+C_{39} \\ & +\sum_{i=NfichesCreee}^{1}\left(\left(C_{41}+C_{42}\right)+\left(C_{43}+C_{44}+C_{45}+C_{46}\right)\right. \\ & \left.\times NetudiantInscrit+C_{47}+C_{48}\right)+C_{51}+C_{52}+C_{54} \\ & +\sum_{i=NetudiantInscrit}^{1}\left(C_{55}+C_{56}+\left(C_{57}+C_{58}+C_{59}+C_{60}\right)\right. \\ & \left.\times recupereNfiche+C_{61}+C_{62}+C_{63}+C_{64}\right)+C_{65}+C_{66} \\ & +C_{68}+C_{69}+C_{70} \end{aligned} \tag{8}$$

$$\begin{aligned}T(n)=&\left(C_{32}+C_{33}+C_{34}+C_{35}+C_{36}\right)\times NcoursEv+C_{38}+C_{39}\\&+\left(\left(C_{41}+C_{42}\right)+\left(C_{43}+C_{44}+C_{45}+C_{46}\right)\times NetudiantInscrit\right.\\&\left.+C_{47}+C_{48}\right)\sum\nolimits_{i=NfichesCreee}^{1}1+C_{51}+C_{52}+C_{54}\\&+\left(C_{55}+C_{56}+\left(C_{57}+C_{58}+C_{59}+C_{60}\right)\times recupereNfiche\right.\\&\left.+C_{61}+C_{62}+C_{63}+C_{64}\right)\sum\nolimits_{i=NetudiantInscrit}^{1}1+C_{65}+C_{66}+C_{68}\\&+C_{69}+C_{70}\end{aligned}\tag{9}$$

$$\begin{aligned}T(n)=&\left(C_{32}+C_{33}+C_{34}+C_{35}+C_{36}\right)\times NcoursEv+C_{38}+C_{39}\\&+\left(\left(C_{41}+C_{42}\right)+\left(C_{43}+C_{44}+C_{45}+C_{46}\right)\times NetudiantInscrit\right.\\&\left.+C_{47}+C_{48}\right)\times NfichesCreee+C_{51}+C_{52}+C_{54}\\&+\left(C_{55}+C_{56}+\left(C_{57}+C_{58}+C_{59}+C_{60}\right)\times recupereNfiche\right.\\&\left.+C_{61}+C_{62}+C_{63}+C_{64}\right)\times NetudiantInscrit\\&+C_{65}+C_{66}+C_{68}+C_{69}+C_{70}\end{aligned}\tag{10}$$

$$\begin{aligned}T(n)=&C_{38}+C_{39}+C_{51}+C_{52}+C_{54}+\left(C_{32}+C_{33}+C_{34}+C_{35}+C_{36}\right)\\&\times NcoursEv+\left(C_{41}+C_{42}\right)\times NfichesCreee\\&+\left(C_{43}+C_{44}+C_{45}+C_{46}\right)\times NetudiantInscrit\times NfichesCreee\\&+\left(C_{47}+C_{48}\right)\times NfichesCreee+\left(C_{55}+C_{56}\right)\times NetudiantInscrit\\&+\left(C_{57}+C_{58}+C_{59}+C_{60}\right)\times recupereNfiche\times NetudiantInscrit\\&+\left(C_{61}+C_{62}+C_{63}+C_{64}\right)\times NetudiantInscrit\\&+C_{65}+C_{66}+C_{68}+C_{69}+C_{70}\end{aligned}\tag{11}$$

$$\begin{aligned}T(n)=&\left(C_{38}+C_{39}+C_{51}+C_{52}+C_{54}+C_{65}+C_{66}+C_{68}+C_{69}+C_{70}\right)\\&+\left(C_{32}+C_{33}+C_{34}+C_{35}+C_{36}\right)\times NcoursEv\\&+\left(C_{41}+C_{42}\right)\times NfichesCreee+\left(C_{47}+C_{48}\right)\times NfichesCreee\\&+\left(C_{55}+C_{56}\right)\times NetudiantInscrit\\&+\left(C_{61}+C_{62}+C_{63}+C_{64}\right)\times NetudiantInscrit\\&+\left(C_{43}+C_{44}+C_{45}+C_{46}\right)\times NetudiantInscrit\times NfichesCreee\\&+\left(C_{57}+C_{58}+C_{59}+C_{60}\right)\times recupereNfiche\times NetudiantInscrit\end{aligned}\tag{12}$$

$$\begin{aligned}T(n)=&\left(C_{38}+C_{39}+C_{51}+C_{52}+C_{54}+C_{65}+C_{66}+C_{68}+C_{69}+C_{70}\right)\\&+\left(C_{32}+C_{33}+C_{34}+C_{35}+C_{36}\right)\times NcoursEv\\&+\left(C_{41}+C_{42}+C_{47}+C_{48}\right)\times NfichesCreee\\&+\left(C_{55}+C_{56}+C_{61}+C_{62}+C_{63}+C_{64}\right)\times NetudiantInscrit\\&+\left(C_{43}+C_{44}+C_{45}+C_{46}\right)\times NetudiantInscrit\times NfichesCreee\\&+\left(C_{57}+C_{58}+C_{59}+C_{60}\right)\times recupereNfiche\times NetudiantInscrit\end{aligned}\tag{13}$$

Now we know that the variable *recupereNfiche* is equal to *NfichesCreee*, *recupereNfiche* is equal to *NcoursEv*. So we have the equation:

$$\begin{aligned}T(n)=&\left(C_{38}+C_{39}+C_{51}+C_{52}+C_{54}+C_{65}+C_{66}+C_{68}+C_{69}+C_{70}\right)\\&+\left(C_{32}+C_{33}+C_{34}+C_{35}+C_{36}\right)\times NfichesCreee\\&+\left(C_{41}+C_{42}+C_{47}+C_{48}\right)\times NfichesCreee\end{aligned}\tag{14}$$

$$+\left(C_{55}+C_{56}+C_{61}+C_{62}+C_{63}+C_{64}\right)\times NetudiantInscrit$$
$$+\left(C_{43}+C_{44}+C_{45}+C_{46}\right)\times NetudiantInscrit\times NfichesCreee$$
$$+\left(C_{57}+C_{58}+C_{59}+C_{60}\right)\times NfichesCreee\times NetudiantInscrit$$

$$\begin{aligned}T\left(n\right)=&\left(C_{38}+C_{39}+C_{51}+C_{52}+C_{54}+C_{65}+C_{66}+C_{68}+C_{69}+C_{70}\right)\\&+\left(C_{32}+C_{33}+C_{34}+C_{35}+C_{36}+C_{41}+C_{42}+C_{47}+C_{48}\right)\\&\times NfichesCreee+\left(C_{55}+C_{56}+C_{61}+C_{62}+C_{63}+C_{64}\right)\\&\times NetudiantInscrit+\left(C_{43}+C_{44}+C_{45}+C_{46}+C_{57}+C_{58}\right.\\&\left.+C_{59}+C_{60}\right)\times NfichesCreee\times NetudiantInscrit\end{aligned} \tag{15}$$

Assume that *NfichesCreee* and *NetudiantInscrit* are equal *n*, then the equation becomes

$$\begin{aligned}T\left(n\right)=&\left(C_{38}+C_{39}+C_{51}+C_{52}+C_{54}+C_{65}+C_{66}+C_{68}+C_{69}+C_{70}\right)\\&+\left(C_{32}+C_{33}+C_{34}+C_{35}+C_{36}+C_{41}+C_{42}+C_{47}+C_{48}\right)\times n\\&+\left(C_{55}+C_{56}+C_{61}+C_{62}+C_{63}+C_{64}\right)\times n\\&+\left(C_{43}+C_{44}+C_{45}+C_{46}+C_{57}+C_{58}+C_{59}+C_{60}\right)\times n\times n\end{aligned} \tag{16}$$

$$\begin{aligned}T\left(n\right)=&\left(C_{38}+C_{39}+C_{51}+C_{52}+C_{54}+C_{65}+C_{66}+C_{68}+C_{69}+C_{70}\right)\\&+\left(C_{32}+C_{33}+C_{34}+C_{35}+C_{36}+C_{41}+C_{42}+C_{47}+C_{48}\right.\\&\left.+C_{55}+C_{56}+C_{61}+C_{62}+C_{63}+C_{64}\right)\times n\\&+\left(C_{43}+C_{44}+C_{45}+C_{46}+C_{57}+C_{58}+C_{59}+C_{60}\right)\times n\times n\end{aligned} \tag{17}$$

$$\begin{aligned}T\left(n\right)=&\left(C_{38}+C_{39}+C_{51}+C_{52}+C_{54}+C_{65}+C_{66}+C_{68}+C_{69}+C_{70}\right)\\&+\left(C_{32}+C_{33}+C_{34}+C_{35}+C_{36}+C_{41}+C_{42}+C_{47}+C_{48}\right.\\&\left.+C_{55}+C_{56}+C_{61}+C_{62}+C_{63}+C_{64}\right)\times n\\&+\left(C_{43}+C_{44}+C_{45}+C_{46}+C_{57}+C_{58}+C_{59}+C_{60}\right)\times n^{2}\end{aligned} \tag{18}$$

Let $C_{43}+C_{44}+C_{45}+C_{46}+C_{57}+C_{58}+C_{59}+C_{60}=p$,
$C_{32}+C_{33}+C_{34}+C_{35}+C_{36}+C_{41}+C_{42}+C_{47}$
$+C_{48}+C_{55}+C_{56}+C_{61}+C_{62}+C_{63}+C_{64}=p$
et $C_{38}+C_{39}+C_{51}+C_{52}+C_{54}+C_{65}+C_{66}+C_{68}+C_{69}+C_{70}=r$

So

$$T\left(n\right)=pn^{2}+qn+r \tag{19}$$

$$T\left(n\right)\in\theta\left(n^{2}\right) \tag{20}$$

3.4. Discussions

The marked Petri net is an effective design and explanation tool. The Petri net that represents the debating process on the outcomes of the BMD system assessments in Burundi can be reinitialized or repeated. Furthermore, the Petri net representing the future system is alive because all of the transitions are active. The network contains no deadlock transitions. At any time, the system retains the ability to reproduce all transitions. Whatever the system's evolution, it will always be possible to cross all transitions. There is no blocking or deadlock condition since there is no network marking that prevents a transition from occurring.

The algorithm for developing, elaborating, and generating the markings and ratings grids based on the BMD high education system assessment procedures, written in Java programming language, can be implemented without requiring significant changes to the environment.

Finally, for the algorithm, as shown in above Listing, let (n) be the execution time, a function with one argument named n, where n is the data size and c_i is the time cost of that line. The worst-case complexity calculation reveals that the complexity is a quadratic function, which in this situation is considered efficient.

3.5. Conclusions

Implementing the future techniques for developing robust and deliberative discussion on marking and rating grid will allow for easy management of student curriculum. Furthermore, the implementation of an automated system intended to be a part of an integrated management system for national higher education institution, such as the University of Burundi necessitates modeling and simulation work during the pre-design stage. In fact, such activities allow for the detection of almost fatal defects or flaws that would otherwise go undetected without simulation and experimentation on the future system models.

The current work demonstrated that the future system is stable since it is reversible after a finite number of functional steps. Moreover, the absence of deadlock or blocking phenomena while under execution ensures system's liveliness. In reality, whatever condition the system is in during its operation from the beginning state, it will only return to the starting state after it has reached all of the system's possible states. In short, there are no dead parts in the system that could cause a temporary functional stop or halt.

Conflicts of Interest

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

References

[1] Rivière, N. (2003) Modélisation et analyse temporelle par réseaux de Petri et logique linéaire. Doctoral Dissertation, INSA de Toulouse.

[2] Peterson, J.L. (1981) Petri Net Theory and the Modeling of Systems. Prentice Hall PTR.

[3] Bouali, M. (2009) Contributions à l'analyse formelle et au diagnostic à partir de réseaux de Petri colorés avec l'accessibilité arrière. Doctoral Dissertation, Université de Technologie de Compiègne.

[4] Van de Wiele, J.P. (1976) La complexité du calcul des polynômes. https://www.numdam.org/item/?id=AST_1976__38-39__265_0

[5] Brassard, G. and Bratley, P. (1996) Fundamentals of Algorithmics. Prentice Hall.

[6] Jaeger, J., Liang, X.-P., Krebs, R. and Bopp, T. (2014) Automated Protection Performance Assessment and Enhancement. *Journal of Power and Energy Engineering*, **2**, 525-531. https://doi.org/10.4236/jpee.2014.24071