Formal Modeling and Analysis of Data Flow in API-Integrated Systems Using the Petri Net Model

Rustamxon Mexmonov
Associate Professor of the Tashkent Branch of the University of Business and Science, Uzbekistan

Otabek Shukurov
Master's student of the Tashkent Branch of the University of Business and Science, Uzbekistan

Received: 26 Mar 2026 | Received Revised Version: 20 Apr 2026 | Accepted: 03 May 2026 | Published: 30 May 2026

Volume 08 Issue 05 2026 | Crossref DOI: 10.37547/tajiir/Volume08Issue05-08

Abstract

This work considers the issue of integrating Internet of Things (IoT) resources into business process management systems. The modern business environment requires processes to be flexible, reusable, and quickly adapt to different conditions. Therefore, configurable process models are considered an effective approach that reduces the need to develop processes from scratch and forms options that meet business requirements. However, the heterogeneity of IoT devices, resource limitations, energy consumption, computational costs, and interoperability problems complicate their full integration into business processes. Existing studies have mainly analyzed the control flow or human resources, and the perspective of IoT resources has not been sufficiently studied. This work proposes a formal introduction of IoT resources into business process models, taking into account their sharing and redeployment properties, and a configurable distribution approach. The proposed model is evaluated based on real data, and its effectiveness and practical applicability are substantiated.

Keywords: Petri nets, API integration, real-time process, formal modeling, timed Petri net, resource allocation, authentication, prioritization, blocking analysis.

Cite This Article: Rustamxon Mexmonov, & Otabek Shukurov. (2026). Formal Modeling and Analysis of Data Flow in API-Integrated Systems Using the Petri Net Model. The American Journal of Interdisciplinary Innovations and Research, 8(05), 68–77. https://doi.org/10.37547/tajiir/Volume08Issue05-08

1. Introduction

Many organizations have begun to implement process-oriented information systems to benefit from the use of simplified processes, called business process management models, based on predefined models. However, today's dynamic business environment requires flexibility and systematic reuse of business processes. This opportunity is provided by the use of configurable process models. It reduces the need to develop processes from scratch, which is time-consuming, error-prone, and facilitates the joint use of a family of business process options that can be customized to specific business requirements.

Petri nets are interpreted as a formal model that models the behavior of multi-agent systems based on the concepts of events and resources. Positions serve as the most important theoretical basis for representing resources or states, and transitions are the most important theoretical framework for representing events occurring in the system. Whether this article is about API applications, an information system, a network process, an IoT system, an educational platform, or a parallel software module, a Petri net can be adapted as shown in Table 1.

Table 1. Petri net element and its representation

System element	Petri net representation
User request	Initial token
Receive data	Position
Authentication / verification	Transition
Database access	Transition or position
Result processing	Position
Send response to user	Final transition
Error condition	Separate position
Block or wait	Deadlock marking

It can also be used for API processes. In the case of API applications, the model is adapted as follows:

API jarayoni	Petri modeldagi ifodasi
Foydalanuvchi so‘rovi	Fishka
Request qabul qilindi	Pozitsiya
Auth tekshiruvi	O‘tish
Token noto‘g‘ri	Inhibitor yoki xatolik pozitsiyasi
Ma’lumotlar bazasiga so‘rov	O‘tish
Server band	Inhibitor shart
Kritik so‘rov	Yuqori ustuvorlik
Javob qaytarildi	Yakuniy markirovka

If a normal request and an emergency request arrive at the same time, the emergency request is executed first through the π\piπ priority function. If authentication is not performed, the inhibitor condition III blocks the next transition. If the server is busy, the token remains in the waiting position.

2. Literature Review

Petri net theory is one of the important mathematical tools used to formally represent complex, parallel, distributed, and real-time systems. The theoretical basis of this approach was proposed by C. A. Petri, in which computational processes are considered not only as a set of sequential instructions, but also as interconnected states, events, and resource exchanges [1]. In Petri nets, positions represent system states, transitions represent processes or events, and tokens represent available resources or active states. Therefore, Petri nets are an effective formal tool for analyzing API-integrated IoT systems. Because in such systems, sensors, microcontrollers, servers, API requests, databases, and user interfaces constitute multi-stage processes that are executed simultaneously or sequentially. An important advantage of Petri nets is that they allow you to represent the system not only as a graphical diagram, but also as a mathematically verifiable formal model. In simple algorithmic block diagrams, the process is usually described by sequential steps such as "information received - processed - response returned". In Petri nets, the execution condition, resource sufficiency, waiting state, blocking probability, and the possibility of reaching the final state of each state are formally analyzed. The issue of modeling and verification of real-time embedded systems based on Petri nets has been considered, which shows that this approach is important for systems with high requirements for time, reliability, and accuracy [2]. In real-time IoT systems, data coming from sensors must be processed without delay and delivered to the appropriate services via API.

The practical significance of Petri nets is that they propose modeling using APIs, which base Internet services not on simple information pages, but on modules that exchange messages with each other, work together, and perform complex business processes [11], that is, they analyze the interaction between services, the sequence of calls and responses through Petri nets [12] and consider modeling and composition based on high-level Petri nets [13]. These studies provide the basis for formal modeling of processes such as authentication, request acceptance, database access, server response generation, error return, and retry in an API-integrated IoT system.

Research on IoT systems shows the need to use extended forms of Petri nets. In simple Petri nets, chips are considered to have the same properties. [6], who evaluated the operation duration of battery-powered wireless nodes in IoT systems based on High-Coloured Petri Net, modeled the interaction between the sensor device and the geographical environment through Petri nets and noted that response time, fault tolerance, and resource consumption are important indicators in sensor applications [14]. Colored Petri nets allow separating sensor data by type, separately representing errors, and monitoring resource status. This approach serves to simultaneously take into account time costs, cost costs, user reliability, and the reliability of Fog resources [15]. On this basis, it is possible to scientifically justify the processes of initial filtering on the Edge server, aggregation in the Fog layer, data storage in the Cloud layer, and launching the prediction model. In IoT-enabled production-logistics systems, they used Timed Colored Petri Nets to describe real-time states, equipment activity, and production-logistics processes as a self-adaptive model [16]. They proposed a PR-IoT strategy based on Petri Nets for real-time IoT systems, focusing on improving real-time efficiency in multi-connection and multi-task conditions [19]. This shows that the combined use of timed, colored, and priority Petri nets is scientifically justified. Issues of security, authentication, and access rights management are also of particular importance in complex API-IoT systems. They developed a workflow-aware access control approach for IoT systems and showed that workflows and access policies can be expressed based on Petri Nets [7]. This approach is important for modeling API authentication, token verification, and authorized and unauthorized user states. They showed that Colored Petri Nets can be used in complex software system diagnostics to identify connections between software components and identify failure states [4]. In real-time monitoring of agent interactions, the Interaction Petri Net model was used and the focus was on detecting undesirable conditions such as protocol delays, lost messages, busy-wait and transmission delay [8]. It allows for formal analysis of conditions such as API request delays, server busyness or no response. Another important analytical capability of Petri nets is to investigate the issues of boundedness and safety analysis for concurrent control systems based on Petri nets by examining properties such as boundedness, safeness, liveness, reachability and deadlock [9]. Using Petri nets, they evaluated a priority queueing system model and focused on the issue of non-blocking of low-priority traffic in priority queues [18]. This approach serves to justify which process is executed first when simple API requests and critical IoT signals arrive simultaneously, so that low-priority requests do not remain in a long waiting state, and the fair operation of the system. They proposed an architecture for controlling the behavior of IoT devices based on High-Level Petri Nets and performed practical tests on CPN Tools and microcontroller devices [17].

Integrate with API Petri Exam Simulation

Petri nets and their modifications are one of the tools for modeling discrete processes. Petri nets are widely used to solve various problems, and this apparatus is actively being developed. They allow you to describe parallel, asynchronous and hierarchical processes with relatively simple tools. The mathematical model of the process described by a Petri net is very accurate. and is easily algorithmized for computer modeling.

The entire exam process can be represented as a temporary Petri net. As in a simple Petri net, a change in the net label reflects a change in the state of the object being described. The net label changes according to a set of rules, as a result of which the net moves from a given initial state to a certain final state, which is determined by the researcher or automatically by the system if the simulation cannot continue with the current net label.

This modification of Petri nets, such as timed nets, allows you to explicitly include the time of actions in the simulation process, for example, the time between student arrivals, the time to prepare for an answer, etc.

Another addition to the Petri net is the special nodes used for operational control of the object. Their labels can be changed by clicking on the appropriate buttons.

The Petri net modified for the purpose of simulating the exam is shown in Fig.1. The numbers in the circles indicate the number of tokens in the corresponding places (positions) in the net. Each transition represents a defined action with a certain duration. All positions and transitions are described in Table 3.

Implementing a transition involves removing markers from the inputs of each arc and adding additional markers to the outputs of each arc. A transition can be implemented if there are enough markers at all inputs to remove them.

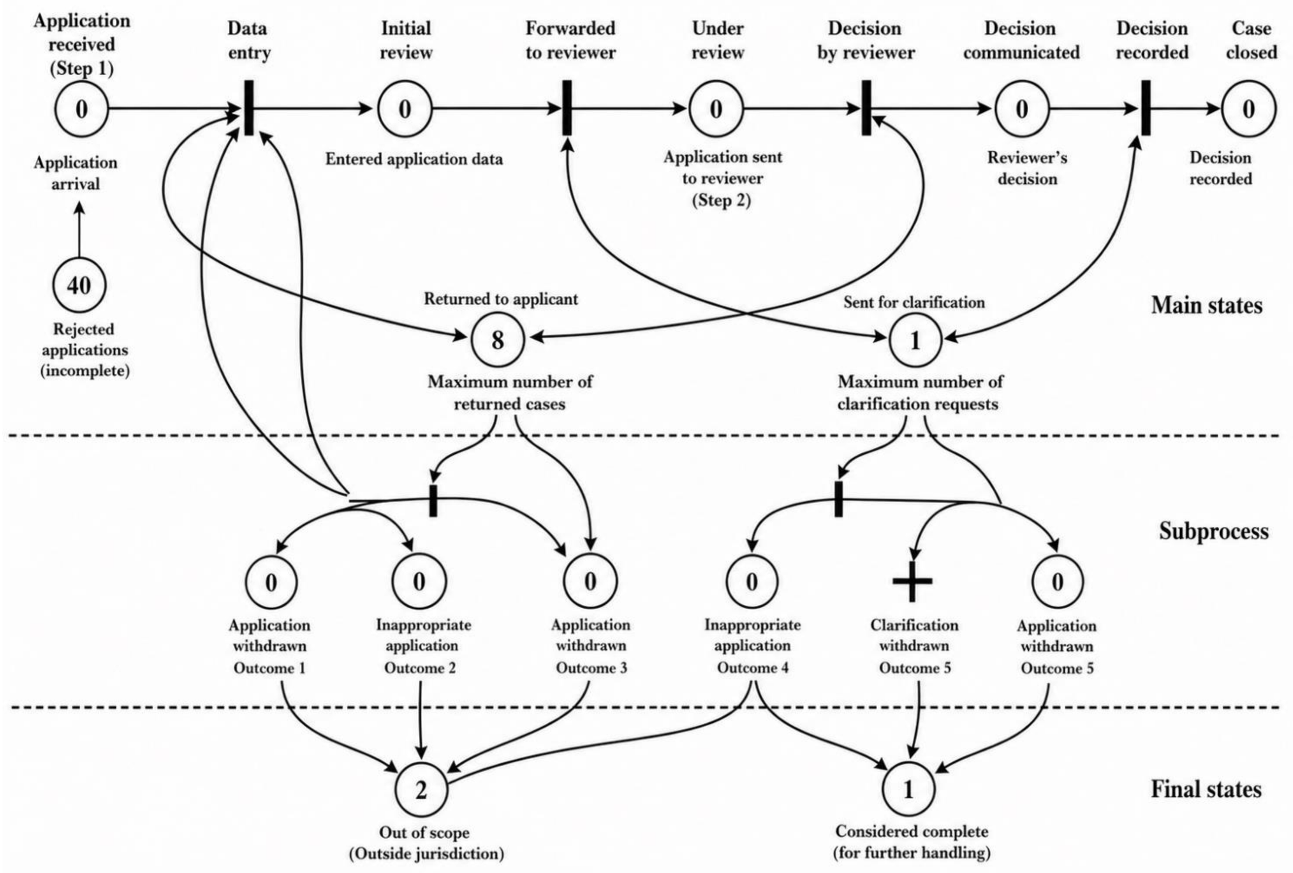


Figure 1. Modeling the selection process based on a delayed Petri net

The operation of the system is characterized by a sequential change in its states, that is, a sequential change in labeling. A change in labeling at the network positions can occur only after a certain transition is made, that is, after performing an action during verification or after the user changes the label of the control nodes.

When implementing a Petri net on a computer, situations arise when it is possible to perform several transitions. Priorities can be used to precisely select one transition to be performed. As noted in the description of the operation object, the operation described earlier in the text has the highest priority.

Looking from left to right and top to bottom in the diagram, the initial default network label is (0, 0, 0, 0, 0, 40, 8, 1, 0, 0, 0, 0, 0, 2, 1). The simulation process continues until all students have been sampled, which corresponds to the labeling (0, 0, 0, 0, 40, 0, …)

Table 1. Description of locations, initial designations, and transitions in Petri net-based exam process modeling

	Places and initial	define
Page 1	Students in the corridor (1st turn) = 0	The number of students who came to the exam and stood at the door due to the lack of free seats or the class being closed
Page 2	Students preparing = 0	The number of students who entered the class and started preparing
Page 3	Students ready to answer (2nd turn) = 0	The number of students who are ready to answer and are waiting for one of the teachers to be free
Page 4	Students who answered = 0	The number of students who answered and teachers who are busy
Page 5	Students who answered = 0	The number of students who have already passed the exam. The exam ends when all students in the group have passed it.
Page 6	Students in the group (not yet arrived) <= 100	The number of students required to take the exam. At the beginning of the simulation, it is equal to the number of students in the group; at the end, it is equal to □0.
Page 7	Vacancies	At the beginning of the simulation, □the number of single-seat tables that can be occupied at the same time in the auditorium. During the simulation, the number of free seats □is determined. This is controlled by the control seats p10 and p11.
Page 8	= 10 p 14 <= 10	At the beginning, □the number of teachers in the exam. During the exam, □, the number of teachers who did not talk to students in the class.
Page 9	Available teachers = 2 p 15 <= 2	Control position for restricting students from entering the auditorium. Pressing the "SPACE" key closes or opens the door.
Page 10	Open/close the auditorium	When the "DOWN" cursor control key is pressed, the number "1" is marked.
Page 11	(Move 1) = 0 < = 1	Control position for arranging the queue of students ready to answer. When the "UP" cursor control key is pressed, the number "1" is marked.
Page 12	Decrease	Control position for adjusting the number of examinees in the room. Pressing the "CTRL" + "DOWN" keys sets the number to one.
Page 13	(Move 2) = 0 < = 1	Control position for adjusting the number of examinees in the room. Pressing the "CTRL" + "UP" keys sets the number to "1".
Page 14	Increase	Number of desks that must remain empty in the auditorium
Page 15	(Move 3) = 0 < = 1	Number of teachers who can be called outside the classroom to conduct the exam
Page 1	Decrease	
t 1	(Move 4) = 0 < = 1	If there are students who have not yet arrived and the passage is not busy, it is performed. The interval between the arrival of students is calculated as follows. The first three arrive in the intervals of 0.01, 0.02, and 0.01 hours, the second four in the intervals of 0.02 to 0.10 hours, the third three in the intervals of 0.03, 0.04, and 0.08 hours, and the rest in the intervals of 0.0 to

		0.5 hours generated by a random number generator according to the exponential law with an average of 0.2.
t 2	Increase	Executed if the classroom door is open and there are empty seats. Duration = 0.0
t 3	(Move 5) = 0 < = 1	This is done when a student arrives in class, in the interval of time obtained using a pseudo-random number generator according to the normal law, with a mathematical expectation of 0.4 hours (24 minutes) and a variance of 0.1 (6 minutes) in the range of 0.1 (6 minutes) to 0.6 hours (36 minutes).
t 4	Reserve seats = 10 p 7 <= 10	This is done when the teacher is available and the student is ready to answer. This opens a vacancy. Duration = 0.0
t 5	Teachers on vacation = 2 p 8 <= 2	The test ends when the student is available. The response time is calculated using a pseudo-random number generator using a uniform distribution in the range of 0.1 (6 minutes) to 0.3 hours (18 minutes).
t 6	Transitions	It is done when there are vacancies and a manager is assigned to reduce vacancies. Duration = 0.0
t 7	Student arrival	It is done when there are spare seats and a manager is assigned to increase vacancies. Duration = 0.0
t 8	Entering the auditorium	It is done when there are teachers in the class and a checkpoint is assigned to reduce the number of examinees. Duration = 0.0
t 9	Preparing for the answer	This is done when there is a designated checkpoint to increase the number of teachers and examinees on the vacation. Duration = 0.0

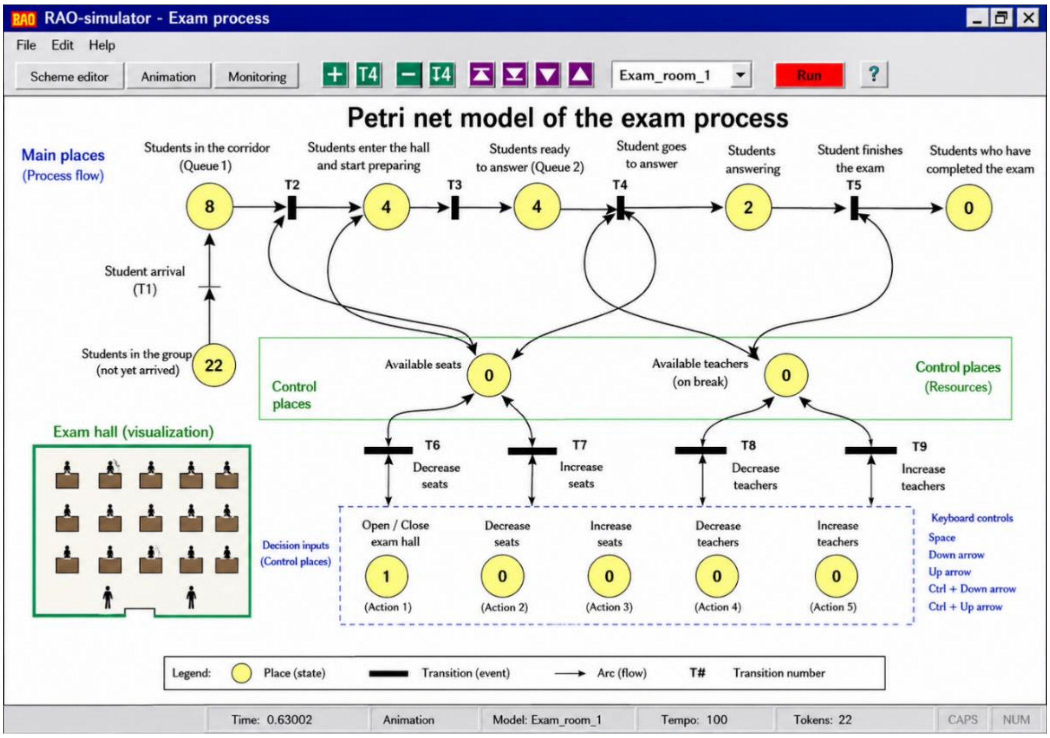
The researcher's goal is to determine the teacher's workload and the time it takes for students to be ready to answer before the examiner appears. To control for these factors, the number of teachers and seats in the classroom, as well as the opening and closing of the entrance door, can be adjusted. It is also possible to change the initial settings for the number of examiners and students in the group at the launch facility and the duration of various operations.

The exam model includes four types of resources (Table 6.2), which are described in more detail in the comments to the parameters.6.2-jadval

Resurs turi	Tavsif
Imtihonlar	Imtihonning umumiy holatini aks ettiruvchi doimiy resurs
O'qituvchilar	O'qituvchining ahvolini aks ettiruvchi doimiy resurs
Stollar	Auditoriyadagi o'rindiqlarning holatini aks ettiruvchi doimiy resurs (animatsiya uchun)
Talabalar	Imtihonga kelgan talabaning imtihonda qolish muddati tugaguniga qadar uning holatini aks ettiruvchi vaqtinchalik resurs

The interactive exam management operations are not only associated with keyboard keys, but also have corresponding active areas in each of the two graphic frames shown in Figures 6.1 and 6.2. In the first frame, these areas are the corresponding positions of the Petri net, and in the second frame, they are the image of a door (open or closed) and triangular buttons for increasing and decreasing the number of teachers and empty seats. The frames can be changed via the frame list.

After creating the model, a series of experiments were conducted to determine the optimal number of empty seats in the classroom when the exams were conducted by two or one teacher. The optimal number of seats was determined as the number that ensures that no more than two people are in line to see the teacher.



Modeled control scheme of the examination process based on Petri nets

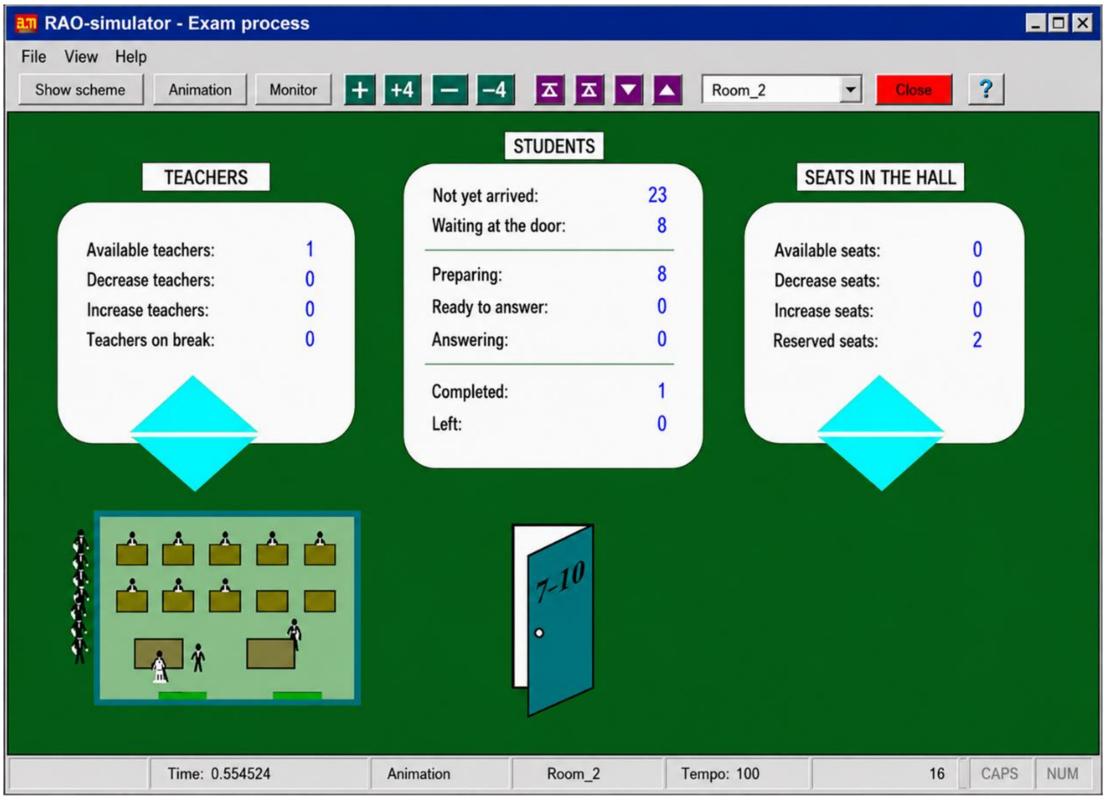


Figure 2. Second frame of the animation

As a result of experiments (Tables 6.3 and 6.4), it was found that when the exam is conducted by two teachers, the exam lasts about 4.5 hours and 7-8 desks should be occupied; when conducted by one examiner, the exam lasts almost 9 hours and 3-4 desks are required.

Parameters \ Experiments	1	2	3	4	5	6
Parameters \ Experiments {Random_number_generator_databases:} Second_fourth.Seed = Exponential_interval.Seed = Normal_Distribution.Seed = Uniform_Distribution.Seed = {Ranges} Group = {1 ..100} Locations = {0 ..10} Teachers = {0 .. 2} {0.1 hour = 6 minutes} Revenue_average = {0.0 .. 0.5 } Mean = {0.1 .. 0.6} Preparation_variability = Response_minute = Response_max =	12312 3 12312 3 12312 3 12312 3 40 9 2 0,2 0,4 0,1 0,1 0,3	99887 7 99887 7 99887 7 99887 7 40 9 2 0,2 0,4 0,1 0,1 0,3	12312 3 12312 3 12312 3 12312 3 40 8 2 0,2 0,4 0,1 0,1 0,3	99887 7 99887 7 99887 7 99887 7 40 8 2 0,2 0,4 0,1 0,1 0,3	12312 3 12312 3 12312 3 12312 3 40 7 2 0,2 0,4 0,1 0,1 0,3	99887 7 99887 7 99887 7 99887 7 40 7 2 0,2 0,4 0,1 0,1 0,3
Queue to see teacher = Teacher_waiting_time = {t busy /t not resting :} Teacher_Workload_1 = Teacher_Workload_2 =	4,1769 0.4815 0.9008 0.8998	4,2341 0.4854 0.9241 0.8585	3,404 0.3914 0.9008 0.8998	3,4858 0.4007 0.9238 0.8651	2,6330 0.3051 0.9008 0.8998	2,6840 0.3111 0.9238 0.8651

Second_fourth.Seed =	12312	99887	12312	99887	12312	99887
Exponential_interval.Seed =	3	7	3	7	3	7
Normal_Distribution.Seed =	12312	99887	12312	99887	12312	99887
Uniform_Distribution.Seed =	3	7	3	7	3	7
Group = {1 ..100}	12312	99887	12312	99887	12312	99887
Places = {0 ..10}	3	7	3	7	3	7
Teachers = {0 .. 2}	12312	99887	12312	99887	12312	99887
{0.1 hour = 6 minutes}	3	7	3	7	3	7
Revenue_average = {0.0 .. 0.5 }	40	40	40	40	40	40
Mean = {0.1 .. 0.6}	5	5	4	4	3	3
Preparation_variability =	1	1	1	1	1	1
Response_minute =						
Response_max =	0.2	0,2	0,2	0,2	0.2	0.2
	0.4	0,4	0,4	0,4	0.4	0.4
	0.1	0,1	0,1	0,1	0.1	0.1
	0.1	0,1	0,1	0,1	0.1	0.1
	0.3	0,3	0,3	0,3	0.3	0.3
Queue to see teacher =	2,7600	2,8019	1,8867	1,9137	0.9917	1,0114
Teacher_waiting_time =	0.6034	0.6005	0.4113	0.4121	0.2165	0.2269
{t busy /t not resting :}						
Teacher_Workload_1 =	0.9473	0.9591	0.9473	0.9591	0.9403	0.9254
Teacher_Workload_2 =	0	0	0	0	0	0

The results of the experiment show that the number of seats in the auditorium and the number of teachers directly affect the main performance indicators of the examination process. In experiments involving 2 teachers, when the number of seats was reduced from 9 to 7, the waiting time for a teacher was reduced from 0.48 hours to 0.31 hours. This indicates that the length of the queue decreases when the flow of students is limited. At the same time, the workload of teachers was kept at a high level - approximately in the range of 0.86 - 0.92.

3. Conclusion

The second table presents the results of experiments with 1 teacher. In this case, when the number of places was reduced from 5 to 3, the waiting time for the teacher decreased from 0.60 hours to 0.22 hours. However, the workload of one teacher was very high, forming a range of 0.92–0.96. Since the second teacher was not present, his workload was equal to 0. In general, reducing the number of places in the auditorium during the examination process regulates the flow of students and reduces the waiting time for the teacher. However, if the number of teachers is small, the workload on the existing teacher increases. Therefore, in order to effectively organize the examination process, it is necessary to choose a balance between the auditorium capacity, student flow, and the number of teachers.

References

1. C.A.Petri, Kommunikation mit Automaten, Ph.D. dissertation, Darmstadt University of Technology, 1962.
2. L. A. Cortés, A Petri Net Based Modeling and Verification Technique for Real-Time Embedded Systems, Ph.D. dissertation, Linköping University, Sweden, 2001.
3. A. Achilleos, Model-Driven Petri Net Based Framework for Pervasive Service Creation, Ph.D. dissertation, 2010.
4. Y.Li, Diagnosis of Large Software Systems Based on Colored Petri Nets, Ph.D. dissertation, Université Paris-Sud XI, France, 2010.
5. D.L.Lages, A Colored Petri Net-Based Model for Internet of Things Networks, thesis, Universidade Federal de Pernambuco, 2024.
6. Y.M.A.Alsarhan, A General High-Coloured Petri Net-Based Approach to the Estimation of the Lifetime of Wireless Battery-Powered Nodes in Internet of Things, Ph.D. dissertation, University of Portsmouth, 2021.
7. P.Kasinathan, Workflow-Aware Access Control for the Internet of Things, Ph.D. dissertation, Universität Passau, 2021.
8. N.Aldahash, Assuring Agent Interaction Through Run-Time Monitoring and Control, Ph.D. thesis, University of York, 2023.
9. M.Wojnakowski, Analysis of Boundedness and Safeness in a Petri Net-Based Specification of Concurrent Control Systems, doctoral dissertation, 2021.
10. K. Suri, Modeling the Internet of Things in Configurable Process Models, Ph.D. thesis, Université Paris-Saclay, 2019.
11. R. Hamadi and B. Benatallah, “A Petri Net-Based Model for Web Service Composition,” in Proc. 14th Australasian Database Conf. (ADC), vol. 17, pp. 191–200, 2003.
12. V. Valero, M. E. Cambronero, G. Díaz, and H. Macià, “A Petri Net Approach for the Design and Analysis of Web Services Choreographies,” Journal of Logic and Algebraic Programming, vol. 78, no. 5, pp. 359–380, 2009.
13. S. Chemaa, F. Bachtarzi, and A. Chaoui, “A High-Level Petri Net Based Approach for Modeling and Composition of Web Services,” Procedia Computer Science, vol. 9, pp. 469–478, 2012, doi: 10.1016/j.procs.2012.04.050.
14. F. Zhang, Y. Xu, and J. Chou, “A Novel Petri Nets-Based Modeling Method for the Interaction Between the Sensor and the Geographic Environment in Emerging Sensor Networks,” Sensors, vol. 16, no. 10, Art. no. 1571, 2016, doi: 10.3390/s16101571.
15. L.Ni, J.Zhang, C.Jiang, C.Yan, and K.Yu, “Resource Allocation Strategy in Fog Computing Based on Priced Timed Petri Nets,” IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1216–1228, 2017, doi: 10.1109/JIOT.2017.2709814.
16. Z.Guo, Y. Zhang, X. Zhao, and X. Song, “A Timed Colored Petri Net Simulation-Based Self-Adaptive Collaboration Method for Production-Logistics Systems,” Applied Sciences, vol. 7, no. 3, Art. no. 235, 2017.
17. J.P.da Silva Fonseca, A. R. de Sousa, and J. J.-P. Z. de Souza Tavares, “Modeling and Controlling IoT-Based Devices’ Behavior with High-Level Petri Nets,” Procedia Computer Science, 2023.
18. D. Strzęciwilk, “Timed Petri Nets for Modeling and Performance Evaluation of a Priority Queueing System,” Energies, vol. 16, no. 23, Art. no. 7690, 2023, doi: 10.3390/en16237690.
19. Y. Bao, “A Design Strategy for Real-Time IoT Systems Based on Petri Net,” International Journal of Sensor Networks, 2023, doi: 10.1504/IJSNET.2023.134904.
20. C.Y. Yang, Y.N. Lin, V. R. L. Shen, F.H.C.Shen, and Y.C. Lin, “Petri Net Modeling and Analysis of an IoT-Enabled System for Real-Time Monitoring of Eggplants,” Systems Engineering, vol. 28, no. 2, pp. 270–283, 2025.