Formal Verification of a Block chain-enabled Smart Learning Environment Framework using Petri NETs

G R Anil[1*], Salman Abdul Moiz[2]
[1,2] School of Computer and Information Sciences, University of Hyderabad, Hyderabad, India 500046

Abstract

Smart learning is a technology-enhanced learning system that is effective, efficient, engaging style of learning. Smart Learning Environment (SLE) is the physical learning environment that supports context aware ubiquitous learning by using sophisticated technologies. The existing Learning Technology Systems Architectures (Frameworks) need to be redesigned to integrate the sophisticated technologies. The enhanced frameworks must reach the higher levels of smartness. The "Smartness levels" are a measure to assess the learning environments. While achieving the higher smart levels, a formal verification of the proposed architectures is required to confirm the smartness levels attained. It is significant as these frameworks are used as reference models for the development of the Learning Management Systems. A Scenario based formal verification procedure of an SLE is proposed in this paper. The formal verification procedure includes representing the a scenario into an activity diagram which is then converted to a Petrinet that is helpful in performing Matrix analysis, Reachability graph analysis, Generalized Stochastic Petri Nets(GSPN) analysis, Simulation, and State-space analysis. For the latest Blockchain enabled SLE framework, an algorithm for online examination scenario is proposed. As this frameworks has more secured mechanism than traditional frameworks, the standardization challenge "security" which is at first two levels of SLE Smartness is addressed. The proposed Algorithm is formally validated using the proposed procedure using petri nets..

Keywords: *Petrinets; Blockchain; Enhancement; Security; Activity Scenario;*

1. Introduction

In the domain of online learning, the IEEE standard of Learning Technology Systems Architecture is a widely known standard framework proposed in the year 2003[10]. It provides a generic Software architecture for the Learning Management systems. In the same year IMS Global Learning Consortium has proposed IMS abstract framework [22]. MIT has proposed (Open Knowledge Initiative)[24]. But in the later period, the OKI was dropped from the archives of MIT due to its impediments. A universally accepted framework stemmed from the Joint Information Systems Council (JISC)[23] from the UK which has the approval of the online learning research community. However, these are not regarded as smart learning frameworks [4][9].

Smart learning is concerned with the context-aware ubiquitous learning [27]. Contexts include the interactions between learner and the learning environments. The smart learning environments comprises of technology enhanced learning environments to fulfill the need to provide right content at right time. The smartness level is a measure to assess the smartness of the smart learning environments [9]. There are certain Standardization challenges, which are associated with the six Smartness levels [4][9].

The existing learning architecture frameworks have their own limitations. They stand at various levels of smartness some of them are not considered to be smart [3][4]. There exist limitations such as missing of some functional areas and could not achieve the current complex requirements [20]. Thus IEEE LTSA standard [10] was withdrawn in the year 2009[20][10]. In the assessment of smartness of the Learning technology frameworks, IEEE LTSA is concluded to be at the Pre-Smartness level[4]. There exists a requirement for the new age smart Learning Environment Frameworks which can adopt the latest sophisticated technologies [4]. There is a challenge in adopting new technologies to the learning environments to enhance the analytical capability of

ISSN: 2233-7857 IJFGCN

the learning environments [4]. This has motivated to address the challenges in Smart Learning Environments.

The existing learning architecture frameworks have their own limitations. They stand at various levels of smartness some of them are not considered to be smart[3][4]. There exist limitations such as missing of some functional areas and could not achieve the current complex requirements[20]. Thus IEEE LTSA standard[10] was withdrawn in the year 2009[20][10]. In the assessment of smartness of the Learning technology frameworks, IEEE LTSA is concluded to be at the Pre-Smartness level[4]. There exists a requirement for the newage smart Learning Environment Frameworks which can adopt the latest sophisticated technologies [4]. There is a challenge in adopting new technologies to the learning environments to enhance the analytical capability of the learning environments[4]. This has motivated to address the challenges in Smart Learning Environments.

To address the most important standardization challenge " Security" which is present in both the first and second levels of smartness(Adapt, Sense), the IEEE LTSA framework is extended[1] . The risk parameters are quantitatively evaluated on the IEEE LTSA framework and Blockchain-enabled Smart learning environment framework [1]. It is concluded that the Blockchain-enabled architecture is more robust, secure and immutable[1]. This paper proposes an algorithmic procedure for storing the online examination responses into the Blockchain-enabled Smart learning environment. There is a need to formally validate the proposed architecture for various scenarios.

The online examination scenario is represented into equivalent activity diagram , as the UML diagrams are powerful mechanism to write software blueprints[2][16]. The UML diagrams are informal notations, thus they are converted into the Petrinets using the procedure proposed by Yasmina[5]. Petrinets are a formal representation of the system that can perform various analysis to prove the correctness of the system. A system design must be safe, bounded and deadlock-free that can be analyzed with the State space analysis of the Petrinets.

The Research contributions of this paper are listed below

- A Procedure to verify an SLE framework
- An Algorithmic procedure for a scenario of the Blockchain storage of Online Examination responses from the students
- Formal analysis of the proposed scenario's procedure for verification

The organization of this paper is as follows. Section II briefs the Related work in the smart learning environment frameworks and Formal validation of architectures. Section III describes the Preliminaries to understand the Petri nets and PIPE tool. Section IV describes the Procedure for formal verification of a Software Architecture based on a Scenario. Section V presents the proposed Algorithm for the Online examination Scenario for Block chain enabled SLE. Section VI perform the Verification of the proposed algorithm. Results and Discussion are presented at Section VII. Section VIII specify the Conclusion and the Future work.

2. RELATED WORK

2.1 Learning Technology Systems Architecture

Learning Technology Standard Committee (LTSC)[10] has proposed a standard, called as IEEE Learning Technology Systems Architecture 1484.1 in 2003. It is a high-level architecture for Online learning that provides an abstract framework. There are 2 data stores, 4 Processes, and 13 data flows as described in the standard [10]. The Learning Resources stores the learning content and the Learner Records is the dedicated data store to save the learner's information. It’s built on the centralized data-store system. The processes in the architecture communicate with the data flows mentioned in the Figure 1. There are security threats due to the centralized storage

ISSN: 2233-7857 IJFGCN

system. The non functional parameters like availability get compromised which is one of the major limitations.

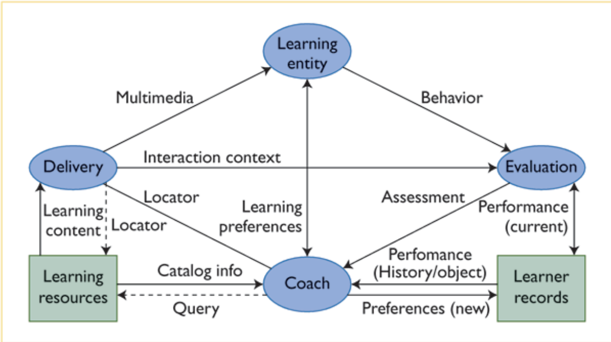

Figure 1 IEEE LTSA framework [10]

Prveen et Al [20] have attempted to convert the centralized data storage schema in LTSA to distributed schema. They have emphasized the need of security in online learning due to continuous evolution of hacking techniques that makes the possibility of attacker's intrusion into information system. Formal evaluation has not been performed to this architecture.

2.2 Threats in Online Learning

The smart learning environments designed are supposed to possess high security and reliability [4][9]. The e-learning process is facing challenges in assessing the learning objectives and in conducting examinations through an online
mechanism, as the evaluated results reflect the impact directly on the learning outcomes[11][12]. The data generated and stored during the assessment must be tamper-proof and immutable.

Abrar et al [12] have classified the threats in remote online examinations into Intrusion based and Non-intrusion based. The threats include impersonation, collusion and abetting. To address these threats, a framework must ensure transparency and non modifiability of the examination data.

Shaibu [19] listed out the security issues in e-Learning and M-learning environments as SQL Code Injection, Cross site scripting, Cross site request forgery, Stack-smashing attacks, Session Hijacking, Denial-of-Service attack (DoS). A Framework design should avoid four types of threats which are Fabrication, Modification, Interruption and interception. This mandates the designer to explore and adopt a robust secure mechanism to the SLE framework.

2.3 Blockchain-based Storage Systems

A Blockchain is a distributed database that has data definition and update mechanism [6]. It allows to add new data as well as ensures that uniform data is present in whole network. The Blockchain is a decentralized linked data structure for retrieval and data storage. The data stored once is resistant to any modification, which is a robust storage mechanism. Blockchain mechanism provides Integrity, Transparency, Immutability, audit-ability and fault tolerance. Muhammad Muzammal et Al [6] proposed an application platform that has Blockchain mechanism. ChainSQL is used in this research work, that explained the components in the Blockchain system. The architecture of this Blockchain based system contain the flow of interaction of Application, Network nodes and Database. This Blockchain system architecture [6] is adopted in our work for designing the Smart Learning Environment Framework.

2.4 Blockchain-enabled Smart Learning Environment Framework

The IEEE LTSA is extended into a Blockchain-enabled Smart Learning Environment framework [1]. As the Blockchain storage is computationally expensive, we have limited the usage of Blockchain storage for the examination data only. The Risk levels of IEEE LTSA with centralized storage system and the proposed architecture are evaluated quantitatively [1]. It was

ISSN: 2233-7857 IJFGCN

concluded that the Risk is minimal in the Blockchain based SLE. The proposed architecture is depicted in the Figure 2.

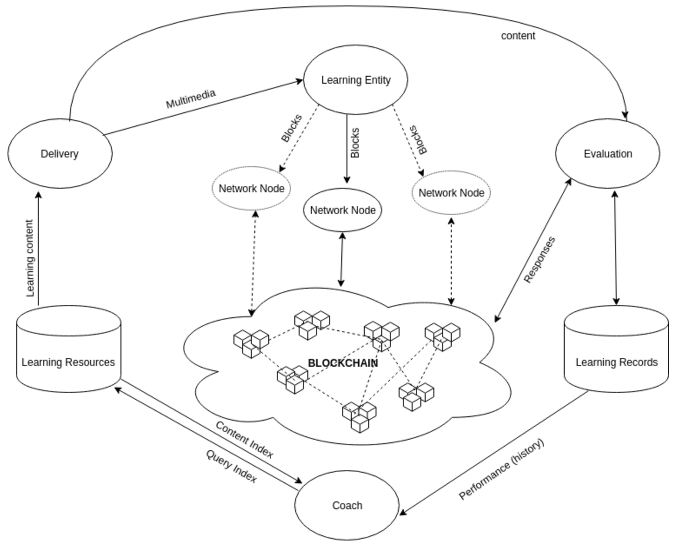

Figure 2 Blockchain-enabled Smart Learning Environment framework [1]

2.5 Security Evaluation of Software Architectures

A. Alkussayer et Al[28] has explained the Software architecture security evaluation methods. They are listed as 1. Experience-based Assessment 2. Mathematical Modeling Assessment 3. Simulation-based Evaluation 4. Scenario-based Evaluation.

In the Experience based assessment, the security architects are involved and their intuitions are considered for assessment. Its a manual assessment thus it is not used as for a formal assessment. The Mathematical model involves the code metrics. This assessment is possible after the code is written. Thus, it is not feasible in the early design stages. Simulation based evaluation requires an executable model of the system, thus design cannot be evaluated with it. Thus Scenario based evaluation can be adopted for the security assessment of the Software Architectures' design.

2.6 Petrinet-based Formal Evaluation of a Software Architecture

A. Pinna et Al [16] proposed a novel approach for analyzing Blockchain with Petri net model. It is mentioned that well defined models allows the structural analysis of straight forward algorithms. Petrinets are generated for the simplified transaction chains. The incidence matrices are computed for the entities Petri net and analyzed. This work can be referred for generate Petri net based formal analysis.

M.I. Fakhir et Al[17] presented a Formal specification and verification of Self-adaptive concurrent systems. Colored Petri nets are used in their formal specification of the framework proposed. A Colored Petri net model for the traffic monitoring system is taken as the case study and validated the same for the Safeness, Liveness, Deadlock free properties.

Ma et Al[25] proposed Petrinet based behavioral analysis of Software Architectures. A Software system is modeled by a Petri net and behavioral analysis is performed on the slices of the Petri net.

Camila Araujo et Al[13] had explained the Formal verification of Software Architecture Description. In the process of formally verifying the systems, Software architecture specification and properties are considered as input and translated to formal notations. This concept can be adopted for converting Software architecture into formal notations. Wenxin et Al[14] explained the specification of Software Architectures using Colored Petrinets.

ISSN: 2233-7857 IJFGCN

Lopez et Al[29] has explained the significance of performing the Formal verification before software development. A formal verification through Petrinets is performed for the Task description languages that are helpful in modeling a task in a workflow.

It can be observed that for assessing the correctness of any Software architecture, it must be formally verified. The Formal notations like Petrinet provide several analytical results for evaluation.

3. Algorithm for the Online examination Scenario for Block chain

The Scenario based formal verification of a framework can be performed with the following procedure. The workflow of each scenario needs to be constructed for this purpose. There can be many scenarios addressing various services of the framework. General verification follows by prioritizing the scenarios and proceeding with the most significant one.

The verification begins with the Requirements of the framework. The requirements are Functional and non-functional. To verify the correctness of the methodology for a scenario, the following steps are to be followed.

1. Specify the workflow of tasks necessary to provide a service
2. Express the workflow in an Activity diagram
3. Convert the Activity diagram (informal notation) into a Petri net (Formal notation)
4. Evaluate and Analyze the Results obtained from the Petri net and Architecture
5. Revise: Repeat the procedure until complete Requirements are addressed.

A workflow for a scenario can be developed if a set of tasks are well defined, predictable and repetitive. The successful completion of any workflow mainly relies on two factors:

- Modeling power
- Decision power

Modeling power is delineated as the capability to represent the system to be modeled, and the decision power is set out to evaluate the model and define the feature of the modeled system. The framework of the workflow needs to be formally represented which works as a powerful tool that yields a proper analysis. This can be better represented with UML diagrams.

Unified Modeling Language (UML) is a powerful mechanism to write software blueprints. The activity diagram, one of the UML diagram that can demonstrate the flow of activities to provide a service. They are helpful in modeling use cases for representing business workflows. They help model the coordination of a collection of use cases for representing business workflows.

In the activity diagram, every event is taken as an activity.

To build an activity diagram,

- The tasks in the workflow need to be identified
- The order of the tasks must be listed
- The initial and final tasks are to be marked.
- The precondition and post condition for each task must be derived
- The Workflow between tasks must be modeled.

The conversion of UML activity diagrams into a Petri net is performed by the framework proposed by Yasmina[5]. This procedure involves identifying the Places and transitions from the activity diagram. The rules for this conversion are The initial node is taken as a Place, An Action to a transition and Every decision to a place. From this procedure, a Petri net can be generated. A Petrinet analysis can conclude the correctness through the Safeness, Liveness(Deadlock free), Boundedness[18].A holistic picture of the projected procedure is depicted in the Figure 3.

ISSN: 2233-7857 IJFGCN

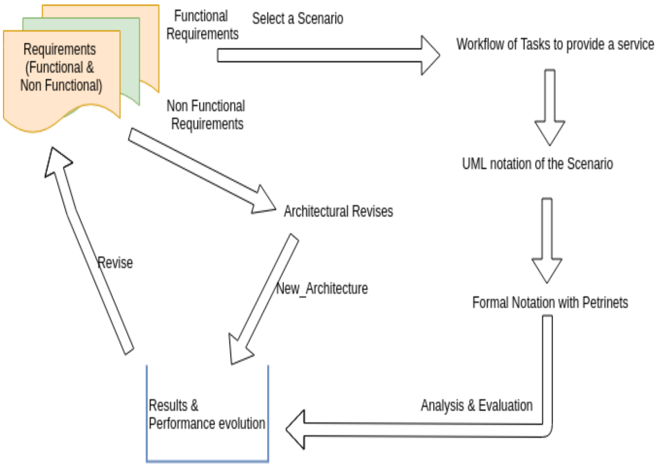

Figure 3 Procedure of Verification of a Software Architecture

4. Algorithm for the Online examination Scenario for Block chain enabled SLE

To address the "Security" threat for the online examination scenario, an Algorithmic procedure in accordance with the Blockchain based Smart Learning Environment framework[1] is presented as below

Data: Exam Activity log = $\{a_{t_0}, a_{t_1}, a_{t_2}, ...a_{t_n}\}$, where a_{t_i} is an activity at instant t_i. Initial time t_0, Current Time t_c, Duration of exam d, $n(t_i)$ the number of transaction from t_0 to t_i, termination time t_t, initially $t_c = t_0$, Final time $t_f = t_0 + d$

Result: The Responses and activities during online examination are stored securely into the Blockchain

```
while t_c < t_f & t_t ≠ NULL do
    if (n(t_c)%s ← 0) then
        flag = t_c ;
        generate(H(a_{t_c})) ;
        P = { collection of the hash values of a_{t_0}, a_{t_1}, a_{t_2},.....
          a_{(t_c−flag)+1} } ;
        if validation_ failed then
            initiate Rescue Mechanism
        else
            t_c = t_c + 1;
            generate(H(a_i)) ;
        end
        current section becomes this one;
    end
end
Upload the final block
Conclude the exam
```

Algorithm 1 : Algorithm for the online examination Scenario

Each exam has a fixed duration '*d*' defined while creating the examination. The Exam Activity log is a set of student responses recorded during the time duration *d*. Each record in the log including other parameters [1] is a transaction. A block consists of a fixed number of transactions. Once a student authenticates and begins an exam at t_0, their responses are stored into the Exam activity log for the duration d. If the current instant of time t_c is less than the final instant of time ($t_f = t_0 + d$) then the transactions are stored. A hash value for each transaction is computed instantly and stored in the exam activity log. The block generation process is initiated once the required numbers of transactions are recorded. The slot number *'s'* defines the number of transactions per block. Once a block is generated by any local node, it is attempted to validate. On successful validation, the corresponding block is to be added to the blockchain. In the failure of block validation, a rescue mechanism is initiated. The rescue mechanism is subjective to business policies. If the exam duration is completed, or if the student submits the exam before the final time, a final block needs to be generated. The final block consists of the exam log

ISSN: 2233-7857 IJFGCN

transaction along with the final response record of that student. The behavioral data obtained during the exam helps in deriving various properties. The final block helps in evaluation. As per the architecture, once the exam is done, the Evaluation process commences. The evaluation process can be carried just by confirming the last block hash value with the Quiz (exam) responses hash value. An algorithm for the above-mentioned procedure is mentioned at Algorithm 1

.

5. Validation of the proposed Examination scenario

The workflow for the online Examination scenario is presented through the Algorithmic procedure 1.

5.1 UML Activity Diagram

The examination activity begins once the student authenticates. As mentioned in the algorithm, the activity will be alive until the final time t_f . For every slot number's' of transactions, the procedure of block generation validation is carried out. As long as the clock is not expired or the student does not terminate, the system is live and active. If the student terminates in between, then the session will be closed after generating the final block. There are 4 decision elements in the activity diagram which leads to multiple paths. If the exam time expires or the student terminates voluntarily then the system reaches the end state.

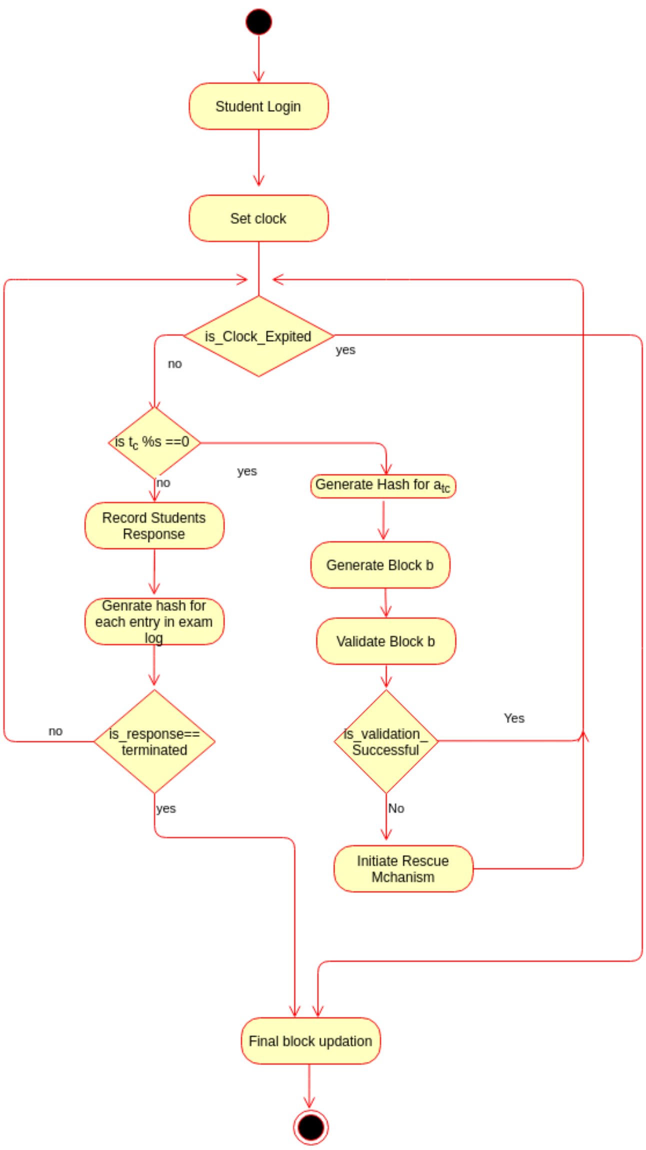

Figure 4 Activity diagram notation of the proposed Algorithm

ISSN: 2233-7857 IJFGCN

5.2 Petri Net Model for the Examination Scenario

A Petri net model is generated for the activity diagram generated above. The procedure described [5] for converting the activity diagram into the Petrinet is adopted and realized through the PIPE tool[15][26]. This scenario is for storing the response for a valid authentication. So, only login successful case is considered. To begin conversion, the initial node is taken as a place i.e. "Start". The next Place will be “Student login", which is obtained by the transition that student providing login credentials. Once this transition is fired, then student moves from Start state to the "Login Successful" state. Similarly, the intermediary steps/actions required to get into next state are considered as transitions represented with the rectangular bars, and the states are represented by circles in the following Figure 5.The generated Petri net is depicted in Figure 5.

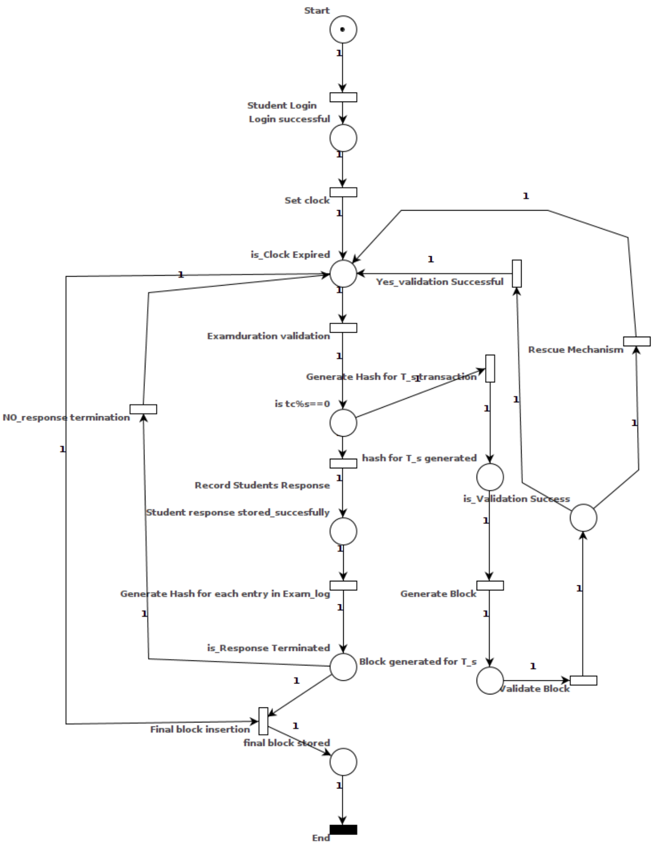

Figure 5 Petri net generated for the proposed Algorithm

6. Results & Discussion

The Generated Petri net Figure 5 is analyzed through PIPE tool. There are places and corresponding 13 Transitions in this Petri net.

Table 1 The Places in the generated Petri net

Place	Place name
P0	Start
P1	Login Success
P2	is_Clock Expired
P3	Is t_c%s==0
P4	Student Response stored successfully
P5	Is Response Terminated
P6	Hash for Ts generated

ISSN: 2233-7857 IJFGCN

P7	Block generated for t_s
P8	is_Validation Success
P9	final block stored

Table 2: Transition in the generated Petri net

Transition	Transition Name
T0	Student Login
T1	Set clock
T2	Exam duration validation
T3	Record Students Response
T4	Generate Hash for each entry in Exam Log
T5	Generate Hash for T_s transaction
T6	Generate Block
T7	Validate Block
T8	Rescue Mechanism
T9	Yes_validation Successful
T10	NO_response termination
T11	Final Block Insertion
T12	End

6.1 Matrix Analysis

The Incidence matrix is an alternative way of describing a graph from conventional methods. The matrix representation has The Incidence matrix D is obtained with the Backward and forward incidence matrices. The Backward incidence matrix is represented by D^- and Forward Incidence matrix with D^+. The incidence matrices D^+, D^-, D are (n_t x n_p) sized matrices where n_t is the number of transitions and n_p is the number of places. The backward incidence matrix also known as the input matrix consists of elements like d^-_{ij} which depicts the count of arcs joining p_j with transition t_i. The Forward incidence matrix D^+ also known as output matrix consisting of the elements like d^+_{ij} which depicts the count of arcs joining transition t_i with place p_j. Table 3,4,5 represents the forward incidence matrix, backward incidence matrix, combined incidence matrix.

	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
P0	1	0	0	0	0	0	0	0	0	0	0	0	0
P1	0	1	0	0	0	0	0	0	0	0	0	0	0
P2	0	0	1	0	0	0	0	0	0	0	0	1	0
P3	0	0	0	1	0	1	0	0	0	0	0	0	0
P4	0	0	0	0	1	0	0	0	0	0	0	0	0
P5	0	0	0	0	0	0	0	0	0	0	1	1	0
P6	0	0	0	0	0	0	1	0	0	0	0	0	0
P7	0	0	0	0	0	0	0	1	0	0	0	0	0
P8	0	0	0	0	0	0	0	0	1	1	0	0	0
P9	0	0	0	0	0	0	0	0	0	0	0	1	0

Figure 6 Forward Incidence Matrix D-

ISSN: 2233-7857 IJFGCN

	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
P0	0	0	0	0	0	0	0	0	0	0	0	0	0
P1	1	0	0	0	0	0	0	0	0	0	0	0	0
P2	0	1	0	0	0	0	0	0	1	1	1	0	0
P3	0	0	1	0	0	0	0	0	0	0	0	0	0
P4	0	0	0	1	0	0	0	0	0	0	0	0	0
P5	0	0	0	0	1	0	0	0	0	0	0	0	0
P6	0	0	0	0	0	1	0	0	0	0	0	0	0
P7	0	0	0	0	0	0	1	0	0	0	0	0	0
P8	0	0	0	0	0	0	0	1	0	0	0	0	0
P9	0	0	0	0	0	0	0	0	0	0	0	1	0

Figure 7 Backward Incidence Matrix D+

The combined incidence matrix D is defined as

$$\mathbf{D} = \mathbf{D}^{+} - \mathbf{D}^{-}.$$

	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
P0	−1	0	0	0	0	0	0	0	0	0	0	0	0
P1	1	−1	0	0	0	0	0	0	0	0	0	0	0
P2	0	1	−1	0	0	0	0	0	1	1	1	−1	0
P3	0	0	1	−1	0	−1	0	0	0	0	0	0	0
P4	0	0	0	0	1	−1	0	0	0	0	0	0	0
P5	0	0	0	0	1	0	0	0	0	0	−1	−1	0
P6	0	0	0	0	0	1	−1	0	0	0	0	0	0
P7	0	0	0	0	0	0	1	−1	0	0	0	0	0
P8	0	0	0	0	0	0	0	1	−1	−1	0	0	0
P9	0	0	0	0	0	0	0	0	0	0	0	1	−1

Figure 8 The Incidence matrix

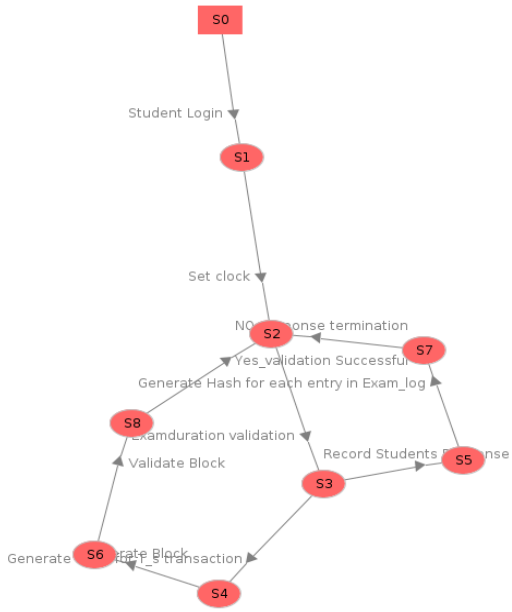

Figure 9 Reachability graph R for the Petri net N

The incidence matrix represents the relation between places and Transitions. The generated incidence matrices are helpful in the analysis of the Petri net. The generation of the Reachability graph and further analysis is explained below.

ISSN: 2233-7857 IJFGCN

6.2 The Reachability Graph Analysis

The Petri net analysis begins with the generation of its state space and representing it with a Reachability Graph [21]. The Markings of a Petri net are the nodes of the Reachability graph. The firing of an enabled transition is responsible for changing the state of the model. The sequence of firing results in getting a sequence of marking. This sequence of marking can be portrayed as a Reachability graph. There are two types of states in the state space, Vanishing, and Tangible. A state is said to be vanishing if transitions are fired immediately. Thus no time is spent in it. A tangible state is a state where there is no immediate firing of transitions. There will be some finite time spent in the tangible state.

There are 9 distinct states in the generated Reachability graph(Figure 9). The beginning state S0 is represented with a square box. The remaining tangible states are represented with elliptical shapes. The finite number of states in a Reachability graph confirms the Boundedness of the Petri net.GSPN ANALYSIS

The Stochastic Petri net becomes Markovian as it has both timed and immediate transitions. In a Markov process [2] future space only depends on the present state. As it's a Markov process, it generates a Markov chain. The evaluation of various performance parameters can be done with Markov Chain.

6.3.1 Steady-State Analysis

The Steady-State distribution for a Markov chain is computed using the following equations.

$$\pi Q = 0 \qquad (1)$$

$$\sum_{i=1}^{n} \pi_j = 1 \qquad (2)$$

Here Q is a square matrix where a generic term of Q is q_{ij} of order s where s=|R(PN)|, R is a Reachability matrix, PN is the Petri net . From the vector $\pi = (\pi_1 , \pi_2 , ..., \pi_s)$ the following performance measures are computed. Figure 10 represents the GSPN steady state Analysis Results.

$$\begin{array}{c} \\ M0 \\ M1 \\ M2 \\ M3 \\ M4 \\ M5 \\ M6 \\ M7 \\ M8 \end{array} \begin{array}{c} \begin{array}{cccccccccc} P0 & P1 & P2 & P3 & P4 & P5 & P6 & P7 & P8 & P9 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Figure 10 GSPN steady state Analysis Results

The Reachability graph has 9 markings out of which M0 is the initial marking which is represented with a square as depicted in Figure4. The states in the Reachability graph Correspond to the markings in the GSPN Steady-State Analysis matrix. Each state is a marking in the matrix.

6.3.2 Steady-State Distribution of Tangible States

The average Steady-State distribution explains the time spent at each marking in traversing a Petri net.

$$E[T] = \frac{E[N]}{E[S]} \qquad (3)$$

ISSN: 2233-7857 IJFGCN

E[N] is the average number of tokens in the network and E[S] is the average input rate i.e the average number of tokens into the Petri net.

Marking	Value
M0	-0
M1	-0
M2	0.27586
M3	0.13793
M4	0.13793
M5	0.13793
M6	0.13793
M7	0.13793
M8	0.03448

Figure 11 Steady State Distribution of Tangible States

The time spent at State 2 (Marking M2) is a bit higher than other states, as the condition in this state is verified for every transaction. The States S3,S4, S5, S6, S7 have equal chances of being taken, thus there are same Steady-State distribution values. The State S8 is explored only at the end to store the final block, thus it has a lesser value.

6.3.3 Probability of a Token to be Part of a Subset Marking

Let the set of markings in a specific SPN be represented by $H \subseteq$ R(PN). The probability of a token to be present in a state of the related subset of the Markov chain is attained by:

$$P[H] = \sum_{M_i \in M} \pi_i \qquad (4)$$

	μ=0	μ=1
Block generated for T_s	0.86207	0.13793
hash for T_s generated	0.86207	0.13793
is_Clock Expired	0.72414	0.27586
is_Response Terminated	0.86207	0.13793
is_Validation Success	0.96552	0.03448
is tc%s==0	0.86207	0.13793
Login successful	1	0
Start	1	0
Student response stored_succesfully	0.86207	0.13793
final block stored	1	0

Figure 12 Token Probability Density

This is the probability that each place has 0 or 1 tokens. It is a proportion of a time that a given place has given marking.

6.3.4 Average Number of Tokens

A Reachability graph consists of a set of Markings. A Marking is a set of places with tokens. A subset H(Pi,n) of the Reachability graph of Petri net R(PN) presents the sum of the number of tokens `n' at a place pi belongs to M.

$$H(p_i, n) = M \in R(PN) | M(p_i) = n \qquad (5)$$

For any place pi, has an average count of tokens mi attained by

ISSN: 2233-7857 IJFGCN

$$\bar{m_i} = \sum_{n=1}^{\infty} nP[H(p_i, n)]) \qquad (6)$$

The following table represents the Average number of tokens at a place.

Place	Number of Tokens
Block generated for T_s	0.13793
hash for T_s generated	0.13793
is_Clock Expired	0.27586
is_Response Terminated	0.13793
is_Validation Success	0.03448
is tc%s==0	0.13793
Login successful	0
Start	0
Student response stored_succesfully	0.13793
final block stored	0

Figure 13 Average Number of Tokens at a Place

It can be observed that the place "is Clock Expired" has the highest average number of tokens as this place is visited frequently compared to other places. From the activity diagram, we can deduce that multiple paths include this place. The initial places P0, P1 and the final place P9 are visited only once thus they have the least average number of tokens in any sequence of firing.

6.3.5 Probability of a Transition being Fired

Let the Reachability graph R(PN) of a Petri net has a subset HN_j which consists of an enabled transition t_j. Then the probability k_j for transition t_j being fired next is given by

$$k_j = \sum_{M_i \in HN_j} \pi_i(\lambda_j/(-q_{ij})) \qquad (7)$$

Here, the - q_{ij} is the total number of transitions enabled out of marking M_i.

6.3.6 Throughput of a Timed Transition

Throughput of a specific timed transition j, can be derived in the following manner

$$d_j = \sum_{s_i \in HN_j} \pi_i \qquad (8)$$

Where the throughput of a transition j is denoted by d_j , HN_j is the collection of each state in which the transition j is fired, the transition rate of transition j as λ_j in state s_i. The throughput of the produced Petri net is explained in Table 10.

ISSN: 2233-7857 IJFGCN

Transition	Throughput
Examduration validation	0.27586
Generate Block	0.13793
Generate Hash for each entry in Exam_log	0.13793
Generate Hash for T_s transaction	0.13793
NO_response termination	0.13793
Record Students Response	0.13793
Rescue Mechanism	0.03448
Set clock	0
Student Login	0
Validate Block	0.13793
Yes_validation Successful	0.03448
Final block insertion	0

Figure 14 The throughput of Timed Transitions

6.4 Simulation Results

The Simulation module in PIPE uses the Monte-Carlo random number generator to select a new state every time in the Petri net. At each new state, the average number of tokens is stored over a given number of transitions. This cycle can be repeated to find the average number of tokens associated with each marking. The standard error in this value is proportional to the square root of the number of cycles executed. 95% confidence interval for the average number of tokens per place is calculated using the intermediate values of each cycle. The developed Petri net is simulated with 100 firings and 5 replications.

Place	Average number of tokens	95% confidence interval (+/-)
Block generated for T_s	0.10891	0.02632
final block stored	0	0
hash for T_s generated	0.10891	0.03153
is_Clock Expired	0.21782	0.00951
is_Response Terminated	0.10891	0.03153
is_Validation Success	0.10891	0.02632
is tc%s==0	0.21782	0.00776
Login successful	0.0099	0
Start	0.0099	0
Student response stored_succesfully	0.10891	0.03803

Figure 15 Petri nets Simulation Results

The timed transitions in the Petri net are given default weightage. That can be observed in the semi uniform results in the throughput. Based on the design and the selection of real-time systems real time values can be given to transitions and perform analysis further.

6.4.1 State Space Analysis

The State space analysis determines the qualitative properties of Petri net. With the analysis of state space, we can conclude that the generated Petrinet is valid. Its Boundedness, Safeness, Deadlock Free (Liveness) are formally proven.

ISSN: 2233-7857 IJFGCN

Petri net state space analysis results

Bounded	true
Safe	true
Deadlock	false

Figure 16 Petri nets Analysis Results

Conclusion and Future Work

A procedure for the validation of SLE is proposed. An algorithmic procedure for the online examination scenario is presented. The correctness of the proposed procedure for achieving enhanced security is verified. Its workflow is converted into a UML-Activity diagram. Later, a Petri net is generated for formal analysis. The generated Petri net confirms the proposed procedure of Blockchain-enabled data storage mechanism for the online examination is valid. This formal validation procedure can be used for any Scenario-based evaluation of a Software architectures. This procedure can be further continued to achieve the other standardization challenges to reach higher levels of smartness.

Acknowledgment

The corresponding author avails funding from University Grants Commission, India in the form of NET-JRF (Junior Research Fellowship).

References

[1] Anil G.R., Moiz S.A. (2020) Blockchain Enabled Smart Learning Environment Framework. In: Satapathy S., Raju K., Shyamala K., Krishna D., Favorskaya M. (eds) Advances in Decision Sciences, Image Processing, Security and Computer Vision. Learning and Analytics in Intelligent Systems, vol 4. Springer, Cham

[2] L. Shi, Y. Li, and H. Feng, ‘‘Performance analysis of honeypot with petri nets,’’ Information, vol. 9, no. 10, p. 245, 2018.

[3] Jon Dron, “Smart learning environments, and not so smart learning environments: a systems view”, Smart Learning Environments (2018) 5:25, Chen, W.K. (1993). Linear Networks and Systems. Wadsworth, Belmont, 123-135.

[4] Tore Hoel, Jon Mason. Standards for smart education { towards a development frame-work. Smart Learning Environments 2018, 5: 3. at: https://doi.org/10.1186/s40561-018-0052-3.

[5] Yasmina Rahmoune, Allaoua Chaoui, Elhillali Kerkouche, “ A Framework for Modeling and Analysis UML Activity Diagram Using Graph Transformation”, International Workshop on the Use of Formal Methods in Future Communication Networks, (UFMFCN 2015)

[6] Muhammad Muzammal, Qiang Qu, Bulat Nasrulin.: Renovating blockchain with distributed databases: An open source system: Future Generation Computer Systems 90 (2019) 105–117

[7] Ali Fardinpour, Mir Mohsen Pedram, Martha Burkle, “Intelligent Learning Management Systems: Definition, Features and Measurement of Intelligence “,International Journal of Distance Education Technologies, 12(4), 19-31, October-December 2014

[8] Spector, JM. “*Conceptualizing the emerging field of smart learning environments”.* Smart Learning Environments 1(1), 5–10 (2014) https://doi.org/10.1186/s40561-014-0002-7

[9] Uskov, V L., Howlett, R J., Jain L C. (eds.), Smart Education and Smart e-Learning (Smart Innovation, Systems, and Technologies, 41) (Springer, London, 2015)

[10] IEEE Standard for Learning Technology-Learning Technology Systems Architecture (LTSA)," in IEEE Std 1484.1-2003, vol., no., pp.0_1-97 (2003).

[11] Adetoba B. T, Awodele O, and Kuyoro S. O.: E-learning security issues and challenges: A review: Journal of Scientific Research and Studies Vol. 3(5), pp. 96-100, (May 2016)

ISSN: 2233-7857 IJFGCN

[12] Abrar Ullah, Hannan Xiao, Trevor Barker: A classification of threats to remote online examinations: IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)(2016)

[13] C. Araújo, E. Cavalcante, T. Batista, M. Oliveira and F. Oquendo, "A Research Landscape on Formal Verification of Software Architecture Descriptions," in *IEEE Access*, vol. 7, pp. 171752-171764, 2019. doi: 10.1109/ACCESS.2019.2953858

[14] Wenxin WU, Motoshi SAEKI, " Specifying Software Architecture Based on Coloured Petri Nets", IEICE TRANS. & INF.& SYST., VOL.E83-D, NO.4 APRIL 2000 https://www.worldscientific.com/doi/pdf/10.1142/9789812389701_0011

[15] Sarah Tattersall, 2014, "PIPE — The Great Re-Plumbing", Department of Computing, Imperial College London.

[16] Andrea Pinna, Roberto Tonelli, Matteo Orru´ and Michele Marchesi, "A Petri Nets Model for Blockchain Analysis", arXiv:1709.07790v2 [cs.CR] 26 Sep 2017

[17] M. I. Fakhir and S. A. R. Kazmi, "Formal Specification and Verification of Self-Adaptive Concurrent Systems," in *IEEE Access*, vol. 6, pp. 34790-34803, 2018. doi: 10.1109/ACCESS.2018.2849821

[18] Irina Lomazova, Louchka Popova-Zeugmann, Arthur Bartels, " Controlling Boundedness for Live Petri Nets", Proceedings of 2017 4th International Conference on Control, Decision and Information Technologies (CoDIT'17) / April 5-7, 2017, Barcelona, Spain

[19] Shaibu Adekunle Shonola, Mike Joy: Security issues in E-learning and M-learning Systems: A Comparative Analysis: A Proceeding of 2nd WMG Doctoral Research and Innovation Conference (WMGRIC2015)

[20] Prveen Kumar, Shefalika Ghosh Samadd, Arun B. Samadd, Arun Kumar Misra. : Extending IEEE L TSA e-Learning Framework in Secured SOA Environment: 2nd International Conference on Education Technology and Computer (ICETC)(2010)

[21] Andrea Bobbio: SYSTEM MODELLING WITH PETRI NETS, A. G. Colombo and A. Saiz de Bustamante (eds.), Istituto Elettrotecnico Nazionale Galelio Ferraris, System Reliability Assessment, Kluwer,Kluwer p. c. pp 102-143, Italy,1990.

[22] IMS Global Learning Consortium (2003a). IMS Abstract Framework: White Paper Version 1.0. Online: https://www.imsglobal.org/af/afv1p0/imsafwhitepaperv1p0.htm

[23] JISC (n.d.) The E-Learning Framework. Joint Information Systems Council. http://www.elframework.org/framework.html.

[24] OKI (Open Knowledge Initiative): http://www.mit.edu/afs.new/athena/project/okidev/okiproject/apps/okichange/web/index.html

[25] Ma, J., Han, W., Ding, Z.: Behavior Analysis of Software Systems Based on Petri Net Slicing. In: Huang,D.-Sh., Jiang Ch., Bevilacqua, V., Figueroa, J.C. (eds.) ICIC 2012. LNCS, vol. 7389, pp. 475-482. Springer, Heidelberg (2012)

[26] PIPE: Platform Independent Petri net Editor, http://pipe2.sourceforge.net/ .

[28] Ossiannilsson, E. Challenges and Opportunities for Active and Hybrid Learning related to UNESCO Post 2015. In Handbook of Research on Active Learning and the Flipped Classroom Model in the Digital Age; J. Keengwe, G. Onchwari (Eds.); Hershey, PA: IGI Global. (pp. 333-351). doi:10.4018/978-1-4666-9680-8.ch017

[29] Abdulaziz Alkussayer; William H. Allen: A Scenario-Based Framework for the Security Evaluation of Software Architecture. In Proceedings of 2010 3rd International Conference on Computer Science and Information Technology

[30] Lopez, J.; Santana-Alonso, A.; Diaz-Cacho Medina, M. Formal Verification for Task Description Languages. A Petri Net Approach. Sensors, 2019, 19, 4965..

ISSN: 2233-7857 IJFGCN